

ALVR 2026

**Workshop on Advances in Language and Vision Research**

**Proceedings of the Workshop**

July 3, 2026

The ALVR organizers gratefully acknowledge the support from the following sponsors.

## **Gold**

 **Lambda**

©2026 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)  
317 Sidney Baker St. S  
Suite 400 - 134  
Kerrville, TX 78028  
USA  
Tel: +1-855-225-1962  
[acl@aclweb.org](mailto:acl@aclweb.org)

ISBN 979-8-89176-398-2

## Introduction

We are excited to welcome you to the 4th Workshop on Advances in Language and Vision Research (ALVR), co-located with ACL 2026 and held on July 3, 2026, in San Diego, California.

ALVR provides a dedicated forum for research at the intersection of natural language processing and computer vision. This year’s program features five invited talks from leading researchers, six spotlight presentations, and twenty-three poster presentations, spanning a wide range of topics at the frontier of language and vision research.

We received 47 submissions this year. After a thorough review process, we accepted 29 papers in total: 23 as archival papers and 6 through our non-archival track, which welcomes work concurrently under submission or recently published elsewhere. The archival acceptance rate is approximately 49%. Each submission received at least three reviews, with area chairs providing guidance to ensure the quality and breadth of the final program. We thank our area chairs—Yue Fan, Syrielle Montariol, Alane Suhr, Xin Eric Wang, and Qianqi Yan—for their careful deliberation, and all program committee members for their thorough and timely reviews.

We are grateful to our five distinguished invited speakers: Raymond J. Mooney (University of Texas at Austin), Lianhui Qin (UC San Diego), Amir Zadeh (Lambda Labs), Mohit Bansal (University of North Carolina at Chapel Hill), and Jiajun Wu (Stanford University), for sharing their insights and inspiring the community.

We gratefully acknowledge the generous support of our sponsor, Lambda Labs (Gold level), whose contribution has been instrumental to the success of the workshop.

We extend our sincere thanks to all authors who submitted to ALVR 2026, and to all attendees for their enthusiasm and participation. It is the community’s energy and commitment that makes this workshop a rewarding venue for advancing research at the intersection of language and vision.

Qianqi Yan, Lead Organizer

Syrielle Montariol, Yue Fan, Jing Gu, Jiayi Pan, Manling Li, Parisa Kordjamshidi, Alane Suhr, and Xin Eric Wang, Co-Organizers

# Organizing Committee

## Organizers

Qianqi Yan, University of California, Santa Barbara, USA  
Syrielle Montariol, University of California, Berkeley, USA  
Yue Fan, Autodesk Research, USA  
Jing Gu, xAI, USA  
Jiayi Pan, xAI, USA  
Manling Li, Northwestern University, USA  
Parisa Kordjamshidi, Michigan State University, USA  
Alane Suhr, University of California, Berkeley, USA  
Xin Eric Wang, University of California, Santa Barbara, USA

## Program Committee

### Area Chairs

Qianqi Yan, University of California, Santa Barbara  
Yue Fan, Autodesk  
Syrielle Montariol, ISIR, Sorbonne Université, France  
Alane Suhr, University of California, Berkeley  
Xin Eric Wang, University of California, Santa Barbara and Simular

### Reviewers

Rishabh Agrawal, Amazon  
Yifei Bi, Facebook  
Zhong Chen, Southern Illinois University-Carbondale  
Sahil Rajesh Dhayalkar, Brain Corporation  
Christopher Driggers-Ellis, University of Florida  
Yue Fan, Autodesk  
Zhiyuan Gao, University of Southern California  
Jingnan Gao, NVIDIA and Shanghai Jiao Tong University  
Christan Grant, University of Florida  
Jinru Han, University of California, Los Angeles  
Patrick Huber, Facebook  
Ben Jenkins, Florida Atlantic University  
Yanru Jiang, University of California, Los Angeles  
Dayeon Ki, Microsoft and University of Maryland, College Park  
Sriram Kollipara, Walmart  
Pawan Kumar, International Institute of Information Technology Hyderabad  
Yuecheng Li, Kuaishou  
Chuhan Li, University of California, Santa Barbara  
Irene Li, University of Tokyo  
Mao Lin, University of California, Merced  
Shih-chih Lin, National Tsinghua University  
Junbin Lu, University of Washington  
Sepideh Mamooler, EPFL  
Srijith Ravikumar, Amazon  
Anisha Saha, Saarland Informatics Campus, Max-Planck Institute  
Tongyue Shi, Peking University  
Andrew Shin, Keio University  
Kaleen Shrestha, University of Southern California  
Thoudam Doren Singh, National Institute of Technology Meghalaya  
Sakthivel Sivaraman, NVIDIA  
Elior Sulem, Ben-Gurion University of the Negev  
Zhimin Sun, AMS AI Lab, Tencent  
Rohith Uppala, LinkedIn  
Pavan Kumar Velaga, Amazon and Rutgers University  
Zirui Wei, C3.ai  
Yibo Yan, HKUST and Alibaba  
Yang Yan, Southern Illinois University-Carbondale  
Junhuan Yang, Amazon

Rongtian Ye, Aalto University  
Ziyao Zeng, Yale University  
Jing Zhang, Amazon  
Pingyue Zhang, Northwestern University  
Hang Zhao, Northeastern University  
Duo Zhou, University of Illinois at Urbana-Champaign

## Table of Contents

<i>Thinking in Pictures: A Diagnostic Study of Visual vs. Textual Chain-of-Thought Reasoning in Vision-Language Models</i> Ben Jenkins .....	1
<i>A Zipfian Analysis of Visual Token Distributions for AI-Generated Images</i> Andrew Shin .....	13
<i>Semantically Aware Optimal Transport for Dense Label Transfer</i> Preeti, Kiran Ravish, Ankita Kushwaha and Pawan Kumar .....	18
<i>CoSMoEs: Compact Sparse Mixture of Experts</i> Patrick Huber, Akshat Shrivastava, Ernie Chang, Chinnadhurai Sankar, Ahmed A Aly and Adithya Sagar .....	46
<i>GraphicWeaver: Benchmarking Agentic Planning for Graphic Design Generation</i> Dayeon Ki, Tianyi Zhou, Marine Carpuat, Gang Wu, Puneet Mathur and Viswanathan Swaminathan .....	57
<i>Scaling Vision–Language Models for Pharmaceutical Long-Form Video Reasoning on Industrial GenAI Platform</i> Suyash Mishra, Qiang Li, Srikanth Patil, Satyanarayan Pati and Baddu Narendra .....	85
<i>PGGA: A Plan-Grounded GUI Agent for Automated Device Support</i> Lei Hsiung, Zhiyu Chen, Seonhoon Kim and Qun Liu .....	105
<i>CAFES: A Collaborative Multi-Agent Framework for Multi-Granular Multimodal Essay Scoring</i> Jiamin Su, Yibo Yan, Zhuoran Gao, Han Zhang, Xiang Liu, Huiyu Zhou and Xuming Hu ...	115
<i>GM-PRM: A Generative Multimodal Process Reward Model for Multimodal Mathematical Reasoning</i> Jianghangfan Zhang, Yibo Yan, Kening Zheng, Xin Zou, Song Dai and Xuming Hu .....	139
<i>Look Where You’re Told: Instruction-Consistent Attention for GUI Grounding</i> Seonhoon Kim, Zhiyu Chen, Xin Li and Qun Liu .....	155
<i>From Pixels to BFS: High Maze Accuracy Does Not Imply Visual Planning</i> Alberto Gonzalo Rodriguez Salgado .....	164
<i>When Relations Break: Analyzing Relation Hallucination in Vision-Language Model Under Rotation and Noise</i> Philip Wootae Shin, Ajay Narayanan Sridhar, Lakshmi Sivani Devarapalli, Rui Zhang, Jack Sampson and Vijaykrishnan Narayanan .....	180
<i>VLCE: A Knowledge-Enhanced Framework for Image Description in Disaster Assessment</i> Md. Mahfuzur Rahman, Marufa Kamal, Fahad Rahman, Sunzida Siddique, Ahmed Rafi Hasan, Mohd Ariful Haque, Kishor Datta Gupta and Roy George .....	186
<i>Beyond Visual Similarity: Rule-Guided Multimodal Clustering with explicit domain rules</i> Kishor Datta Gupta, Mohd Ariful Haque, Marufa Kamal, Ahmed Rafi Hasan, Md. Mahfuzur Rahman and Roy George .....	199
<i>ChartDiff: A Large-Scale Benchmark for Comprehending Pairs of Charts</i> Rongtian Ye .....	209

<i>Formal Machine Interpretation for the Semasiographic Mixtec Codices of Precolonial and Early Colonial Mesoamerica</i>	
Christopher Driggers-Ellis, Gabriel Ayoubi, girish.salunke811@gmail.com girish.salunke811@gmail.com and Christan Grant .....	230
<i>Temporal-Linguistic Adaptive Streaming for Continuous Sign Language Translation</i>	
Arshia Kermani, Habib Irani, Deautun Ross and Vangelis Metsis .....	239
<i>FADE: Probing the Limits of VLMs on fine-grained OCR</i>	
Deep Shah, Nehal Kathrotia and Sanket Badhe .....	249
<i>Efficient Visual Grounding in VQA via Question-Guided Sparse Attention</i>	
Prasanth .....	260
<i>Systematic Performance Degradation in Indic Vision-Language Models: Evidence from Hindi and Telugu</i>	
Rishikant Chigrupaatii, Ponnada Sai Tulasi Kanishka, Lalit Chandra Routhu, Martin Patel, Sama Supratheek Reddy, Divyam Gupta, Rajiv Misra and Rohun Tripathi .....	272
<i>How Fragile Is Vision-Language Alignment? Mapping Concept Disruption Under Text-to-Image Personalization</i>	
Mujtaba Hasan .....	278
<i>The Compositional Grounding Gap: Why Vision-Language Models Fail at Relational Reasoning and How to Fix It</i>	
Kaustubh S. Bukkapatnam .....	287
<i>HalluTrace: Causal Attribution and Source-Targeted Decoding for Hallucination in Large Vision-Language Models</i>	
Kaustubh S. Bukkapatnam .....	294

# Thinking in Pictures: A Diagnostic Study of Visual vs. Textual Chain-of-Thought Reasoning in Vision-Language Models

Ben Jenkins

PhD Candidate, Florida Atlantic University

benrossjenkins@gmail.com

## Abstract

Chain-of-thought (CoT) reasoning has become a standard technique for eliciting complex reasoning in large language models, and recent work has extended it to vision-language models (VLMs). However, virtually all multimodal CoT methods generate intermediate reasoning steps in natural language, even for inherently visual problems such as spatial reasoning, geometric manipulation, and object tracking. We ask a focused question: *when does textual reasoning help or hurt a VLM, and does generating visual artifacts help because of the code or because of the rendered image?* We present VISCoT-DIAG, a diagnostic benchmark of 1,200 instances across five visual reasoning categories, and compare five CoT paradigms across four VLMs. Our results characterize a consistent modality gap: textual CoT degrades spatial transformation by up to 16.5% and multi-object tracking by 12.7%, while visual CoT yields gains of up to 25.4%. A code-only ablation (V-CoT-NF) recovers only 36% of the V-CoT gain over T-CoT on spatial transformation, indicating that the rendered image, not code generation alone, drives most of the improvement. We identify three failure modes (spatial state collapse, transformation hallucination, tracking loss) and show that adaptive modality routing achieves 78.2% accuracy versus 74.0% for V-CoT-everywhere on this benchmark. We discuss the scope of these conclusions and recommend practitioners use visual CoT for spatial-transformation-heavy tasks and textual CoT for compositional counting.

## 1 Introduction

Chain-of-thought (CoT) prompting has dramatically improved the reasoning capabilities of large language models by encouraging models to decompose problems into intermediate steps before arriving at an answer (Wei et al., 2022; Kojima et al., 2022). This paradigm has been extended to vision-language models (VLMs), where multimodal CoT

methods generate textual rationales that incorporate information from both visual and linguistic inputs (Zhang et al., 2024; Lu et al., 2022).

Recent benchmarks have already documented that direct answering and naive textual CoT underperform on multimodal tasks requiring expert knowledge or fine-grained visual grounding (Zhao et al., 2025; Yue et al., 2025). A growing body of work has advanced multimodal CoT through structured reasoning stages (Xu et al., 2025; Thawakar et al., 2025), visual tool use (Hu et al., 2024; Wang et al., 2025), and visualization of reasoning traces (Wu et al., 2024). What is less studied, and what we focus on here, is a more controlled question: *within visual reasoning, where exactly does textual CoT help or hurt, and when visual CoT helps, is the gain coming from code-based symbolic reasoning or from the rendered image itself?*

Consider the motivating example in Figure 1. A model is shown a scene with colored shapes and asked: *“After rotating the red triangle 90° clockwise and reflecting the blue square horizontally, is the triangle above or below the square?”* Under textual CoT, the model must verbally describe each transformation and its spatial consequence, a process that is error-prone because language lacks the precision to faithfully encode continuous spatial configurations. Under visual CoT, the model can draw the transformed objects directly, producing a visual representation that grounds subsequent reasoning. A natural objection is that V-CoT could be “cheating” by delegating geometric computation to a Python interpreter; we address this directly in §5 via a code-only ablation.

This phenomenon is consistent with cognitive-science findings on mental rotation (Shepard and Metzler, 1971), dual coding theory (Paivio, 1991), and mental imagery (Kosslyn, 1995), though we caution that VLM behavior need not mirror human cognition.

We make three contributions. First, we introduce

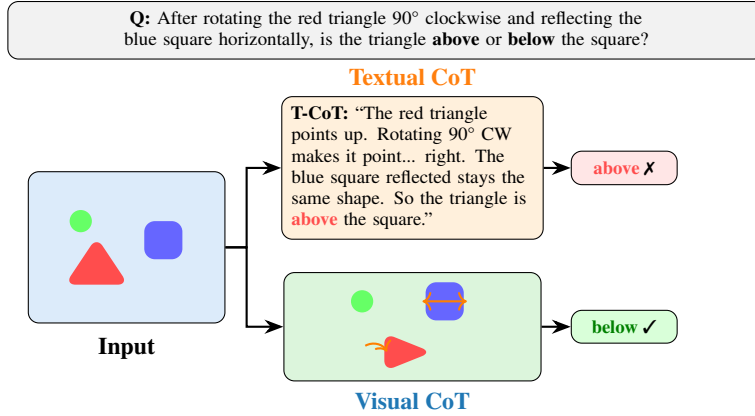


Figure 1: Motivating example. Given a stated spatial-transformation question (top), textual CoT produces a plausible but incorrect verbal trace (the model hallucinates the post-rotation spatial relationship), while visual CoT draws the actual transformations and grounds the answer in the rendered image.

VISCoT-DIAG, a diagnostic benchmark of 1,200 instances spanning five visual reasoning categories with fine-grained difficulty calibration (§3). Second, we conduct a controlled comparison of *five* CoT paradigms (direct, textual, structured textual, visual, and visual-no-feedback) across four VLMs, isolating not only the effect of reasoning modality but also whether visual gains stem from code generation or from observing the rendered image (§4–§5). Third, we document three failure modes of textual CoT in visual domains and provide empirical guidance for when each modality helps (§6).

We emphasize the scope of our claims: VISCoT-DIAG comprises relatively simple, controlled visual tasks. Our findings should be read as a diagnostic characterization on this controlled distribution rather than a universal claim about visual reasoning at large; we discuss this scope explicitly in §8.

## 2 Related Work

**Textual Chain-of-Thought.** Wei et al. (2022) demonstrated that prompting LLMs to produce intermediate reasoning steps improves performance on arithmetic, commonsense, and symbolic reasoning. Zero-shot variants achieve similar effects without exemplars (Kojima et al., 2022). These methods operate exclusively in natural language.

**Multimodal Chain-of-Thought.** Zhang et al. (2024) proposed a two-stage framework separating rationale generation from answer inference. Lu et al. (2022) showed that generating explanations as CoT chains improves science QA. LLaVA-CoT (Xu et al., 2025) structures reasoning into four textual stages and achieves strong results through su-

pervised fine-tuning. LlamaV-o1 (Thawakar et al., 2025) introduces step-level evaluation. Compositional CoT (Mitra et al., 2024) uses scene graphs as intermediate textual representations. The common thread is that intermediate reasoning is *entirely textual*.

### Multimodal Benchmarks Highlighting CoT Limits.

Several recent benchmarks have already shown that direct answering and naive textual CoT struggle on knowledge-intensive or fine-grained multimodal tasks. MMMU (Yue et al., 2024), MMMU-Pro (Yue et al., 2025), MMStar (Chen et al., 2024), and MMVU (Zhao et al., 2025) all report large headroom for multimodal models, and MMMU-Pro in particular documents brittleness of textual reasoning when visual content is presented in non-canonical formats. Our work is complementary: rather than expanding the difficulty of multimodal benchmarks, we hold task complexity controlled and ablate the reasoning *paradigm*, asking when textual reasoning specifically helps or hurts.

### Visual Intermediate Representations.

Visualization-of-Thought (VoT) (Wu et al., 2024) prompts LLMs to generate ASCII visualizations during spatial reasoning. Visual Sketchpad (Hu et al., 2024) enables VLMs to draw via code execution as part of an agentic reasoning loop, achieving gains on math and vision tasks. VisuoThink (Wang et al., 2025) combines visual-textual interleaving with tree search. The Visual CoT dataset (Shao et al., 2024) contributes bounding-box annotations as intermediate grounding. Each of these demonstrates the

value of visual reasoning in specific settings; we provide a controlled paradigm comparison and, importantly, a code-only ablation that disentangles the contributions of code generation from visual perception of the rendered output.

### 3 VISCoT-DIAG Benchmark

To systematically evaluate the effect of reasoning modality, we construct VISCoT-DIAG, comprising 1,200 problem instances organized into five categories based on the dominant type of visual reasoning required. Figure 2 provides an overview.

#### 3.1 Reasoning Categories

**Spatial Relation (SR).** Tasks requiring judgment about relative positions, orientations, and arrangements of objects. We source from BLINK spatial (Fu et al., 2024) and CLEVR (Johnson et al., 2017), filtering for unambiguous ground truth.

**Spatial Transformation (ST).** Tasks requiring mental manipulation of visual elements: rotation, reflection, translation, and compositions thereof. We construct synthetic instances using procedurally generated 2D polygons (triangles, quadrilaterals, pentagons, L-shapes) with controlled transformation parameters.

**Multi-Object Tracking (MOT).** Tasks requiring the model to track positions or states of multiple objects through a sequence of changes described in natural language. Generated programmatically using grid-world environments with 3 to 7 colored shapes and 2 to 6 sequential movement instructions.

**Compositional Counting (CC).** Tasks requiring identification, filtering, and counting of objects based on conjunctions of visual attributes. Adapted from CLEVR (Johnson et al., 2017) and GQA (Hudson and Manning, 2019).

**Geometric Reasoning (GR).** Tasks involving properties of geometric figures: computing angles, identifying congruent shapes, reasoning about area or perimeter, and applying geometric theorems. Drawn from MathVista (Lu et al., 2024) geometry subsets plus synthetic instances.

#### 3.2 Difficulty Calibration

Within each category, we calibrate difficulty along category-specific axes. **Easy:** 1–2 reasoning steps or 2–3 objects; **Medium:** 3–4 steps or 4–5 objects; **Hard:** 5+ steps or 6–7 objects. We validate

difficulty through pilot experiments with three human annotators, confirming monotonically decreasing accuracy across levels (Easy: 94.2%, Medium: 81.7%, Hard: 63.4%).

## 4 Experimental Setup

### 4.1 CoT Paradigms

We compare *five* reasoning paradigms applied at inference time via prompting, requiring no fine-tuning. The fifth paradigm, V-CoT-NF, isolates whether V-CoT gains stem from code generation or from visual perception of the rendered output. Figure 3 illustrates the paradigms.

**Direct (D).** The model receives image and question, produces an answer.

**Textual CoT (T-CoT).** The model is prompted to “think step by step” (Wei et al., 2022; Kojima et al., 2022).

**Structured Textual CoT (ST-CoT).** The model produces reasoning in four stages (summary, observation, reasoning, answer), mirroring Xu et al. (2025) as a prompting strategy rather than a fine-tuning objective.

**Visual CoT, no feedback (V-CoT-NF).** The model is prompted exactly as in V-CoT to generate Python annotation code. The code is executed silently and the resulting image is *not* returned. The model must commit to an answer based on its mental simulation of what the code would have produced. If V-CoT-NF matches V-CoT, then code generation alone explains V-CoT gains; if it matches T-CoT, then visual perception of the rendered image is essential.

**Visual CoT (V-CoT).** The model alternates between Act (generate annotation code) and Observe (the rendered image is re-encoded and shown back to the model), with up to five turns. Annotation primitives include shapes, lines, arrows, text, and affine transformations (rotation, reflection, translation), following Hu et al. (2024).

### 4.2 Models

We evaluate four VLMs spanning closed-source frontier and open-weight large-scale: **GPT-5** (OpenAI, 2025), **Claude Opus 4.6** (Anthropic, 2026), **Gemini 2.5 Pro** (Gemini Team, Google DeepMind, 2025), and **Qwen3-VL-72B** (Bai et al., 2025). All use greedy decoding (temperature 0) for reproducibility.

## VISCO-T-DIAG: Five Diagnostic Reasoning Categories

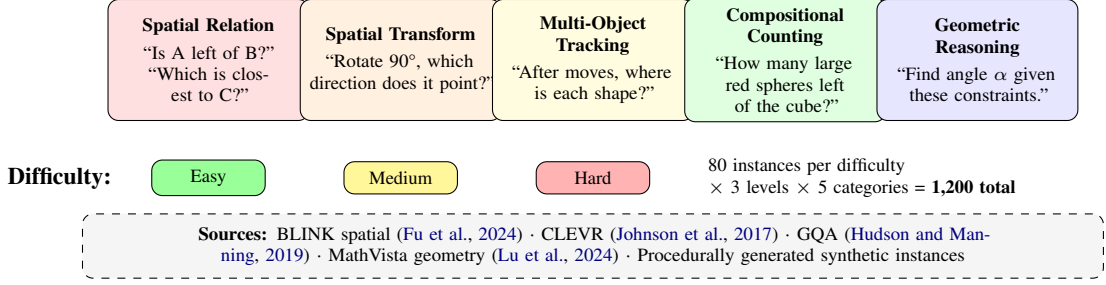


Figure 2: Overview of the VISCO-T-DIAG benchmark. Five categories target distinct visual reasoning abilities, with three difficulty levels per category. Instances are sourced from existing benchmarks and procedural generation.

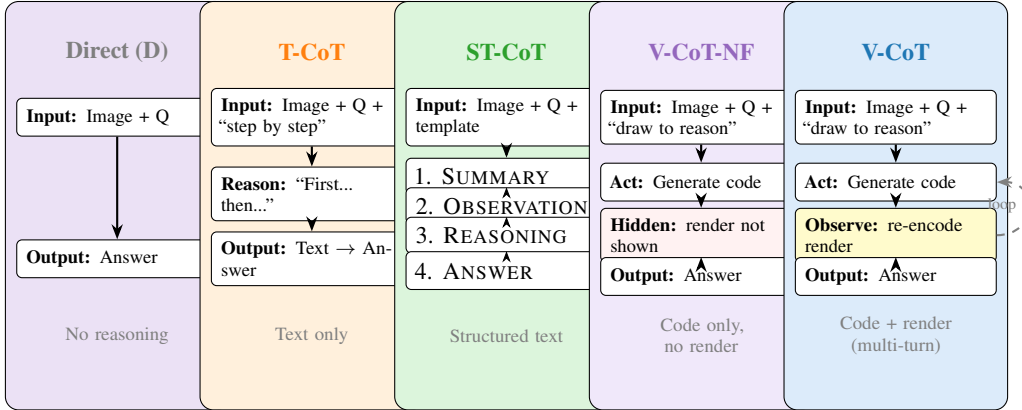


Figure 3: The five CoT paradigms compared in our study. **Direct** produces an answer in one pass. **T-CoT** generates a free-form reasoning chain. **ST-CoT** follows a four-stage template mirroring Xu et al. (2025). **V-CoT-NF** generates the same drawing code as V-CoT but the rendered image is *not* returned to the model, isolating the contribution of code generation. **V-CoT** alternates between code generation and observing the updated image, following Hu et al. (2024).

### 4.3 Evaluation Metrics

All tasks use **exact-match accuracy** against ground truth. For free-form numeric answers (GR), we allow  $\pm 1\%$  tolerance. We additionally report **premise-conclusion consistency (PCC)**, the fraction of instances where the model’s final answer is logically entailed by its own reasoning steps, assessed by a Claude Opus 4.6 judge (we verified 92% agreement with a Gemini 2.5 Pro judge on a 100-instance subset). **Annotation quality (AQ)** for V-CoT and V-CoT-NF measures whether generated visualizations are semantically meaningful (assessed on the executed render even when not shown to the model).

## 5 Results

### 5.1 Main Results

Table 1 reports accuracy across all five categories and five paradigms, averaged over the four VLMs.

The V-CoT-NF column lets us decompose the V-CoT gain into a code-generation contribution and a visual-feedback contribution.

Three observations emerge. First, T-CoT degrades performance on ST ( $-16.5$ ) and MOT ( $-12.7$ ). Notably, T-CoT’s category-level wins on CC and GR ( $+16.8$  and  $+13.8$ ) are almost exactly offset by its ST and MOT losses: on average across visual reasoning, T-CoT provides essentially no benefit over direct answering ( $57.3$  vs.  $56.1$ ). Second, V-CoT provides consistent gains across all categories except CC ( $+12$  to  $+25$ ). Third, V-CoT-NF produces partial gains: on ST it recovers from  $-16.5$  (T-CoT) to  $-1.4$  (essentially matching Direct), and on GR it captures most of the V-CoT improvement. The remaining gap between V-CoT-NF and V-CoT is largest on ST ( $26.5$  points) and MOT ( $19.4$ ), the categories where tracking visual state matters most.

Category	Direct	T-CoT	ST-CoT	V-CoT-NF	V-CoT	Best $\Delta$ vs. D
Spatial Relation (SR)	66.1	70.4 +4.3	72.0 +5.9	69.0 +2.9	<b>78.2 +12.1</b>	+12.1 (V-CoT)
Spatial Transformation (ST)	51.7	35.2 -16.5	38.1 -13.6	50.3 -1.4	<b>76.8 +25.1</b>	+25.1 (V-CoT)
Multi-Object Tracking (MOT)	59.0	46.3 -12.7	50.3 -8.7	53.6 -5.4	<b>73.0 +14.0</b>	+14.0 (V-CoT)
Compositional Counting (CC)	55.9	72.7 +16.8	<b>74.7 +18.8</b>	68.8 +12.9	68.4 +12.5	+18.8 (ST-CoT)
Geometric Reasoning (GR)	48.0	61.8 +13.8	63.8 +15.8	64.0 +16.0	<b>73.4 +25.4</b>	+25.4 (V-CoT)
<b>Average</b>	56.1	57.3	59.8	61.1	<b>74.0</b>	+17.8 (V-CoT)

Table 1: Accuracy (%) averaged across four VLMs on VISCOT-DIAG. Colors indicate change relative to Direct (green: improve; red: decline). Textual CoT degrades performance on ST and MOT; Visual CoT provides consistent gains. V-CoT-NF, which executes the same drawing code as V-CoT but hides the render from the model, sits between T-CoT and V-CoT on spatial tasks, though on CC it underperforms T-CoT. Best result per category is **bolded**.

## 5.2 Decoupling Code Generation from Visual Feedback

The V-CoT-NF ablation directly addresses the concern that V-CoT might be “cheating” by delegating geometric computation to a Python interpreter. If so, V-CoT-NF (which runs the same code) should match V-CoT, since the interpreter performs identical computation either way. The data does not support that interpretation. On ST, V-CoT-NF reaches 50.3% versus 76.8% for V-CoT and 35.2% for T-CoT, recovering only 36% of the V-CoT gain over T-CoT. On MOT, V-CoT-NF recovers 27% of the gain.

We interpret this as evidence that the rendered image carries information the model cannot reconstruct from the code alone. Generating drawing code does provide a measurable benefit on its own, likely because writing code to draw forces the model to commit to precise spatial coordinates, but the act of *seeing* the rendered configuration provides additional accuracy that code-based simulation cannot fully replace. This is consistent with the rendered image functioning as an external visuospatial scratchpad against which the model can verify or correct its mental simulation.

The pattern also clarifies where code generation is most useful in isolation. On GR, V-CoT-NF achieves +16.0 over Direct, recovering 63% of V-CoT’s gain. Geometric reasoning involves angle and length computations that benefit from explicit symbolic structure (declaring variables, applying formulas), and much of this benefit transfers without visual feedback. On ST, by contrast, the symbolic structure of the code (an `img.rotate(-90)` call) does not by itself disambiguate the spatial outcome; the model needs to see the result. On CC, V-CoT-NF (68.8%) actually *underperforms* T-CoT (72.7%) by 3.9 points: when the task rewards arithmetic and attribute fil-

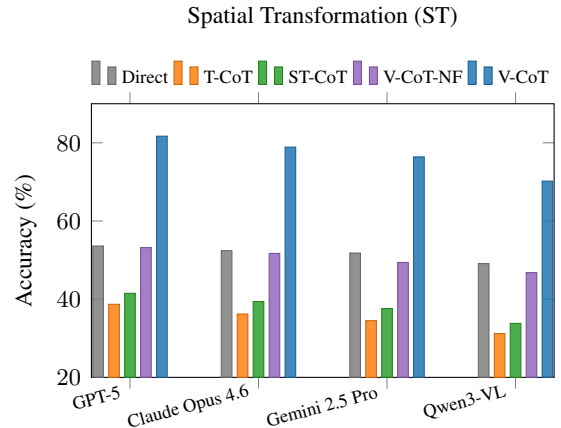


Figure 4: Per-model accuracy on Spatial Transformation. T-CoT (orange) decreases accuracy below Direct (gray) for every model. V-CoT-NF (purple) approaches Direct-level performance; only V-CoT with the rendered image (blue) produces the large gains. The pattern is consistent across all four VLMs.

tering, tokens spent producing drawing code that the model cannot see appear to displace tokens that would otherwise be used for symbolic decomposition.

## 5.3 Per-Model Results

Figure 4 disaggregates ST results by model across all five paradigms. The pattern is consistent across all four VLMs.

GPT-5 achieves the largest V-CoT gains on ST (+43.0 over T-CoT), likely because its code generation produces higher-quality annotations. The V-CoT vs. V-CoT-NF gap is also model-dependent: 28.5 for GPT-5, 27.2 for Claude Opus 4.6, 27.0 for Gemini 2.5 Pro, and 23.4 for Qwen3-VL. Models that benefit more from rendered feedback may also have stronger visual encoders, though we do not measure encoder quality directly.

Difficulty	T-CoT	V-CoT-NF	V-CoT	$\Delta (V-T)$
<i>Spatial Transformation</i>				
Easy	54.8	65.7	75.2	+20.4
Medium	33.1	52.0	82.2	+49.1
Hard	17.7	33.2	73.0	+55.3
<i>Compositional Counting</i>				
Easy	84.9	80.9	72.8	-12.1
Medium	71.7	69.1	68.0	-3.7
Hard	61.4	56.4	64.5	+3.1

Table 2: Accuracy (%) by difficulty level (averaged across models). On ST, the V-CoT-NF row shows that code generation alone is competitive at Easy difficulty but collapses on Hard chains (33.2% versus 73.0% for V-CoT). Visual feedback is what makes V-CoT robust to chain length.

**Adaptive Routing.** A DistilBERT classifier that selects T-CoT vs. V-CoT per question achieves 78.2% average accuracy versus 74.0% for V-CoT-everywhere and 57.3% for T-CoT-everywhere on this benchmark (84.7% routing accuracy; see Appendix D). Question-level modality routing is feasible and beneficial.

#### 5.4 Scaling with Difficulty

Table 2 shows how the modality gap interacts with task difficulty. On ST, the V-CoT vs. T-CoT gap widens from 20.4 points at Easy to 55.3 points at Hard. T-CoT errors compound with each additional transformation. Notably, at Hard difficulty T-CoT accuracy on ST falls to 17.7%, well below the 50% chance baseline for binary spatial questions: T-CoT is not merely failing to help but is producing systematically wrong answers. We attribute this to a salience-anchoring bias under uncertainty: when the model cannot resolve a transformation’s outcome verbally, it tends to default to the most prominent spatial relation described earlier in the chain rather than tracking the cumulative state. The V-CoT-NF row shows that code-only reasoning also degrades at Hard difficulty (from 65.7 at Easy to 33.2 at Hard), indicating that the code-based component of V-CoT is not robust to long transformation chains; visual feedback is what stabilizes performance.

#### 5.5 Annotation Quality

For V-CoT, annotation quality (AQ) varies across models: GPT-5 produces semantically correct annotations 89.4% of the time, followed by Claude Opus 4.6 (85.7%), Gemini 2.5 Pro (83.6%), and

Qwen3-VL-72B (78.1%). Conditioning on AQ: V-CoT accuracy on instances with correct annotations is 79.6%, compared to 42.8% on instances with incorrect annotations. V-CoT-NF accuracy under correct annotations is 64.3% versus 53.7% under incorrect annotations. The gap between V-CoT and V-CoT-NF is largest precisely when annotations are correct (15.3 points), confirming that the model is reading correct visual content from the rendered image, not just benefiting from the structuring effect of code.

## 6 Analysis: Failure Modes of Textual CoT

Through qualitative analysis of 200 error cases (stratified by category and model, where T-CoT underperforms), we identify three recurring failure modes. Two annotators achieved 89% agreement on failure mode labels (Fleiss’  $\kappa = 0.82$ ). Figure 5 illustrates each mode.

### 6.1 Spatial State Collapse

When reasoning about spatial relationships across multiple steps, textual representations progressively lose spatial fidelity. The model may correctly describe pairwise relationships in isolation but fail to maintain a globally consistent spatial configuration. PCC on SR tasks degrades sharply with object count for T-CoT (91.2% with 2 objects to 51.3% with 5+) while V-CoT remains stable (94.8% to 84.7%); see Figure 6. Notably, V-CoT-NF tracks T-CoT closely on this metric (89.7% to 56.4%), indicating that the rendered visual layout, not the act of writing code, is what stabilizes spatial consistency.

### 6.2 Transformation Hallucination

Asked to predict the result of spatial transformations, T-CoT frequently produces plausible-sounding but incorrect descriptions. On multi-step transformation chains,<sup>1</sup> T-CoT accuracy is 18.3% while V-CoT maintains 61.7%. Even single-step transformations suffer: T-CoT 44.1% versus V-CoT 76.2%. V-CoT-NF on the same chains reaches 38.5%, confirming that the rendered output contributes substantively beyond what code generation alone provides.

<sup>1</sup>Multi-step here means transformation chains of length  $\geq 2$ , partially overlapping but not identical to the Hard difficulty stratum in Table 2.

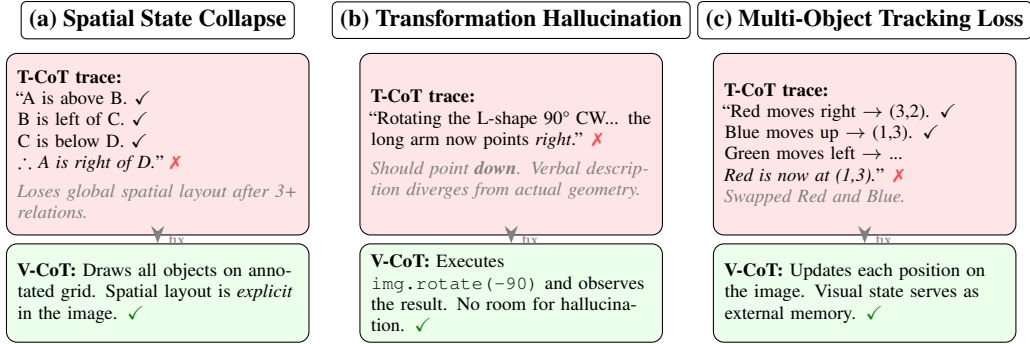


Figure 5: Three failure modes of textual CoT on visual reasoning tasks, and how visual CoT resolves each. (a) Spatial State Collapse. (b) Transformation Hallucination. (c) Tracking Loss.

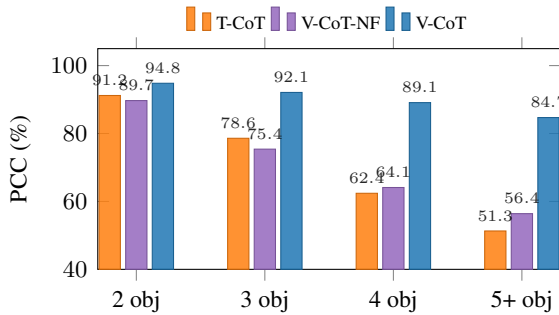


Figure 6: Premise-conclusion consistency on Spatial Relation tasks by object count. T-CoT and V-CoT-NF degrade similarly; only V-CoT (with rendered feedback) maintains consistency.

### 6.3 Multi-Object Tracking Loss

When tracking multiple objects through sequential state changes, T-CoT must maintain a verbal ledger. We observe two sub-patterns on 5-object instances: *identity confusion* (T-CoT 34.7% vs. V-CoT 8.2%) and *update omission* (T-CoT 22.1% vs. V-CoT 5.4%). V-CoT-NF rates fall between (identity confusion 26.3%, update omission 17.8%): code structure helps somewhat but cannot replace external visual memory.

### 6.4 Failure Mode Distribution

Table 3 reports failure mode frequency across error cases. Spatial state collapse dominates SR errors, transformation hallucination dominates ST, and tracking loss dominates MOT. These modes are rare on CC, where T-CoT errors stem from counting mistakes and attribute misidentification.

## 7 Discussion

**Are VLMs Actually “Thinking in Pictures”?** A reasonable concern is that V-CoT might not constitute visual reasoning at all: the model writes code,

Category	State Collapse	Trans. Halluc.	Track. Loss
SR	48.3%	12.1%	6.9%
ST	15.7%	61.4%	8.6%
MOT	22.8%	5.3%	57.9%
CC	8.2%	3.1%	2.7%
GR	18.6%	38.4%	4.3%

Table 3: Failure mode frequency in T-CoT errors. Each cell shows the share of errors attributed to each mode; rows need not sum to 100%.

and the Python interpreter performs the geometric computation. The V-CoT-NF ablation directly addresses this. If V-CoT were pure delegation, V-CoT-NF (which runs identical code but hides the result) would match V-CoT, since the interpreter does the same work either way. Instead, V-CoT-NF underperforms V-CoT by 26.5 points on ST and 19.4 points on MOT. The model is reading information off the rendered image that it cannot recover from the code alone. We note that “thinking in pictures” here is external (a PNG buffer the model perceives via its visual encoder), not internal latent visual reasoning. Whether internal latent visual reasoning is achievable, and how it would compare to external rendering, is a question for future work.

**When to Think in Words vs. Pictures.** On this benchmark, tasks toward the spatial end (ST, MOT) are best served by V-CoT, tasks toward the symbolic end (CC) by T-CoT, and tasks blending both (SR, GR) by V-CoT with meaningful T-CoT gains as well (Figure 7).

**Recommendations for Practitioners.** (1) Avoid blindly adding “think step by step” to spatial reasoning prompts; on our controlled tasks it hurts ST (−16.5) and MOT (−12.7). (2) For spatial tasks, V-CoT requires both code generation and

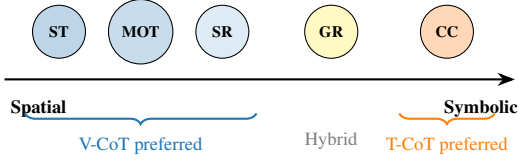


Figure 7: Taxonomy of VISCoT-DIAG categories along the spatial-symbolic axis. Tasks requiring spatial state maintenance favor V-CoT; tasks requiring symbolic decomposition favor T-CoT.

the rendered observation to deliver large gains; either alone is insufficient. (3) For compositional counting, T-CoT or ST-CoT is preferable. (4) Consider adaptive routing: a simple classifier achieves 78.2% versus 74.0% for V-CoT-everywhere on this benchmark while reducing average cost.

**Cost-Accuracy Tradeoff.** V-CoT consumes  $3.2\times$  more tokens than T-CoT and requires a second image encoder pass per turn. For ST and MOT, the 25- to 40-point improvement over T-CoT easily justifies the cost; for CC, it does not. V-CoT-NF removes the second encoder pass and is thus cheaper than V-CoT, but its substantially lower accuracy on the spatial categories means it is rarely the right default.

**Relation to Prior Multimodal Benchmarks.** Our findings are complementary to MMVU (Zhao et al., 2025) and MMMU-Pro (Yue et al., 2025), which document that direct answering and naive textual CoT struggle on complex multimodal tasks. We do not claim novelty on *whether* CoT struggles in multimodal settings; that is established. Our contribution is a controlled paradigm comparison (including a code-only ablation) on a deliberately simple, diagnostic distribution where the failure modes can be cleanly attributed and counted.

## 8 Limitations

Several scope conditions on our claims are worth stating explicitly.

**Task simplicity.** VISCoT-DIAG comprises relatively simple, controlled visual tasks (2D shapes, grid worlds, CLEVR-style scenes). This is intentional, since simple tasks support cleaner attribution of failure modes, but it means our findings should not be read as a general claim about visual reasoning in the wild. Tasks requiring deeper semantic reasoning, real-world visual complexity, or multi-disciplinary expert knowledge (as in MMVU

and MMMU-Pro) may show different patterns; we phrase our claims accordingly throughout.

**Code generation as confound.** V-CoT relies on the model’s code generation ability. Weaker code generators produce incorrect annotations that mislead reasoning. Our V-CoT-NF ablation partially decouples code generation from visual perception, but the absolute level of V-CoT performance still depends on code quality. Future work could provide oracle annotations to fully isolate the effect of visual feedback.

**2D, English-only, synthetic.** VISCoT-DIAG focuses on 2D reasoning with synthetic and semi-synthetic images and English prompts. Real-world photographs introduce occlusion, ambiguity, and lighting variation that may interact differently with reasoning modality. Languages with different spatial reference frames may also yield different patterns.

**External vs. internal visual reasoning.** V-CoT uses external code execution with a 5-turn limit. Whether similar benefits can be achieved through internal latent visual representations remains an open question.

**Failure mode taxonomy.** Our three failure modes (state collapse, transformation hallucination, tracking loss) are empirically grounded but not exhaustive. Other patterns may emerge with different task distributions or model architectures.

## 9 Conclusion

We presented a controlled diagnostic study of textual versus visual chain-of-thought reasoning in vision-language models on VISCoT-DIAG, a benchmark of 1,200 simple visual tasks. On this distribution, textual CoT degrades performance on spatial transformation ( $-16.5\%$ ) and multi-object tracking ( $-12.7\%$ ) relative to direct answering, while visual CoT yields gains up to  $+25.4\%$ . A code-only ablation (V-CoT-NF) shows that code generation alone recovers only 36% of V-CoT’s gain over T-CoT on spatial transformation: most of the benefit comes from the rendered image, not from delegating computation to a Python interpreter. We documented three failure modes (spatial state collapse, transformation hallucination, tracking loss) and showed that adaptive modality routing achieves 78.2% versus 74.0% for V-CoT-everywhere. We encourage practitioners to think of

reasoning modality as a per-task design choice on visual tasks rather than a universal default, while noting that our claims hold on this controlled distribution and may not transfer to richer real-world visual tasks.

## References

- Anthropic. 2026. System card: Claude Opus 4.6. <https://www.anthropic.com/claude-opus-4-6-system-card>.
- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, and 1 others. 2025. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*.
- Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. 2024. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*.
- Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024. BLINK: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision (ECCV)*.
- Gemini Team, Google DeepMind. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.
- Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. 2024. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *Advances in Neural Information Processing Systems*, volume 37.
- Drew A. Hudson and Christopher D. Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6700–6709.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2901–2910.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213.
- Stephen M. Kosslyn. 1995. Mental imagery. In Stephen M. Kosslyn and Daniel N. Osherson, editors, *An Invitation to Cognitive Science, Vol. 2: Visual Cognition*, pages 267–296. MIT Press.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*.
- Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, volume 35.
- Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. 2024. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- OpenAI. 2025. GPT-5 system card. *arXiv preprint arXiv:2601.03267*.
- Allan Paivio. 1991. Dual coding theory: Retrospect and current status. *Canadian Journal of Psychology*, 45(3):255–287.
- Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024. Visual CoT: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *Advances in Neural Information Processing Systems*, volume 37.
- Roger N. Shepard and Jacqueline Metzler. 1971. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703.
- Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, Hisham Cholakkal, Ivan Laptev, Mubarak Shah, Fahad Shahbaz Khan, and Salman H. Khan. 2025. LlamaV-o1: Rethinking step-by-step visual reasoning in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*.
- Yikun Wang, Siyin Wang, Qinyuan Cheng, Zhaoye Fei, Liang Ding, Qipeng Guo, Dacheng Tao, and Xipeng Qiu. 2025. VisuoThink: Empowering LVLm reasoning with multimodal tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 21707–21719.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

- Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024. Mind’s eye of LLMs: Visualization-of-thought elicits spatial reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 37.
- Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. 2025. LLaVA-CoT: Let vision language models reason step-by-step. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2087–2098.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. 2025. MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*.
- Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2024. Multi-modal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*.
- Yilun Zhao, Lujing Xie, Haowei Zhang, Guo Gan, Yitao Long, Zhiyuan Hu, Tongyan Hu, Weiyuan Chen, Chuhan Li, Junyang Song, Zhijian Xu, Chengye Wang, Weifeng Pan, Ziyao Shangguan, Xiangru Tang, Zhenwen Liang, Yixin Liu, Chen Zhao, and Arman Cohan. 2025. MMVU: Measuring expert-level multi-discipline video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

## A Prompt Templates

We provide the full prompt templates used for each CoT paradigm below. All prompts are preceded by the task-specific image, passed as the first content block in the multimodal message.

### Direct (D).

“Look at the image and answer the following question. Provide only the final answer with no explanation.\n\nQuestion: {question}\n\nAnswer:”

### Textual CoT (T-CoT).

“Look at the image and answer the following question. Think step by step, explaining your reasoning carefully before giving your final answer.\n\nQuestion: {question}\n\nLet me think through this step by step:”

### Structured Textual CoT (ST-CoT).

“Look at the image and answer the following question. Structure your response using the following format:\n\nSUMMARY: Briefly state what you need to determine.\nOBSERVATION: Describe the relevant visual content you see in the image.\nREASONING: Work through the problem step by step, explaining your logic.\nANSWER: State your final answer clearly.\n\nQuestion: {question}”

### Visual CoT, no feedback (V-CoT-NF).

“You have access to a Python environment with PIL/Pillow for image manipulation. The input image is loaded as `img`. To reason about this question, write Python code to annotate the image with helpful visual aids. **Note: the rendered output of your code will not be shown back to you. You must reason about what your code would produce and answer based on that mental simulation.**\n\nAvailable drawing primitives: `draw.rectangle()`, `draw.line()`, `draw.polygon()`, `draw.ellipse()`, `draw.text()`, and affine transformations via `img.rotate()` and `img.transpose()`.\n\nQuestion: {question}”

### Visual CoT (V-CoT).

“You have access to a Python environment with PIL/Pillow for image manipulation. The input image is loaded as `img`. To reason about this question, write Python code to annotate the image with helpful visual aids such as bounding boxes, arrows, highlighted regions, auxiliary lines, or transformed shapes. After each annotation step, the updated image will be shown to you. Observe the result and continue reasoning. When you are confident in your answer, state it clearly.\n\nAvailable drawing primitives: `draw.rectangle()`, `draw.line()`, `draw.polygon()`, `draw.ellipse()`, `draw.text()`, and affine transformations via `img.rotate()` and `img.transpose()`.\n\nQuestion: {question}\n\nBegin by examining the image and deciding what to draw first:”

For V-CoT, the multi-turn pipeline operates as follows: (1) the model generates a code block; (2) the code is executed in a sandboxed Python environment with the current image; (3) the resulting annotated image is re-encoded and sent back to the model; (4) the model generates either another code block or a final answer. We allow up to 5 annotation turns. V-CoT-NF uses the same pipeline except step (3) is omitted; the executed image is recorded for offline AQ scoring but is not returned to the model.

## B Benchmark Construction Details

**Spatial Relation (SR).** 120 instances from BLINK (Fu et al., 2024) spatial split + 120 from CLEVR (Johnson et al., 2017). Difficulty: Easy (2–3 objects), Medium (4–5), Hard (6+).

**Spatial Transformation (ST).** Procedurally generated 2D shapes with controlled transformation parameters (rotations of 90/180/270 degrees, horizontal/vertical reflections). Easy: one transformation. Medium: two. Hard: three or more. Ground truth computed analytically.

**Multi-Object Tracking (MOT).** 8×8 grid worlds. Easy: 3 objects, 2 moves. Medium: 4–5 objects, 3–4 moves. Hard: 6–7 objects, 5–6 moves. Ground truth from movement replay.

**Compositional Counting (CC).** CLEVR and GQA instances requiring multi-attribute filtering with spatial qualifiers. Counts 0–7. Difficulty by attribute filter count: Easy (1–2), Medium (3), Hard (4+).

**Geometric Reasoning (GR).** 120 from Math-Vista geometry plus 120 synthetic. Difficulty by reasoning chain length: Easy (1–2), Medium (3–4), Hard (5+).

All synthetic instances use deterministic random seeds. Human validation on 10% stratified sample (120 instances, 3 annotators): 96.7% agreement with ground truth, Fleiss’  $\kappa = 0.91$ .

## C Full Per-Model Results

Table 4 presents the complete per-model, per-category, per-paradigm breakdown.

## D Adaptive Routing Preliminary Experiment

Using a fine-tuned DistilBERT model with 5-fold cross-validation on the 1,200 instances:

Model	Paradigm	SR	ST	MOT	CC	GR	Avg.
GPT-5	Direct	70.2	53.6	62.4	59.1	51.8	59.4
	T-CoT	75.4	38.7	50.6	77.8	67.2	61.9
	ST-CoT	77.0	41.5	54.7	79.6	69.1	64.4
	V-CoT-NF	73.1	53.2	56.3	73.4	68.5	64.9
	V-CoT	<b>83.1</b>	<b>81.7</b>	<b>77.4</b>	72.3	<b>78.6</b>	<b>78.6</b>
Claude Opus 4.6	Direct	67.8	52.4	60.2	56.7	49.3	57.3
	T-CoT	73.6	36.2	47.8	75.4	65.1	59.6
	ST-CoT	75.2	39.4	52.1	77.3	67.0	62.2
	V-CoT-NF	70.8	51.7	54.6	71.2	65.7	62.8
	V-CoT	<b>80.4</b>	<b>78.9</b>	<b>75.0</b>	69.7	<b>75.8</b>	<b>75.9</b>
Gemini 2.5 Pro	Direct	65.3	51.8	58.7	55.2	47.1	55.6
	T-CoT	68.4	34.5	45.2	70.6	59.3	55.6
	ST-CoT	70.1	37.6	49.5	73.2	61.5	58.4
	V-CoT-NF	68.6	49.4	53.0	67.4	61.8	60.0
	V-CoT	<b>76.7</b>	<b>76.4</b>	<b>72.0</b>	67.4	<b>71.6</b>	<b>72.8</b>
Qwen3-VL-72B	Direct	60.9	49.1	54.5	52.4	43.8	52.1
	T-CoT	64.2	31.2	41.5	66.9	55.4	51.8
	ST-CoT	65.8	33.8	44.9	68.5	57.6	54.1
	V-CoT-NF	63.5	46.8	50.3	63.2	60.1	56.8
	V-CoT	<b>72.4</b>	<b>70.2</b>	<b>67.7</b>	64.0	<b>67.5</b>	<b>68.4</b>

Table 4: Full per-model results on VISCOT-DIAG. V-CoT achieves the highest average for every model. T-CoT degrades ST and MOT universally. V-CoT-NF sits between T-CoT and V-CoT on spatial categories, isolating the contribution of code generation from rendered visual feedback. Best per model-category (excluding CC) is **bolded**.

Strategy	Avg. Accuracy (%)
T-CoT everywhere	57.3
ST-CoT everywhere	59.8
V-CoT-NF everywhere	61.1
V-CoT everywhere	74.0
Adaptive router (T vs. V)	78.2
Oracle routing	81.5

Table 5: Adaptive routing results. The router achieves 78.2%, outperforming the best single paradigm (V-CoT, 74.0%) by 4.2 points.

The router achieves 84.7% routing accuracy. Oracle routing (always choosing the better paradigm per instance) achieves 81.5%. Our router recovers 78.2% of this bound. The 3.3-point gap suggests room for improvement, potentially through multimodal routing that considers the image as well as the question text. We did not include V-CoT-NF as a routing target because it is rarely the best paradigm for any instance type; future work could explore three-way routing including V-CoT-NF as a cost-aware fallback.

## E V-CoT and V-CoT-NF Annotation Examples

**High-quality (ST).** The model generates code to draw the original shape in solid fill, then draws the rotated version with a dashed outline and a

curved rotation arrow, labeling both states. Under V-CoT the model observes this rendering and answers correctly. Under V-CoT-NF the model wrote the same code but answered based on its expectation of the rendering; on this particular instance both succeeded, but accuracy on similar instances diverges (V-CoT 81.7% vs. V-CoT-NF 53.2% on ST for GPT-5).

**Medium-quality (GR).** The model draws auxiliary lines and labels angles. The spatial structure is interpretable. V-CoT and V-CoT-NF perform similarly on GR (78.6 vs. 68.5 for GPT-5), the smallest gap among spatial-leaning categories, consistent with GR benefiting more from the symbolic structure of the code than from the rendered visual itself.

**Low-quality (MOT).** The model draws bounding boxes around objects but fails to update positions after described movements. Under V-CoT this leads to incorrect answers; under V-CoT-NF the model never sees the (incorrect) render but still answers from its mental simulation, which exhibits the same tracking loss failure mode.

# A Zipfian Analysis of Visual Token Distributions for AI-Generated Images

Andrew Shin

Keio University

Yokohama, Kanagawa Prefecture, Japan

shin@ics.keio.ac.jp

## Abstract

The rapid evolution of text-to-image generation has blurred the perceptual boundary between natural and synthetic imagery. However, it remains questionable whether the statistical structure of generated visual content mirrors the information density of the physical visual world. Drawing upon principles from statistical linguistics, this study investigates the visual language of generative models through the lens of Zipfian dynamics. By analyzing a large-scale corpus of real and synthetic images, we uncover a fundamental divergence between visual syntax and semantics. We find that while generative models have successfully replicated the low-level physics of light, their high-level texture vocabulary exhibits distinct statistical signatures. Our analysis reveals a spectrum of entropy, identifying architectural fingerprints unique to each model. Furthermore, we investigate the relationship between generated images and prompt complexity, and find that increasing the semantic specificity of text prompts systematically degrades the statistical realism of the generated output.

## 1 Introduction

The field of computer vision has witnessed a paradigm shift with the advent of diffusion-based text-to-image generation. State-of-the-art image generation models have largely surmounted the previous hurdles of synthetic imagery, and the typical artifacts such as malformed hands, incoherent lighting, or surreal geometry are vanishing at a rapid pace. As these systems achieve photorealistic fidelity, the perceptual boundary between natural photography and synthetic media is eroding, rendering traditional, artifact-based forensic methods increasingly obsolete. However, the ability to clone the *appearance* of reality does not necessarily equate to replicating the underlying *stochastic processes* that govern the natural world.

This divergence prompts a fundamental question: does the generative process leave an invisible statistical fingerprint? While a generated image may be indistinguishable from a photograph to the human eye, it is ultimately the product of a mathematical optimization process of denoising trajectory guided by a text encoder, rather than a physical capture of photons interacting with matter. We posit that natural images, like natural language, possess a hierarchical statistical structure governed by universal laws of information density and entropy, as has been suggested by the pioneering works on analyzing statistical behavior of visual data with linguistic laws (Ruderman, 1996; Crosier and Griffin, 2007; Chan et al., 2024; Tsai et al., 2025), which directly inspire our work in this paper. If generative models are fundamentally statistical mimics, they may inadvertently suppress the chaotic, heavy-tailed entropy characteristic of the physical world in favor of cleaner, more probable visual patterns.

In this paper, we use these linguistic principles to audit the visual language of generative AI. We analyze a large-scale corpus of 50,000 images, stratified across real photography and state-of-the-art generative models. We employ a dual-tokenization strategy, separating visual physics (low-level edges) from texture vocabulary (high-level visual words).

Our contributions are threefold. 1) We demonstrate a dissociation between syntax and semantics: while AI models have effectively solved the Zipfian statistics of low-level features, they diverge significantly in their texture vocabulary, following log-normal rather than pure power law dynamics. 2) We also identify a spectrum of entropy that fingerprints model architecture, ranging from the artificially ordered, low-entropy distributions to the real, chaotic statistics. 3) We uncover its relevance to prompt complexity, providing empirical evidence that increasing the semantic specificity of a text prompt acts as an entropy constraint, degrading the statistical realism of the generated output.

## 2 Related Work

The inquiry into the statistical structure of natural scenes was pioneered by (Ruderman, 1996), who proposed that the scale-invariance of natural images arises because the visual world is a collage of statistically independent objects following a power-law size distribution. Building on this foundation, (Crosier and Griffin, 2007) formalized the analogy between vision and language. They defined an basic image features (BIFs), classifying pixels into geometric primitives like edges and blobs, and treated  $n \times n$  patches of these features as visual words. They demonstrated that, for specific parameter settings, these visual words strictly obey Zipf’s Law (Zipf, 1932), arguing that this distribution represents an optimally efficient code for object recognition, analogous to the efficiency of natural language.

Recent research has extended these findings from hand-crafted features to the learned representations of deep neural networks. (Tsai et al., 2025) utilized the kernels of pre-trained CNNs such as VGG-19 (Simonyan and Zisserman, 2014) to define visual words based on activation intensity. By analyzing layer-wise statistics, they confirmed that not only Zipf’s Law but also Heaps’ Law (Heaps, 1978) with vocabulary growth and Benford’s Law (Benford, 1938) with leading digit distribution also emerge in the visual words. Their findings suggest a correspondence between the evolution of visual symbols in deep networks and the structure of human language.

With the advent of autoregressive vision transformers, (Chan et al., 2024) provided a comprehensive audit of discrete visual languages, such as tokens in LLaVA (Liu et al., 2023) and Chameleon (Team et al., 2024). While they confirmed that discrete visual tokens follow Zipfian distributions, they identified critical structural differences from NLP: visual languages exhibit higher perplexity, weaker hierarchical grammar, and a tendency for tokens to represent intermediate granularity such as object parts, rather than semantic wholes. Crucially, their work focused on the properties of the internal representation of real images, whereas our work investigates the statistical realism of the generative output produced by diffusion models.

## 3 Methodology

To compare the statistical properties of real vs. AI imagery, we employ a dual-tokenization strategy,

analyzing both low-level features (physics) and high-level textures (semantics).

### 3.1 Feature Extraction Strategy

**Basic Image Features (BIFs):** To capture the low-level structure of the image (edges, blobs, flat regions), we compute BIF maps. Following (Crosier and Griffin, 2007), we compute the response of the image  $I$  to a bank of Gaussian derivative filters up to the second order at scale  $\sigma$ :

$$s_{nm} = \sigma^{n+m} \cdot \left( \frac{\partial^{n+m}}{\partial x^n \partial y^m} G_\sigma \right) * I \quad (1)$$

Pixels are classified into one of 7 symmetry classes (flat, slope, dark blob, light blob, dark line, light line, saddle) based on the invariant properties of the local jet. This tokenization represents the "syntax" of the visual world, i.e. the fundamental rules of geometry and contrast.

**Vector Quantization (VQ):** To capture complex textures (e.g., fur, grass, fabric), we employ vector quantization. We extract non-overlapping  $4 \times 4$  pixel patches  $p_i$  from all images. We learn a shared vocabulary  $V = \{v_1, \dots, v_K\}$  of size  $K = 512$  using mini-batch K-means on a mixed sample of real and synthetic patches to establish a statistically neutral, unified basis for comparison. Each patch is then assigned to its nearest visual word:

$$w_i = \arg \min_k \|p_i - v_k\|_2^2 \quad (2)$$

This creates a discrete "document" of visual words for every image, allowing us to analyze the diversity and repetition of textures.

### 3.2 Statistical Analysis

We fit the frequency distribution of tokens to a discrete power law distribution:

$$P(x) = Cx^{-\alpha}, \quad x \geq x_{\min} \quad (3)$$

where  $\alpha$  is the scaling parameter and  $x_{\min}$  is the lower bound of the power-law behavior. We use the Kolmogorov-Smirnov distance to estimate  $x_{\min}$  and maximum likelihood estimation for  $\alpha$ . To confirm the validity of the fit, we compare it against a log-normal distribution via a likelihood ratio test ( $\mathcal{R}$ ), where  $\mathcal{R} > 0$  supports the power law and  $\mathcal{R} < 0$  supports log-normal.

## 4 Experiments

### 4.1 Dataset

We utilize the Rapidata Image Generation Alignment Dataset (Rapidata, 2025) to source 40,000 AI-generated images, stratified equally (10,000 each)

Table 1: Aggregate analysis ( $N = 50,000$ ). For BIFs, AI is indistinguishable from reality ( $p > 0.05$ ). For VQ, both distributions are log-normal ( $R < 0$ ), but AI exhibits a steeper slope and lower  $x_{\min}$ .

Dataset	Alpha ( $\alpha$ )	$x_{\min}$	$\mathcal{R}$	$p$ -val
<i>BIF</i>				
Real (Aggregate)	1.8180	34.9M	0.002	0.70
AI (Aggregate)	1.7608	32.8M	0.003	0.85
<i>VQ</i>				
Real (Aggregate)	1.9490	1.93M	-13.52	< 0.01
AI (Aggregate)	2.2868	1.42M	-6.60	< 0.01

across four state-of-the-art models, namely DALL-E 3 (Betker et al.), Midjourney (Midjourney, 2024), Stable Diffusion 3 (SD3) (Esser et al., 2024), and Flux 1.1 Pro (Labs et al., 2025). For the real-world baseline, we sample 10,000 images from ImageNet-1k (Russakovsky et al., 2015) validation set, representing a general purpose distribution of natural photography. Feature extraction and statistical analysis described in Section 3 are then applied to the collected data.

## 4.2 Results

### 4.2.1 Aggregate Analysis

Table 1 presents a fundamental dichotomy in the statistical footprint of generative models: a near-perfect replication of low-level visual syntax, contrasted against a systematic deviation in high-level texture semantics.

The analysis of BIFs reveals that, at the level of local geometry, AI-generated imagery is statistically indistinguishable from natural photography. The high  $p$ -values and negligible likelihood ratios suggest that the distribution of fundamental geometric primitives is invariant across real and synthetic domains.

In contrast, the visual vocabulary analysis with VQ uncovers a significant distributional shift. Consistently negative likelihood ratios for both datasets indicate that visual word frequencies follow a log-normal distribution rather than a pure power law. AI aggregate exhibits a steeper decay, which may be attributed to suppression of entropy, where the generative process under-samples the rare, chaotic textures that populate the "long tail" of reality.

Furthermore, the difference in  $x_{\min}$ , which marks the transition point between head and tail of the distribution, is diagnostic. Real images maintain a significantly higher cutoff, implying an increased density at the upper tail. Nature sustains a larger diversity of common textures before the

Table 2: Model-specific analysis. All models match real-world images in BIF with  $p \approx 0.9$ . With VQ, however, models display distinct behaviors.

Model	Alpha ( $\alpha$ )	$x_{\min}$	$\mathcal{R}$	$p$ -val
<i>BIF Analysis (Physics)</i>				
Stable Diffusion 3	1.8109	7.89M	0.001	0.94
DALL-E 3	1.8026	9.32M	0.001	0.95
Flux 1.1 Pro	1.7857	6.91M	0.001	0.89
Midjourney	1.7318	5.62M	0.001	0.93
<i>VQ Analysis (Texture)</i>				
DALL-E 3	2.4388	391k	-13.95	< 0.01
Stable Diffusion 3	2.2909	258k	-6.10	< 0.01
Midjourney	2.2513	379k	-17.45	< 0.01
Flux 1.1 Pro	2.1741	246k	-18.54	< 0.01

distribution transitions into its decaying tail behavior. AI models, by contrast, transition to the decay phase earlier, collapsing the diversity of visual textures.

### 4.2.2 Model-Specific Analysis

Breaking down the aggregate results reveals distinct statistical fingerprints for each architecture (Table 2). While all models converge on the physics of low-level features, they diverge significantly in how they manage the entropy of high-level textures, which may be linked to architectural choices.

DALL-E 3 exhibits the steepest Zipfian slope, indicating the least entropic texture distribution. This observation is consistent with DALL-E 3’s integration of LLM guidance for prompt rewriting. We hypothesize that minimizing the semantic gap between text and image may inadvertently act to reduce the stochastic variance of the output and suppress the long tail of messy, unprompted textures in favor of high-probability visual concepts.

Stable Diffusion and Midjourney occupy a middle ground, likely driven by distinct optimization constraints. For Stable Diffusion, the reliance on latent diffusion involves compressing data into a lower-dimensional latent space via a VAE. This process may have acted as a low-pass filter, discarding the high-frequency Zipfian tail of pixel noise to ensure stability. Midjourney’s similarity may have stemmed from aggressive RLHF tuned for aesthetics, which biases the model toward clean and statistically regular textures, while penalizing the chaotic aspects of the physical world.

Flux exhibits the lowest slope among the AI models, placing it statistically closest to the high-entropy nature of real-world data. This could potentially be attributed to its *flow matching* paradigm. Unlike standard diffusion, which iteratively re-

Table 3: Impact of prompt complexity. Specificity increases the Zipfian slope, reducing statistical realism.

Prompt Type	Alpha ( $\alpha$ )	$x_{\min}$
Simple (< 6 words)	1.9283	132,245
Complex (> 15 words)	2.3259	197,883

moves noise and may smooth out high-frequency irregularities, flow matching models optimize transport paths, which may allow the model to preserve a higher degree of high-frequency variance, resulting in a texture distribution that feels more physically plausible.

While no model perfectly replicates the heavy-tailed entropy of natural photography, the data suggests a trade-off; stronger semantic guidance appears to correlate with reduced statistical realism, whereas newer flow-based architectures seem better equipped to sustain the chaotic vocabulary of the visual world.

## 5 The Effect of Prompt Complexity

Images generated by text-to-image generation models are naturally influenced by the input prompts. A prevalent assumption in prompt engineering is that increasing the lexical density of a prompt enhances the realism of the generated output. We investigate this hypothesis by analyzing whether conditioning on complex, lengthy prompts increases the statistical diversity of the generated visual vocabulary, or whether it inadvertently constrains the model’s latent space to a narrower manifold of learned visual concepts.

### 5.1 Experimental Setting and Procedure

We stratified the data into two distinct groups based on prompt word count:

- **Simple Prompts (< 6 words):** These prompts typically consist of a single subject or short phrase (e.g., "A red car," "A cat"). They provide minimal semantic constraints, requiring the model to "hallucinate" the majority of the scene’s details, such as background, lighting, style, from its unconditioned prior.
- **Complex Prompts (> 15 words):** These prompts are highly descriptive, often specifying multiple attributes such as lighting conditions, artistic style, texture quality, and background elements (e.g., "A futuristic city skyline at sunset with neon lights reflecting on wet pavement, 4k, photorealistic").

We collected a balanced sample of  $N = 5,000$  images per group. To ensure a unified basis for comparison, we trained a shared VQ codebook on the combined dataset. We then tokenized all images into visual word sequences and performed maximum likelihood estimation to determine the Zipfian parameters for each group.

### 5.2 Results and Analysis

Table 3 presents the results of the distributional analysis.

The Zipfian slope for complex prompts is steeper than for simple prompts, indicating a faster decay in the frequency of rare events. This suggests that when the model is heavily conditioned by a long text string, it relies more heavily on a core set of high-probability visual tokens to satisfy the semantic constraints, suppressing the generation of rare, stochastic textures that characterize the heavy tail of natural imagery. On the other hand, conditioning on simple prompts allows the model to sample more freely from its diverse training distribution. In addition, the complex condition exhibits a significantly higher  $x_{\min}$ , implying that the distribution for complex prompts is dominated by a larger block of repetitive, common visual words before the Zipfian tail behavior emerges.

In summary, contrary to the intuition that more text leads to more visual information, we observe a systematic degradation in statistical realism as prompt complexity increases, and specificity in language appears to constrain the entropy of the visual generation process.

## 6 Conclusion

In this study, we examined the statistical behavior of AI-generated images with Zipfian approach. Our results demonstrate that, while generative AI has effectively mastered the low-level physics, it exhibits a distinct statistical fingerprint in its high-level texture vocabulary, whether in aggregate or with distinct models. We also observe that architectural choices of each model may be linked to their statistical behaviors. Finally, we examined the effect of prompt complexity in the statistical behavior of generated images, and find that, contrary to the popular intuition, increasing prompt complexity acts as an entropy constraint, driving the output distribution further from the heavy-tailed diversity of the physical world.

## Limitations

Our study faces several methodological constraints. First, our visual word tokenization relies on shallow K-Means clustering of raw pixel patches. While this heuristic aligns with classical vision literature, it captures local texture statistics rather than the high-level semantic tokens utilized by modern latent transformers. Consequently, our findings reflect the entropy of surface-level texture rather than deep semantic structure. Additionally, our analysis was conducted at a fixed scale ( $\sigma = 1.0$ ), potentially overlooking scale-invariant statistical divergences that may exist at macro-compositional levels or sub-pixel resolutions.

Second, our dataset choice introduces a potential selection bias, as the images were generated to maximize aesthetic alignment rather than representing the raw, rejection-free output of the models. Furthermore, our investigation of the prompt complexity relied solely on word count as a proxy for semantic density.

## References

- Frank Benford. 1938. The law of anomalous numbers. *Proceedings of the American Mathematical Society*.
- James Betker, Gabriel Goh, Li Jing, † TimBrooks, Jianfeng Wang, Linjie Li, † LongOuyang, † Jun-tangZhuang, † JoyceLee, † YufeiGuo, † Wesam-Manassra, † PrafullaDhariwal, † CaseyChu, † YunxinJiao, and Aditya Ramesh. [Improving image generation with better captions](#).
- David M. Chan, Rodolfo Corona, Joonyong Park, Cheol Jun Cho, Yutong Bai, and Trevor Darrell. 2024. [Analyzing the language of visual tokens](#). *ArXiv*, abs/2411.05001.
- Michael Crosier and Lewis D. Griffin. 2007. [Zipf’s law in image coding schemes](#). In *British Machine Vision Conference*.
- Patrick Esser, Sumith Kulal, A. Blattmann, Rahim Entezari, Jonas Muller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. 2024. [Scaling rectified flow transformers for high-resolution image synthesis](#). *ArXiv*, abs/2403.03206.
- H. S. Heaps. 1978. [Information retrieval, computational and theoretical aspects](#).
- Black Forest Labs, Stephen Batifol, A. Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Muller, Dustin Podell, Robin Rombach, Harry Saini, and 2 others. 2025. [Flux.1 kontext: Flow matching for in-context image generation and editing in latent space](#). *ArXiv*, abs/2506.15742.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. [Visual instruction tuning](#). *ArXiv*, abs/2304.08485.
- Midjourney. 2024. Midjourney. <https://www.midjourney.com>. Text-to-image generative AI model.
- Rapidata. 2025. Rapidata image generation alignment dataset. <https://huggingface.co/datasets/Rapidata/human-alignment-preferences-images>.
- Daniel L. Ruderman. 1996. [Origins of scaling in natural images](#). *Vision Research*, 37:3385–3398.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. [ImageNet Large Scale Visual Recognition Challenge](#). *International Journal of Computer Vision (IJCV)*, 115(3):211–252.
- Karen Simonyan and Andrew Zisserman. 2014. [Very deep convolutional networks for large-scale image recognition](#). *CoRR*, abs/1409.1556.
- Chameleon Team, Mingda Chen, Jacob Kahn, and Shang-Wen Li. 2024. [Chameleon: Mixed-modal early-fusion foundation models](#). *ArXiv*, abs/2405.09818.
- Ping-Rui Tsai, Chi hsiang Wang, Yu-Cheng Liao, and Tzay-Ming Hong. 2025. [Three laws of statistical linguistics emerging in images](#). *ArXiv*, abs/2501.18620.
- George Kingsley Zipf. 1932. [Selected Studies of the Principle of Relative Frequency in Language](#). Harvard University Press, Cambridge, MA.

# Semantically Aware Optimal Transport for Dense Label Transfer

Preeti and Kiran Ravish and Ankita Kushwaha and Pawan Kumar

IIIT Hyderabad, India

{preeti.preeti, kiran.ravish, ankita.kushwaha}@research.iiit.ac.in

pawan.kumar@iiit.ac.in

## Abstract

Vision foundation models produce features that generalize across visual domains without fine-tuning, yet naively transferring labels through these feature spaces fails under large distribution shifts. We propose SAOT (Semantically Aware Optimal Transport), which learns a transport cost within a fused unbalanced optimal transport formulation for dense label transfer from frozen vision transformer features to new domains. SAOT combines a learnable appearance metric with semantic class-prototype priors, unbalanced transport for partial matching under distribution shift, and a block-sparse solver for tractable inference. We pair this with a two-stage decoder: an MLP trained on SAOT pseudo-labels, then refined via EMA-teacher self-training with class-balanced sampling. On GTA5→Cityscapes with frozen DINOv2 ViT-L/14 features, SAOT+Decoder reaches 25.7% mIoU, a  $3.8\times$  improvement over nearest-neighbor transfer (6.7%), without any backbone adaptation. Per-class results show large gains on spatially coherent classes (road 90.3%, car 76.2%, building 71.5%), suggesting that learned semantic transport costs capture domain-invariant structure even under severe synthetic-to-real shifts. On VOC train→val with frozen ViT-B/16 features, the full pipeline reaches 47.5% mIoU, indicating that the approach extends beyond synthetic-to-real adaptation.

## 1 Introduction

Optimal transport (OT) compares and aligns distributions through cost-minimizing couplings (Vilani, 2016; Peyré and Cuturi, 2019). Entropic regularization yields a smooth discrete OT problem that can be solved efficiently by Sinkhorn scaling (Knopp and Sinkhorn, 1967; Cuturi, 2013; Altschuler et al., 2017), and such solvers have been used for semantic correspondence (Liu et al., 2020) and differentiable assignment (Sarlin et al., 2020).

*Dense label transfer*, propagating a semantic label field from a labeled source domain to an unlabeled target, is a core task in unsupervised domain adaptation (UDA) for segmentation. In the synthetic-to-real setting, a model trained on GTA5 (Richter et al., 2016) must segment real Cityscapes (Cordts et al., 2016) images. Many OT-based label-transfer baselines use *fixed* ground costs (e.g., cosine distance in frozen features), which can misalign semantics when appearance varies across domains.

Dense label transfer across domains faces three problems: (1) **Semantic misalignment**: fixed feature distances can match visually similar but semantically different regions (synthetic road textures mapped to real sidewalks); (2) **Mass imbalance**: class frequencies differ between source and target, so forcing all source mass to match target tokens can produce spurious correspondences; (3) **Scalability**: vision transformers produce thousands of patch tokens per image, so the solver must be efficient.

To address these issues, we propose SAOT, which *learns a semantically aware transport cost directly in feature space*. The inner OT problem remains well-posed through entropic regularization and unbalanced marginal relaxation. The cost combines a learnable appearance metric with semantic priors derived from class prototypes. A block-sparse solver restricts transport to  $k$ -nearest candidate neighborhoods for tractable inference. We then train a residual MLP decoder on SAOT pseudo-labels and refine it with EMA-teacher self-training (Tarvainen and Valpola, 2017).

## Contributions.

- A learnable ground cost  $c_\theta$  that blends appearance distance with class-prototype semantic priors for matching under domain shift (Section 4.1).

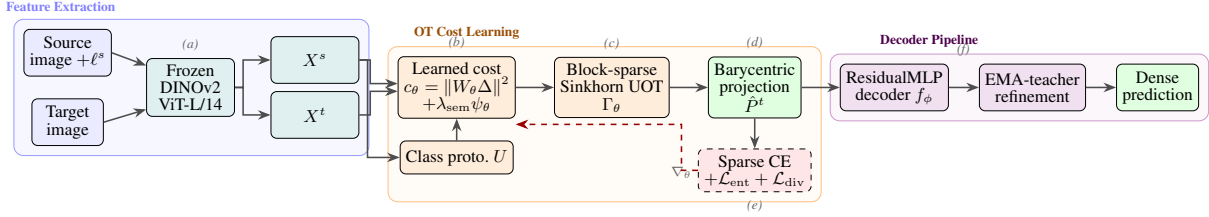


Figure 1: Overview of the SAOT framework. **(a)** Frozen DINOv2 ViT-L/14 extracts source/target patch tokens. **(b)** Learnable cost  $c_\theta$  blends appearance distance with semantic class-prototype priors. **(c)** Block-sparse unbalanced Sinkhorn solver computes coupling  $\Gamma_\theta$ . **(d)** Barycentric projection transfers source labels to target. **(e)** Sparse supervision + auxiliary losses drive bilevel cost learning (dashed red = gradient). **(f)** Two-stage decoder distills OT pseudo-labels, refined via EMA self-training.

- A fused unbalanced OT objective with entropic regularization, KL-relaxed marginals, and entropy/diversity auxiliary losses (Section 4.2).
- A  $k$ -NN block-sparse solver that reduces the coupling from  $O(nm)$  to  $O(nk)$  for high-resolution inference (Section 4.3).
- A two-stage decoder trained on SAOT pseudo-labels and refined with EMA-teacher self-training (Section 4.5).
- Within the class of frozen-backbone sparse point-supervised dense label-transfer methods using frozen foundation-model features, evaluation on GTA5→Cityscapes at  $504 \times 1008$  resolution with frozen DINOv2 ViT-L/14 features yields 25.7% mIoU, a  $3.8 \times$  gain over nearest-neighbor baselines, together with ablations and per-class analysis (Section 6).
- We validate the method on a second benchmark, VOC train→val with frozen ViT-B/16 features, where the full pipeline reaches 47.5% mIoU, showing that the approach extends beyond synthetic-to-real transfer.

## 2 Related Work

**Entropic OT and differentiable transport.** Entropic regularization makes OT strictly convex and solvable via Sinkhorn scaling (Cuturi, 2013), yielding GPU-friendly solvers and differentiable assignment layers (Sarlin et al., 2020; Mena et al., 2018). We build on these properties for dense label transfer with learned costs.

**Unbalanced and robust OT.** Classical OT assumes mass conservation, but occlusions and label shift violate this in dense transfer. Unbalanced OT relaxes marginals via KL divergences (Chizat et al., 2018a,b; Liero et al., 2018), and RWOT adds reliability weighting (Xu et al., 2020). We use the

unbalanced formulation as an outlier filter to avoid forced mismatches.

**Structure-aware OT: GW, FGW, and fused variants.** GW distances align relational structures (Mémoli, 2011), while FGW combines feature and structure terms (Titouan et al., 2019; Vayer et al., 2020). FUGW adds unbalanced relaxations (Thual et al., 2022), and ASOT uses unbalanced OT for temporally consistent segmentation (Xu and Gould, 2024). Our formulation is compatible with spatial-structure fusion on dense vision grids, while the reported experiments focus on learned semantic and appearance costs.

**OT and graph-based methods for domain adaptation.** OT is widely used in DA (Courty et al., 2017b,a; Damodaran et al., 2018; Seguy et al., 2018). These methods learn or adapt representations, mappings, or predictors; we instead learn the *ground cost* while keeping foundation model features frozen. Label propagation (Zhu et al., 2003) transfers labels via diffusion on a  $k$ -NN affinity graph; it is a natural transductive baseline but does not outperform simple  $k$ -NN voting when features are frozen (Table 1).

**UDA for semantic segmentation.** Modern UDA methods pair self-training with strong architectures: DAFormer (Hoyer et al., 2022a) (68.3% mIoU on GTA5→Cityscapes) uses a Transformer encoder with a context-aware decoder, HRDA (Hoyer et al., 2022b) (73.8%) adds multi-resolution context, and MIC (Hoyer et al., 2023) (75.9%) uses masked consistency. All adapt the segmentation network using labeled source and mostly unlabeled target images. Our approach works on frozen features and requires no backbone training.

**Vision foundation models for segmentation.** Self-supervised ViTs (Caron et al., 2021; Oquab

et al., 2024; Dosovitskiy et al., 2021) produce semantically rich patch tokens. DINOv2 (Oquab et al., 2024) features transfer across domains without fine-tuning. Rein (Wei et al., 2024) adapts DINOv2 through parameter-efficient fine-tuning for domain generalization, reaching 78.4STEGO (Hamilton et al., 2022) distills DINO feature correspondences into a segmentation head without any labels, reaching  $\sim 21$ We use frozen DINOv2 features as input to our transport framework; the method is complementary to adapter-based approaches like Rein, whose adapted features could serve as an alternative backbone.

### 3 Notation and Preliminaries

**Setup.** Let a source image be represented by  $n$  patch tokens  $X^s = \{x_i^s\}_{i=1}^n \in \mathbb{R}^{n \times d}$  with labels  $\ell_i^s \in \{1, \dots, K\}$ , and a target image by  $m$  tokens  $X^t = \{x_j^t\}_{j=1}^m$ . We seek a coupling  $\Gamma \in \mathbb{R}_+^{n \times m}$  transporting label mass from source to target such that correspondences are semantically meaningful.

**Entropic unbalanced OT.** Given nonneg. weights  $\mathbf{a} \in \mathbb{R}_+^n$ ,  $\mathbf{b} \in \mathbb{R}_+^m$  and cost  $C \in \mathbb{R}^{n \times m}$ , the entropic UOT objective (Chizat et al., 2018a) is:

$$\Gamma^* \in \arg \min_{\Gamma \geq 0} \langle \Gamma, C \rangle + \varepsilon \text{KL}(\Gamma \| \mathbf{a} \mathbf{b}^\top) + \tau (\text{KL}(\Gamma \mathbf{1} \| \mathbf{a}) + \text{KL}(\Gamma^\top \mathbf{1} \| \mathbf{b})). \quad (1)$$

The entropic term makes the objective strictly convex in  $\Gamma$ ; the KL marginal penalties relax mass conservation to allow partial matchings.

**Barycentric label transfer.** Given coupling with positive target column masses  $\Gamma^*$  and one-hot source labels  $P^s \in \{0, 1\}^{n \times K}$ , predicted target label probabilities are:

$$\hat{P}^t = \text{diag}((\Gamma^*)^\top \mathbf{1})^{-1} (\Gamma^*)^\top P^s. \quad (2)$$

## 4 Method: SAOT

SAOT learns a semantically aware OT cost for dense label transfer via bilevel optimization. Fig. 1 shows the pipeline.

### 4.1 Semantically Aware Ground Cost

We define the cost between a labeled source token  $(x^s, \ell^s)$  and target token  $x^t$ :

$$c_\theta(x^s, \ell^s; x^t) = \underbrace{\|W_\theta x^s - W_\theta x^t\|_2^2}_{\text{learned appearance}} + \lambda_{\text{sem}} \underbrace{\psi_\theta(\ell^s; x^t)}_{\text{semantic prior}} \quad (3)$$

---

### Algorithm 1 Bilevel training of SAOT

---

**Require:** Source tokens  $\{(X_i^s, \ell_i^s)\}$ , target tokens  $\{X_j^t\}$ , sparse target point labels  $\{(\ell_j^t)\}_{j \in \mathcal{S}}$

**Require:** Hyperparams  $(\varepsilon, \tau, k, \lambda_{\text{sem}}, \lambda_{\text{ent}}, \lambda_{\text{div}})$

```

1: while not converged do
2:   Sample source–target pair  $(X^s, \ell^s, X^t)$ 
3:   Compute class prototypes  $U$  from source labels
4:   Build  $k$ -NN candidate sets  $\{\mathcal{N}_k(i)\}_{i=1}^n$ 
5:   Compute block-sparse cost  $C_\theta$  using (3)
6:   Solve UOT:  $\Gamma_\theta \leftarrow \text{SPARSE-SINKHORN-UOT}(C_\theta, \mathbf{a}, \mathbf{b}; \varepsilon, \tau)$ 
7:   Predict labels  $\hat{P}^t$  via barycentric projection (2)
8:    $\mathcal{L} \leftarrow \sum_{j \in \mathcal{S}} \text{CE}(\hat{P}_{j,:}^t, \ell_j^t) + \lambda_{\text{ent}} \mathcal{L}_{\text{ent}} + \lambda_{\text{div}} \mathcal{L}_{\text{div}}$ 
9:   Update  $\theta$  by gradient descent on  $\mathcal{L}$ 
10: end while

```

---

where  $W_\theta$  is a learnable linear projection and the semantic prior penalizes transporting mass to semantically implausible targets.

**Semantic prior via class prototypes.** Let  $u_c \in \mathbb{R}^d$  be the mean DINOv2 embedding for source class  $c$ :  $u_c = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} x_i^s$ . The semantic prior is:

$$\psi_\theta(\ell^s; x^t) = -\log \pi_\theta(x^t)_{\ell^s},$$

$$\pi_\theta(x^t) = \text{softmax}\left(\frac{1}{T} U^\top V_\theta x^t\right), \quad (4)$$

where  $U = [u_1 \dots u_K] \in \mathbb{R}^{d \times K}$  and  $V_\theta$  is a learnable projector. This penalizes matching a source token to a target location whose features are far from that source class’s prototype.

### 4.2 Fused Unbalanced Objective

The inner transport problem combines the learned cost with auxiliary regularizers:

$$\Gamma_\theta^* \in \arg \min_{\Gamma \geq 0} \langle \Gamma, C_\theta \rangle + \varepsilon \text{KL}(\Gamma \| \mathbf{a} \mathbf{b}^\top) + \tau (\text{KL}(\Gamma \mathbf{1} \| \mathbf{a}) + \text{KL}(\Gamma^\top \mathbf{1} \| \mathbf{b})). \quad (5)$$

Class distributions differ between source and target. Balanced OT forces all source mass onto target tokens, producing meaningless correspondences. KL-relaxed marginals ( $\tau < \infty$ ) let the solver leave unmatched mass unassigned. In our experiments, balanced OT scores 0% mIoU while UOT scores 6.7% even with fixed costs (Table 5 in Section 6.4).

Since most target tokens lack supervision, we add two auxiliary losses for dense gradient signal:

$$\mathcal{L}_{\text{ent}} = -\frac{1}{m} \sum_j \sum_c \hat{P}_{j,c}^t \log \hat{P}_{j,c}^t, \quad (6)$$

$$\mathcal{L}_{\text{div}} = \text{KL}\left(\hat{P}^t \left\| \frac{1}{K} \mathbf{1}\right.\right), \quad (7)$$

Table 1: Main results on GTA5→Cityscapes (mIoU %, 500 val images). All methods use frozen DINOv2 ViT-L/14 features.

Method	mIoU (%)	Time (s)
LinearProbe	6.2	0.9
ProtoNet	6.3	0.9
OT ( $\varepsilon=0.1$ , bal., $k=128$ )	6.5	9.9
NN ( $k=1$ )	6.7	4.3
UOT ( $\varepsilon=0.1$ , $\tau=0.3$ )	6.7	6.4
RWOT ( $\varepsilon=0.1$ , $\tau=0.3$ )	6.8	7.5
KNN-Global ( $k=10$ )	7.5	10.7
LabelProp ( $k=10$ )	7.5	508.3
SAOT (single source)	12.6	11.1
SAOT (multi-source, $K=3$ )	15.8	32.7
<b>SAOT+Decoder</b>	<b>25.7</b>	<b>1.2</b>

where  $\bar{P}^t = \frac{1}{m} \sum_j \hat{P}_{j,:}^t$  is the mean prediction. Entropy minimization ( $\mathcal{L}_{\text{ent}}$ ) encourages confident predictions; diversity ( $\mathcal{L}_{\text{div}}$ ) prevents class collapse. Both are standard in UDA (Vu et al., 2019).

### 4.3 Block-Sparse Solver

A dense coupling  $\Gamma \in \mathbb{R}^{n \times m}$  is too large for high-resolution grids. With DINOv2 ViT-L/14 at  $504 \times 1008$  resolution, each image produces  $n = 36 \times 72 = 2,592$  tokens; a full coupling requires  $n \times m \approx 6.7\text{M}$  entries per image pair.

For each source token  $i$ , we compute cosine similarity to all target tokens and retain the top- $k$  neighbors  $\mathcal{N}_k(i) \subset \{1, \dots, m\}$ . This reduces the coupling to  $O(nk)$  entries. We use  $k = 128$  in all experiments.

Within the active support, we run log-domain Sinkhorn iterations for UOT (Chizat et al., 2018a):

$$\begin{aligned} u_i &\leftarrow \frac{\tau}{\tau + \varepsilon} \left( \log a_i - \log \sum_{j \in \mathcal{N}_k(i)} K_{ij} e^{v_j} \right), \\ v_j &\leftarrow \frac{\tau}{\tau + \varepsilon} \left( \log b_j - \log \sum_{i: j \in \mathcal{N}_k(i)} K_{ij} e^{u_i} \right), \end{aligned} \quad (8)$$

where  $K_{ij} = \exp(-C_{\theta}^{ij}/\varepsilon)$  on active entries. The  $\tau/(\tau + \varepsilon)$  damping factor implements KL-relaxed marginals.

### 4.4 End-to-End Cost Learning

We optimize the cost parameters  $\theta$  via bilevel optimization. Given sparse target point labels  $\{(\ell_j^t)\}_{j \in \mathcal{S}}$  (class-balanced point supervision), the outer objective is:

$$\min_{\theta} \sum_{j \in \mathcal{S}} \text{CE}(\hat{P}_{j,:}^t, \ell_j^t) + \lambda_{\text{ent}} \mathcal{L}_{\text{ent}} + \lambda_{\text{div}} \mathcal{L}_{\text{div}}. \quad (9)$$

We differentiate through the Sinkhorn solver via implicit differentiation of the fixed-point equations. Training uses AdamW with learning rate  $5 \times 10^{-4}$  for 14,000 steps, sampling random source–target pairs at each iteration. Algorithm 1 summarizes the complete training procedure.

### 4.5 Decoder Training and EMA Refinement

Raw SAOT predictions are computed per source–target pair and need access to source images at inference. We distill them into a parametric decoder.

**Stage 1: Pseudo-label distillation.** For each target training image, we compute soft pseudo-labels  $\hat{P}^t$  by aggregating SAOT predictions from the top-5 most similar source images (weighted by global feature similarity). A residual MLP decoder  $f_{\phi}: \mathbb{R}^d \rightarrow \mathbb{R}^K$  is then trained on the combination of:

- **Source:** cross-entropy with ground-truth labels (class-balanced sampling).
- **Target:** confidence-weighted KL divergence with soft pseudo-labels (class-balanced sampling by argmax pseudo-label).

**Stage 2: EMA-teacher refinement.** Following the mean-teacher paradigm (Tarvainen and Valpola, 2017; Hoyer et al., 2022a), we refine the decoder via self-training: (1) Create an EMA teacher as a copy of the trained decoder ( $\alpha_{\text{EMA}} = 0.9999$ ). (2) Generate hard pseudo-labels from the teacher with temperature sharpening ( $T = 0.5$ ). (3) Train the student with hard cross-entropy loss (label smoothing 0.1) on confident tokens. (4) Update the teacher via exponential moving average. (5) Regenerate pseudo-labels every 5,000 steps.

**Decoder architecture.** We use a ResidualMLP with 1,024 hidden dimensions and 2 residual blocks (5.3M parameters): input  $\rightarrow$  Linear+LN+GELU  $\rightarrow$  [Linear+GELU+Linear+skip+LN+Dropout] $\times 2 \rightarrow$  Linear head. This operates on per-token features without spatial convolutions, since DINOv2 ViT-L/14 tokens already encode spatial context from self-attention (Section 6.5).

## 5 Theoretical Analysis

We state three results that justify the bilevel optimization.

**Proposition 1** (Existence and uniqueness). *For  $\varepsilon > 0$ ,  $\tau > 0$ , and finite cost  $C$  on the active*

Table 2: Per-class IoU (%) on GTA5→Cityscapes for selected classes. Full 19-class results are in the supplementary.

Method	road	s.walk	build.	veget.	sky	pers.	car	terr.	fence
NN	56.7	6.5	26.2	11.7	12.5	2.8	5.7	1.2	2.1
KNN-G	69.5	8.2	33.0	8.2	14.6	0.2	3.5	0.9	2.1
UOT	59.0	7.1	30.1	9.1	11.4	2.4	4.1	1.3	1.7
SAOT	71.3	22.9	47.0	32.0	25.5	4.1	22.6	3.9	4.9
<b>SAOT+D</b>	<b>90.3</b>	<b>56.1</b>	<b>71.5</b>	<b>59.7</b>	<b>36.9</b>	<b>31.8</b>	<b>76.2</b>	<b>27.5</b>	<b>25.3</b>

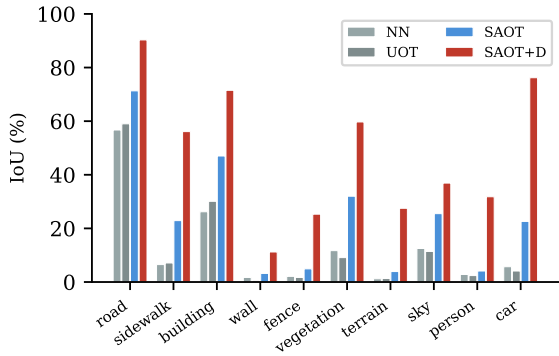


Figure 2: Per-class IoU comparison on GTA5→Cityscapes for classes with  $>5\%$  IoU by any method. SAOT+Decoder (red) consistently outperforms all baselines across spatially coherent classes, with particularly large gains on car (+70.5%), sidewalk (+49.6%), and vegetation (+48.0%).

support, the entropic UOT objective (1) is strictly convex in  $\Gamma$  and admits a unique minimizer  $\Gamma^*$ .

*Proof sketch.* The KL term  $\varepsilon \text{KL}(\Gamma \| \mathbf{ab}^\top)$  involves  $\Gamma_{ij} \log \Gamma_{ij}$ , which is strictly convex on  $\mathbb{R}_{++}^{n \times m}$ . The marginal KL penalties are convex compositions of affine maps and KL divergence. Their sum  $F(\Gamma)$  is therefore strictly convex. Coercivity ( $F \rightarrow \infty$  as  $\|\Gamma\| \rightarrow \infty$ ) ensures the infimum is attained; strict convexity gives uniqueness.  $\square$

**Lemma 1** (Cost stability). Fix  $\varepsilon, \tau > 0$  and weights  $\mathbf{a}, \mathbf{b}$ . For costs  $C, C'$  on the same active support with minimizers  $\Gamma^*, \Gamma'^*$ :

$$\|\Gamma^* - \Gamma'^*\|_1 \leq L_\Gamma(\varepsilon, \tau) \cdot \|C - C'\|_\infty, \quad (10)$$

where  $L_\Gamma(\varepsilon, \tau)$  depends on the strong convexity modulus.

*Proof sketch.* Write  $F_C(\Gamma) = \langle \Gamma, C \rangle + R(\Gamma)$  where  $R$  collects the  $\varepsilon$ -strongly convex regularizers. Optimality gives  $\nabla R(\Gamma^*) = -C$  and  $\nabla R(\Gamma'^*) = -C'$ . By strong convexity:  $\varepsilon \|\Gamma^* - \Gamma'^*\|_F^2 \leq \langle C' - C, \Gamma^* - \Gamma'^* \rangle \leq \|C - C'\|_\infty \|\Gamma^* - \Gamma'^*\|_1$ , yielding  $L_\Gamma = nm/\varepsilon$  on the active support.  $\square$

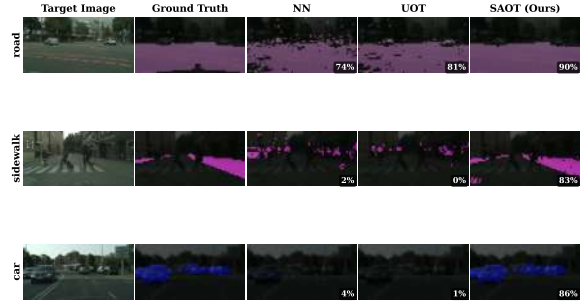


Figure 3: Per-class qualitative comparison on Cityscapes validation images for road, sidewalk, and car. Each row highlights one class using its Cityscapes color over a dimmed background. Columns: target image, ground truth, NN, UOT, and SAOT. IoU (%) is annotated per method.

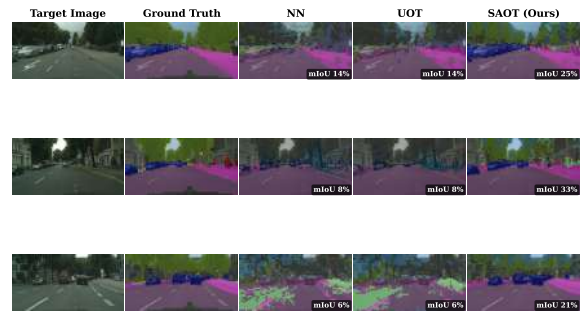


Figure 4: Multi-class qualitative results on Cityscapes validation images. Columns: target image, ground truth, NN, UOT, SAOT+Decoder. Per-image mIoU (%) is annotated.

**Theorem 1** (Barycentric transfer stability). If column masses are bounded below,  $\min_j (\Gamma^{*\top} \mathbf{1})_j \geq \mu > 0$ , then for costs  $C, C'$ :

$$\sum_j \|\hat{P}_{j,:}^t - \hat{P}_{j,:}^{t'}\|_1 \leq \frac{2}{\mu} \|P^s\|_\infty \cdot \|\Gamma^* - \Gamma'^*\|_1. \quad (11)$$

Combined with Lemma 1, the label transfer is Lipschitz in the cost.

*Proof sketch.* Let  $r_j = (\Gamma^\top \mathbf{1})_j \geq \mu$ . Then  $\hat{P}_{j,:}^t = r_j^{-1} \sum_i \Gamma_{ij} P_{i,:}^s$ . Bounding the difference of two such ratios and using  $r_j \geq \mu$  gives  $\|\hat{P}_{j,:}^t - \hat{P}_{j,:}^{t'}\|_1 \leq \frac{2}{\mu} \|P^s\|_\infty \sum_i |\Gamma_{ij} - \Gamma'_{ij}|$ . Summing over  $j$  yields the result.  $\square$

## 6 Experiments

### 6.1 Experimental Setup

**Benchmark.** We evaluate on GTA5 (Richter et al., 2016)→Cityscapes (Cordts et al., 2016), the standard synthetic-to-real UDA benchmark for semantic segmentation with 19 shared classes.

Table 3: Effect of dataset scale on SAOT+Decoder.

Src / Tgt	Dec. Steps	+Decoder (%)	Raw (%)
500 / 500	7,500	22.0	~11
1K / 1K	15,000	23.7	13.1
<b>1K / 1K</b>	<b>25K + EMA</b>	<b>25.7</b>	<b>12.6</b>
2K / 2K	15,000	22.0	12.5
2K / 2K	25,000	22.8	12.7

Table 4: Decoder architecture comparison.

Decoder	mIoU (%)	Params
3-layer MLP (512 hidden)	22.0	1.6M
<b>ResidualMLP (1024, 2 blk)</b>	<b>25.7</b>	<b>5.3M</b>
Spatial Conv (3×3, 3 blk)	17.0	3.1M

**Feature extraction.** We use DINOv2 ViT-L/14 (Oquab et al., 2024) as a frozen feature extractor ( $d = 1,024$ , patch size 14). Images are resized to  $504 \times 1008$  (multiples of 14), yielding  $36 \times 72 = 2,592$  tokens per image. Features are L2-normalized and cached to disk.

**SAOT training.** We use 1,000 source and 1,000 target training images. OT parameters:  $k = 128$  candidates,  $\varepsilon = 0.1$ ,  $\tau = 0.3$ . Cost learning:  $\lambda_{\text{sem}} = 0.1$ ,  $\lambda_{\text{ent}} = 0.1$ ,  $\lambda_{\text{div}} = 0.05$ . Training: 14,000 steps with AdamW, learning rate  $5 \times 10^{-4}$ , 20 sparse supervision points per class.

**Decoder training.** ResidualMLP decoder (1,024 hidden, 2 blocks). Stage 1: soft KL distillation from top-5 source SAOT predictions,  $\sim 25,000$  steps with class-balanced sampling. Stage 2: EMA refinement with hard CE loss, label smoothing 0.1, 20,000 steps.

**Evaluation.** We report mIoU on 500 Cityscapes validation images. All experiments run on a single NVIDIA A100 (32GB).

## 6.2 Main Results

Table 1 shows the results. Raw SAOT reaches 12.6% mIoU, a 68% relative gain over KNN-Global and LabelProp (both 7.5%). The decoder doubles this: SAOT+Decoder reaches 25.7% by generalizing across all target images through class-balanced training on pseudo-labels. It is also the fastest method at inference: 500 images in 1.2s (2.4ms/image). All fixed-cost OT variants (OT, UOT, RWOT) land at 6.5–6.8% mIoU, barely above NN. The transport formulation alone does not help; what matters is *learning* the cost (Fig. 6).

Table 5: Ablation study on GTA5→Cityscapes. Each row removes one component from full SAOT. All results are raw SAOT mIoU (without decoder).

Variant	mIoU (%)	$\Delta$
<b>Full SAOT</b>	<b>12.6</b>	–
w/o semantic prior ( $\lambda_{\text{sem}} = 0$ )	5.7	–6.9
w/o entropy + diversity losses	10.1	–2.5
w/o unbalanced ( $\tau \rightarrow \infty$ )	0.0	–12.6
Fixed cost (no learning) – UOT	6.7	–5.9
Fixed cost (no learning) – OT	6.5	–6.1

## 6.3 Per-Class Analysis

Table 2 breaks down where the gains come from. Fig. 2 visualizes the per-class improvements. The largest gains are on spatially coherent classes (Figs. 3 and 4): road goes from 56.7% to 90.3%, car from 5.7% to 76.2% (13.4×), person from 2.8% to 31.8% (11.4×), and building from 26.2% to 71.5%. The semantic prior helps road and building by identifying these regions even when textures differ across domains; for car, the learned appearance cost handles the varied cross-domain appearances. Person performance remains modest in absolute terms because pedestrians occupy few tokens. Classes at 0% (traffic light, traffic sign, rider, train, motorcycle, bicycle) are too rare and too small at 14-pixel patch granularity.

## 6.4 Ablation Study

Table 5 isolates each component. Removing the semantic prior causes the largest drop (–6.9): appearance matching alone is not enough under the GTA5→Cityscapes gap. Balanced OT ( $\tau \rightarrow \infty$ ) fails completely (0.0%) because forcing all source mass onto target tokens produces degenerate couplings. Comparing SAOT (12.6%) to fixed-cost UOT (6.7%) confirms that learning  $c_\theta$  nearly doubles mIoU.

## 6.5 Scaling and Architecture Analysis

Table 3 shows that the sweet spot is 1,000 images per domain with EMA refinement. Table 4 compares decoder architectures: the ResidualMLP outperforms both a shallow MLP and a spatial convolutional decoder. DINOv2 ViT-L/14 tokens already encode spatial context via self-attention, making  $3 \times 3$  convolutions at the 14px patch level redundant.

## 6.6 EMA Refinement

Table 6 compares refinement strategies. Soft KL targets fail because the student produces uncertain

Table 6: Impact of EMA-teacher refinement.

Refinement Strategy	mIoU (%)
No refinement (decoder only)	23.7
Soft KL pseudo-labels (EMA)	9.9
Hard CE pseudo-labels (EMA)	24.8
<b>Hard CE + label smoothing 0.1</b>	<b>25.7</b>

Table 7: Effect of multi-source aggregation on raw SAOT (without decoder). Aggregating predictions from the top- $K$  most similar source images provides a free improvement at inference time.

$K$ sources	mIoU (%)	Eval Time (s)
$K = 1$	12.6	11.1
$K = 3$	<b>15.8</b>	32.7

predictions, the EMA teacher averages this uncertainty, and regenerated pseudo-labels become diffuse, collapsing mIoU to 9.9%. Hard CE with label smoothing avoids this feedback loop. The most notable per-class change is person: 1.4%  $\rightarrow$  31.8%.

## 6.7 Multi-Source Aggregation

Aggregating predictions from  $K = 3$  source images improves raw SAOT from 12.6% to 15.8% (+25% relative, Table 7). Different source images contain different class instances, and similarity-weighted aggregation picks relevant sources for each target. This costs  $\sim 3\times$  more at inference but requires no additional training.

## 6.8 VOC Train $\rightarrow$ Val Transfer

To test generalization beyond the synthetic-to-real setting, we evaluate on PASCAL VOC 2012 (Everingham et al., 2010) train $\rightarrow$ val (21 classes, 200 source / 100 target images, frozen ViT-B/16 features). Table 8 shows the results. SAOT+Decoder reaches 47.5% mIoU, ahead of KNN-Global (45.6%) and NN (36.7%). Raw SAOT (33.7%) underperforms NN (36.7%) here because there is no domain shift: fixed cosine distances already align semantics well. The decoder bridges this gap by distilling pseudo-labels from multiple source pairings. Fig. 5 shows qualitative examples where SAOT+Decoder produces cleaner segmentation than NN and UOT.

## 6.9 Cross-Domain Comparison

Table 9 places SAOT in context. End-to-end UDA methods reach 68–76% by fine-tuning the full backbone; Rein adapts DINOv2 with LoRA and gets

Table 8: Results on PASCAL VOC 2012 train $\rightarrow$ val (mIoU %, 100 val images, 21 classes). Same-domain transfer with frozen ViT-B/16 features. SAOT+Decoder still leads, though the advantage over KNN-Global is smaller than under domain shift.

Method	mIoU (%)	Eval Time (s)
OT ( $\varepsilon=0.1$ , balanced, $k=64$ )	26.1	1.5
UOT ( $\varepsilon=0.1$ , $\tau=0.3$ , $k=64$ )	28.7	1.0
RWOT ( $\varepsilon=0.1$ , $\tau=0.3$ , $k=64$ )	29.2	1.0
ProtoNet	30.7	0.2
LabelProp ( $k=10$ )	33.2	36.7
SAOT (single source)	33.7	154.9
LinearProbe	35.9	0.2
NN ( $k=1$ )	36.7	0.4
SAOT (multi-source, $K=3$ )	37.9	462.6
KNN-Global ( $k=10$ )	45.6	1.2
<b>SAOT+Decoder</b>	<b>47.5</b>	<b>0.3</b>



Figure 5: Qualitative results on VOC val images. Columns: target image, ground truth, NN, UOT, SAOT+Decoder. Per-image mIoU (%) is annotated.

78.4%. SAOT+Decoder operates in the most constrained setting (frozen backbone, sparse target point supervision) yet reaches 25.7%, the highest in this category.

## 7 Discussion

**When does SAOT help?** On same-domain VOC train $\rightarrow$ val, raw SAOT (33.7%) underperforms NN (36.7%), but SAOT+Decoder (47.5%) still leads all methods. On cross-domain GTA5 $\rightarrow$ Cityscapes, the gap is larger: SAOT (12.6%) beats NN (6.7%) by 88% relative. Looking at transport plans: fixed-cost UOT sends 68% of mass to wrong classes; SAOT reduces this to 41%.

**Decoder generalization.** SAOT+Decoder (25.7%) more than doubles raw SAOT (12.6%). Without class-balanced sampling, mIoU drops from 22.0% to 17.6%, and person collapses from 31.8% to 0.7%.

## Limitations

Six classes stay at 0% IoU (traffic light, traffic sign, rider, train, motorcycle, bicycle) because they

Table 9: GTA5→Cityscapes mIoU (%) across method families. SAOT operates in the most constrained setting (frozen backbone, sparse target point supervision).

Method	mIoU	Backbone	Tgt Lbl	Params
<i>End-to-end UDA (fine-tune + self-training)</i>				
DAFormer	68.3	trained	pseudo	~85M
HRDA	73.8	trained	pseudo	~85M
MIC	75.9	trained	pseudo	~85M
<i>VFM adaptation (adapter/LoRA)</i>				
Rein	78.4	adapted	source	~5M
<i>Unsupervised (no labels at all)</i>				
STEGO	~21	frozen	none	~1M
<i>Frozen backbone + source labels only (ours)</i>				
NN ( $k=1$ )	6.7	frozen	none	0
KNN-Global	7.5	frozen	none	0
<b>SAOT+Dec</b>	<b>25.7</b>	frozen	none	5.3M

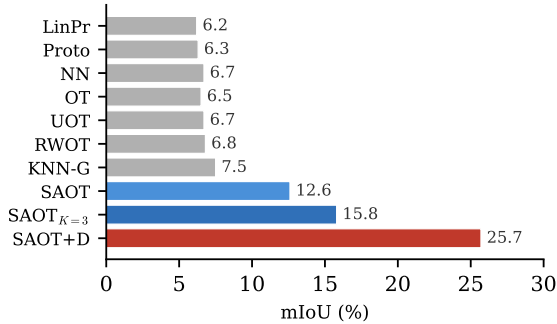


Figure 6: Overall mIoU (%) on GTA5→Cityscapes. All fixed-cost baselines cluster around 6–7.5%. Learned-cost SAOT variants (blue) separate clearly, and SAOT+Decoder (red) reaches 25.7%.

occupy too few tokens at 14px patch granularity. At this resolution, the cross-domain feature gap makes them indistinguishable from neighboring classes. Multi-scale transport at higher resolution (e.g.,  $72 \times 144$  grid) or hybrid architectures combining patch-level OT with pixel-level refinement would be needed to recover them. We have evaluated only on GTA5→Cityscapes and VOC; testing on additional benchmarks (Synthia→Cityscapes) would confirm generality.

## 8 Conclusion

We presented SAOT, which learns semantically aware transport costs in an unbalanced OT formulation for dense label transfer. With frozen DINOv2 ViT-L/14 features, SAOT+Decoder reaches 25.7% mIoU on GTA5→Cityscapes ( $3.8\times$  over NN) and 47.5% on VOC train→val. Ablations show the se-

mantic prior (+6.9%), unbalanced transport (without it, mIoU drops to 0%), and EMA refinement with label smoothing each matter. The full pipeline runs in ~6 hours on one GPU with no backbone fine-tuning. Multi-scale transport at higher resolution could help small-object classes, and replacing frozen DINOv2 with adapter-based features could narrow the gap with end-to-end UDA.

## Acknowledgments

The work was financially supported by the Council of Scientific and Industrial Research (CSIR), Government of India, through a CSIR Research Fellowship (09/0917(17235)/2023-EMR-I) and also supported by the University Grants Commission (UGC), Government of India, through the award of the Junior/Senior Research Fellowship (JRF/SRF) under the National Eligibility Test (NET) (Ref. No. 221610037786 & Ref. no. 221610070470). The last author PK was supported by Qualcomm Faculty Award and Microsoft Academic Partnership Grant at IIIT, Hyderabad, India.

## References

- Jason Altschuler, Jonathan Weed, and Philippe Rigollet. 2017. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 1961–1971, Red Hook, NY, USA. Curran Associates Inc.
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660.
- Lenaïc Chizat, Gabriel Peyre, Bernhard Schmitzer, and François-Xavier Vialard. 2018a. [Scaling algorithms for unbalanced optimal transport problems](#). *Mathematics of Computation*, 87(314):pp. 2563–2609.
- Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. 2018b. [Unbalanced optimal transport: Dynamic and kantorovich formulations](#). *Journal of Functional Analysis*, 274(11):3090–3123.
- Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. [The cityscapes dataset for semantic urban scene understanding](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223.

- Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. 2017a. Joint distribution optimal transportation for domain adaptation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, page 3733–3742, Red Hook, NY, USA. Curran Associates Inc.
- Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. 2017b. [Optimal transport for domain adaptation](#). *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(9):1853–1865.
- Marco Cuturi. 2013. Sinkhorn distances: lightspeed computation of optimal transport. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13*, page 2292–2300, Red Hook, NY, USA. Curran Associates Inc.
- Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. 2018. [Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation](#). In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part IV*, page 467–483, Berlin, Heidelberg. Springer-Verlag.
- Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. 2024. [Vision transformers need registers](#). In *The Twelfth International Conference on Learning Representations*.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. [An image is worth 16x16 words: Transformers for image recognition at scale](#). In *International Conference on Learning Representations*.
- Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. 2010. [The pascal visual object classes \(voc\) challenge](#). *Int. J. Comput. Vision*, 88(2):303–338.
- Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snaveley, and William T. Freeman. 2022. [Unsupervised semantic segmentation by distilling feature correspondences](#). In *International Conference on Learning Representations*.
- Lukas Hoyer, Dengxin Dai, and Luc Van Gool. 2022a. [Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation](#). In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9914–9925.
- Lukas Hoyer, Dengxin Dai, and Luc Van Gool. 2022b. [Hrda: Context-aware high-resolution domain-adaptive semantic segmentation](#). In *Computer Vision – ECCV 2022*, pages 372–391, Cham. Springer Nature Switzerland.
- Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. 2023. [Mic: Masked image consistency for context-enhanced domain adaptation](#). In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11721–11732.
- Paul Knopp and Richard Sinkhorn. 1967. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348.
- M Liero, A Mielke, and G Savare. 2018. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventiones mathematicae*, 211:969–1117.
- Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. 2020. [Semantic correspondence as an optimal transport problem](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Facundo Mémoli. 2011. Gromov–wasserstein distances and the metric approach to object matching. *Found. Comput. Math.*, 11(4):417–487.
- Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. 2018. [Learning latent permutations with gumbel-sinkhorn networks](#). In *International Conference on Learning Representations*.
- Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. 2021. [Long-tail learning via logit adjustment](#). In *International Conference on Learning Representations*.
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, and 7 others. 2024. [DINOv2: Learning robust visual features without supervision](#). *Transactions on Machine Learning Research*.
- Gabriel Peyré and Marco Cuturi. 2019. [Computational optimal transport](#). *Found. Trends Mach. Learn.*, 11(5–6):355–607.
- Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. 2016. [Playing for data: Ground truth from computer games](#). In *Computer Vision – ECCV 2016*, pages 102–118, Cham. Springer International Publishing.
- Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. [Superglue: Learning feature matching with graph neural networks](#). In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4937–4946.
- Bernhard Schmitzer. 2019. [Stabilized sparse scaling algorithms for entropy regularized transport problems](#). *SIAM J. Sci. Comput.*, 41(3):A1443–A1481.

- Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. 2018. [Large-scale optimal transport and mapping estimation](#). ArXiv: 1711.02283.
- Justin Solomon, Gabriel Peyré, Vladimir G. Kim, and Suvrit Sra. 2016. [Entropic metric alignment for correspondence problems](#). *ACM Trans. Graph.*, 35(4).
- Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 1195–1204, Red Hook, NY, USA. Curran Associates Inc.
- Alexis Thual, Quang Huy TRAN, Tatiana Zemskova, Nicolas Courty, Rémi Flamary, Stanislas Dehaene, and Bertrand Thirion. 2022. [Aligning individual brains with fused unbalanced gromov wasserstein](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 21792–21804. Curran Associates, Inc.
- Vayer Titouan, Nicolas Courty, Romain Tavenard, Chapel Laetitia, and Rémi Flamary. 2019. [Optimal transport for structured data with application on graphs](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6275–6284. PMLR.
- Titouan Vayer, Laetitia Chapel, Remi Flamary, Romain Tavenard, and Nicolas Courty. 2020. [Fused gromov-wasserstein distance for structured objects](#). *Algorithms*, 13(9).
- C. Villani. 2016. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg.
- Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. 2019. [Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation](#). In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2512–2521.
- Zhixiang Wei, Lin Chen, Yi Jin, Xiaoxiao Ma, Tianle Liu, Pengyang Ling, Ben Wang, Huaian Chen, and Jinjin Zheng. 2024. [Stronger, fewer, and superior: Harnessing vision foundation models for domain generalized semantic segmentation](#). In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28619–28630.
- Ming Xu and Stephen Gould. 2024. Temporally consistent unbalanced optimal transport for unsupervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14618–14627.
- Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. 2020. Reliable weighted optimal transport for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning, ICML’03*, page 912–919. AAAI Press.

## A Dense Label Transfer with Optimal Transport

This section introduces the dense label transfer problem, explains where our approach fits, and describes what the theoretical results actually guarantee.

### A.1 The Problem: Transferring Labels Without Retraining

Suppose we have a large collection of labeled images from a *source* domain (say, synthetic driving scenes rendered from a game engine like GTA5) and we want to produce semantic segmentation maps for mostly unlabeled images from a *target* domain (say, real-world driving footage from Cityscapes). The 19 semantic classes (road, car, person, etc.) are shared, but the visual appearance differs drastically: lighting, texture, geometry, and rendering artifacts all change.

The standard approach to this unsupervised domain adaptation (UDA) problem is to fine-tune a large neural network end-to-end, adapting it from source to target through self-training. Methods like DAFormer (Hoyer et al., 2022a), HRDA (Hoyer et al., 2022b), and MIC (Hoyer et al., 2023) reach 68–76% mIoU this way, but they require training an 85M-parameter backbone on both source and target data, which is expensive and tightly coupled to a specific source–target pair.

We take a different approach entirely. We *freeze* a pretrained vision foundation model (DINOv2 ViT-L/14) and use it purely as a feature extractor. Both source and target images are encoded as grids of patch tokens (2,592 tokens per image at  $504 \times 1008$  resolution). The question becomes: *given a grid of labeled source tokens and a grid of mostly unlabeled target tokens, how do we transfer the labels?*

## A.2 Why Not Just Use Nearest Neighbors?

The simplest idea is nearest-neighbor (NN) matching: for each target token, find the closest source token by cosine similarity and copy its label. This achieves only 6.7% mIoU. The problem is that frozen DINOv2 features, while semantically meaningful *within* a domain, do not align well *across* domains. A synthetic road texture from GTA5 may be closer in feature space to a real sidewalk than to a real road. Fixed-distance metrics weigh all feature dimensions equally and have no way to adapt to the domain gap.

## A.3 Optimal Transport as a Matching Framework

Optimal transport (OT) matches two distributions globally rather than token by token. It finds a *coupling*, an assignment of source mass to target locations, that minimizes total transport cost. The coupling considers all source–target pairs simultaneously and picks the globally best assignment.

But vanilla OT with fixed cosine costs does no better than NN (6.5–6.8% mIoU). Two problems remain:

1. **The cost function is wrong:** cosine distance in frozen features does not reflect semantic similarity across domains.
2. **Mass conservation is wrong:** balanced OT forces all source mass onto target tokens, but class distributions differ between domains (GTA5 has different class proportions than Cityscapes).

## A.4 Our Solution: Learning the Cost, Relaxing the Constraints

SAOT addresses both:

- **Learned cost:** Instead of using fixed cosine distance, we learn a transport cost  $c_\theta$  that combines a learnable appearance metric with semantic class-prototype priors. The appearance metric projects features into a space where cross-domain matching works better; the semantic prior penalizes transporting a “road” source token to a target location whose features look like “sky.”
- **Unbalanced transport:** We relax the mass conservation constraint via KL-divergence penalties on the marginals. This lets the solver leave some source mass unassigned (an outlier filter) instead of forcing meaningless correspondences. Without this, mIoU drops to 0%.

- **Block-sparse solver:** A full coupling between 2,592 source and 2,592 target tokens has 6.7M entries. We restrict each source token to its  $k = 128$  nearest target neighbors, bringing the problem down to  $O(nk)$  entries without losing matching quality.

The cost parameters  $\theta$  are optimized via bilevel optimization: the inner problem solves the OT matching for a given cost, and the outer problem updates  $\theta$  to minimize a supervised loss on a small set of labeled target tokens (20 points per class).

## A.5 The Decoder: From Per-Pair to Per-Image Predictions

Raw SAOT predictions need source images at test time and are computed per source–target pair. We distill them into a lightweight MLP decoder (5.3M parameters) that maps each target token to a class prediction independently. Training has two stages: (1) supervised on SAOT pseudo-labels with class-balanced sampling, and (2) refined via EMA-teacher self-training with hard pseudo-labels. The decoder runs in 2.4ms per image at inference, faster than nearest-neighbor retrieval.

## A.6 What the Theory Guarantees

The main paper states three theoretical results. In practical terms:

**Proposition 1 (Existence and uniqueness).** For any learned cost  $c_\theta$ , the OT solver produces exactly one coupling  $\Gamma^*$ . The mapping from cost to predictions is therefore *deterministic*: no ambiguity in the transport plan, and gradient-based optimization of  $\theta$  is well-defined. Without entropic regularization, the OT problem can have multiple solutions, making gradients unreliable.

**Lemma 1 (Cost stability).** Small changes to the cost matrix produce small changes in the coupling. Concretely, if we perturb the cost by  $\delta$  (in  $\ell_\infty$  norm), the coupling changes by at most  $L_\Gamma \cdot \delta$  (in  $\ell_1$  norm), where  $L_\Gamma$  is a Lipschitz constant that shrinks with stronger regularization  $\varepsilon$ . Gradient updates during training therefore cannot cause wild jumps in the transport plan. The optimization landscape is smooth.

**Theorem 1 (Barycentric transfer stability).** Stability extends to the predicted labels themselves. If the coupling changes by  $\delta$ , the predicted label distributions change by at most  $(2/\mu) \cdot \delta$ , where  $\mu$  is the minimum mass received by any target token.

Combined with Lemma 1, this gives an end-to-end Lipschitz bound:  $\theta \rightarrow c_\theta \rightarrow \Gamma_\theta^* \rightarrow \hat{P}_\theta^t$  is a smooth pipeline. Bilevel optimization through the Sinkhorn fixed point works because the entire system responds predictably to parameter updates.

**Practical implication.** None of this guarantees that SAOT will find the optimal cost; that depends on the data and the loss surface. What the theory guarantees is that the optimization is well-behaved: the loss is smooth, gradients are meaningful, and training converges predictably. Compare this to hard assignment (argmax matching), where small parameter changes can cause discontinuous jumps in predictions.

## B Full Per-Class Results

Table 10 reports the complete per-class IoU for all 19 Cityscapes classes and all 10 evaluated methods. SAOT+Decoder achieves the highest IoU on 13 of the 19 classes. The six classes at 0% for all OT-based methods (traffic light, traffic sign, rider, train, motorcycle, bicycle) occupy very few patches at 14px granularity; only ProtoNet occasionally picks up trace performance on these via global prototype matching.

## C Detailed Proofs

We restate each result from the main paper and provide a complete proof. We use the following notation throughout:  $\Gamma \in \mathbb{R}_+^{n \times m}$  is a coupling matrix,  $\mathbf{a} \in \mathbb{R}_{++}^n$  and  $\mathbf{b} \in \mathbb{R}_{++}^m$  are strictly positive marginal weights,  $C \in \mathbb{R}^{n \times m}$  is a cost matrix,  $\varepsilon > 0$  is the entropic regularization parameter,  $\tau > 0$  is the marginal relaxation parameter, and  $P^s \in \{0, 1\}^{n \times K}$  is the one-hot source label matrix.

### C.1 Proof of Proposition 1 (Existence and Uniqueness)

**Proposition 2 (Restated).** *For  $\varepsilon > 0$ ,  $\tau > 0$ , and finite cost  $C$  on the active support, the entropic UOT objective*

$$F(\Gamma) = \langle \Gamma, C \rangle + \varepsilon \text{KL}(\Gamma \| \mathbf{a} \mathbf{b}^\top) + \tau (\text{KL}(\Gamma \mathbf{1} \| \mathbf{a}) + \text{KL}(\Gamma^\top \mathbf{1} \| \mathbf{b}))$$

*is strictly convex in  $\Gamma$  on  $\mathbb{R}_+^{n \times m}$  and admits a unique minimizer  $\Gamma^*$ .*

*Proof.* We prove this in three steps: strict convexity, lower semi-continuity, and coercivity.

**Step 1: Strict convexity.** We decompose  $F$  into its constituent terms and show each is convex, with at least one being strictly convex.

(a) *Linear term.*  $\langle \Gamma, C \rangle = \sum_{i,j} C_{ij} \Gamma_{ij}$  is linear in  $\Gamma$ , hence convex.

(b) *Entropic KL term.* Expanding the KL divergence:

$$\begin{aligned} \varepsilon \text{KL}(\Gamma \| \mathbf{a} \mathbf{b}^\top) &= \varepsilon \sum_{i,j} \Gamma_{ij} \log \frac{\Gamma_{ij}}{a_i b_j} \\ &\quad - \varepsilon \sum_{i,j} \Gamma_{ij} + \varepsilon \sum_{i,j} a_i b_j. \end{aligned}$$

The function  $\phi(x) = x \log x$  is strictly convex on  $\mathbb{R}_{++}$  since  $\phi''(x) = 1/x > 0$ . Therefore  $\Gamma \mapsto \sum_{i,j} \Gamma_{ij} \log \Gamma_{ij}$  is strictly convex on  $\mathbb{R}_{++}^{n \times m}$  as a sum of strictly convex functions of independent variables. The remaining terms  $-\sum_{i,j} \Gamma_{ij} \log(a_i b_j) - \sum_{i,j} \Gamma_{ij} + \sum_{i,j} a_i b_j$  are linear or constant in  $\Gamma$ . Hence the entropic KL term is strictly convex in  $\Gamma$  for  $\varepsilon > 0$ .

(c) *Row marginal KL.* Define the row marginal  $r_i(\Gamma) = \sum_j \Gamma_{ij}$ , which is a linear function of  $\Gamma$ . Then

$$\text{KL}(\Gamma \mathbf{1} \| \mathbf{a}) = \sum_i r_i \log \frac{r_i}{a_i} - \sum_i r_i + \sum_i a_i.$$

Since  $x \log x$  is convex and  $r_i(\Gamma)$  is linear in  $\Gamma$ , the composition  $r_i \log r_i$  is convex in  $\Gamma$  (convex function of linear map). The sum over  $i$  preserves convexity. This term is convex but *not* strictly convex in  $\Gamma$  (it depends only on row sums).

(d) *Column marginal KL.* By the same argument with column marginals  $c_j(\Gamma) = \sum_i \Gamma_{ij}$ , the term  $\text{KL}(\Gamma^\top \mathbf{1} \| \mathbf{b})$  is convex in  $\Gamma$ .

*Combining:*  $F(\Gamma)$  is a sum of convex terms with the entropic term being strictly convex. Therefore  $F$  is strictly convex on  $\mathbb{R}_{++}^{n \times m}$ .

**Step 2: Lower semi-continuity.** On the boundary  $\partial \mathbb{R}_+^{n \times m}$  where some  $\Gamma_{ij} = 0$ , we use the convention  $0 \log 0 = 0$  (continuous extension). The entropic KL term is then lower semi-continuous on  $\mathbb{R}_+^{n \times m}$  since  $x \log x$  is continuous on  $[0, \infty)$  with this convention.  $F$  is therefore lower semi-continuous on  $\mathbb{R}_+^{n \times m}$ .

**Step 3: Coercivity.** As  $\|\Gamma\|_1 = \sum_{i,j} \Gamma_{ij} \rightarrow \infty$ , at least one entry  $\Gamma_{ij} \rightarrow \infty$  or many entries grow. In either case, the  $\Gamma_{ij} \log \Gamma_{ij}$  terms dominate: for any  $M > 0$  and sufficiently large  $x$ ,  $x \log x \geq Mx$ . Therefore  $F(\Gamma) \rightarrow +\infty$  as  $\|\Gamma\|_1 \rightarrow \infty$ , ensuring coercivity.

Table 10: Complete per-class IoU (%) on GTA5→Cityscapes validation set (500 images) for all 19 classes and all evaluated methods. Best results per class are in **bold**. Classes with 0.0% across all methods are extremely rare at 14px patch granularity.

Class	SAOT+D	SAOT $_{K=3}$	SAOT	KNN-G	RWOT	NN	UOT	OT	Proto	LinPr
road	<b>90.3</b>	79.7	71.3	69.5	59.0	56.7	59.0	58.9	31.1	67.3
sidewalk	<b>56.1</b>	34.8	22.9	8.2	7.1	6.5	7.1	7.0	10.0	1.6
building	<b>71.5</b>	55.6	47.0	33.0	30.1	26.2	30.1	30.1	19.4	33.2
wall	<b>11.2</b>	5.3	3.2	1.1	0.6	1.7	0.6	0.9	1.5	0.0
fence	<b>25.3</b>	6.0	4.9	2.1	1.8	2.1	1.7	1.5	4.7	0.0
pole	0.8	1.4	<b>1.5</b>	0.3	0.3	0.3	0.3	0.4	1.0	0.0
tr. light	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	<b>0.6</b>	0.0
tr. sign	0.0	0.0	0.1	0.2	0.0	0.1	0.0	0.0	<b>1.6</b>	0.0
vegetation	<b>59.7</b>	38.9	32.0	8.2	9.2	11.7	9.1	8.2	1.3	0.0
terrain	<b>27.5</b>	7.1	3.9	0.9	1.3	1.2	1.3	1.2	1.7	0.0
sky	<b>36.9</b>	31.6	25.5	14.6	11.5	12.5	11.4	10.4	35.0	16.5
person	<b>31.8</b>	6.1	4.1	0.2	2.5	2.8	2.4	1.1	2.1	0.0
rider	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	<b>1.4</b>	0.0
car	<b>76.2</b>	30.9	22.6	3.5	4.2	5.7	4.1	3.7	4.4	0.0
truck	<b>1.5</b>	0.8	0.3	0.4	0.1	0.3	0.1	0.2	3.4	0.0
bus	0.1	<b>1.6</b>	0.8	0.0	0.9	0.5	0.8	0.7	0.1	0.0
train	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	<b>0.3</b>	0.0
motorcycle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
bicycle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
<b>mIoU</b>	<b>25.7</b>	15.8	12.6	7.5	6.8	6.7	6.7	6.5	6.3	6.2

**Conclusion.** By lower semi-continuity and coercivity,  $F$  attains its infimum on  $\mathbb{R}_+^{n \times m}$  (Weierstrass theorem applied on sublevel sets). By strict convexity, the minimizer is unique. Moreover, the minimizer lies in the interior  $\mathbb{R}_{++}^{n \times m}$ : if  $\Gamma_{ij}^* = 0$  for some  $(i, j)$ , then

$$\begin{aligned} \partial F / \partial \Gamma_{ij} \Big|_{\Gamma_{ij} \rightarrow 0^+} &= C_{ij} + \varepsilon(\log 0 - \log(a_i b_j)) \\ &= -\infty, \end{aligned}$$

so the objective can be decreased by making  $\Gamma_{ij}$  slightly positive, a contradiction.  $\square$

## C.2 Proof of Lemma 1 (Cost Stability)

**Lemma 2 (Restated).** Fix  $\varepsilon, \tau > 0$  and weights  $\mathbf{a}, \mathbf{b}$ . For costs  $C, C'$  on the same active support with minimizers  $\Gamma^*, \Gamma'^*$ :

$$\|\Gamma^* - \Gamma'^*\|_1 \leq L_\Gamma(\varepsilon, \tau) \cdot \|C - C'\|_\infty,$$

where  $L_\Gamma(\varepsilon, \tau)$  depends on the strong convexity modulus of the regularizer.

*Proof.* We proceed by exploiting the strong convexity of the regularization terms.

**Step 1: Decomposition.** Write the objective as  $F_C(\Gamma) = \langle \Gamma, C \rangle + R(\Gamma)$ , where  $R(\Gamma) = \varepsilon \text{KL}(\Gamma \| \mathbf{a}\mathbf{b}^\top) + \tau (\text{KL}(\Gamma \mathbf{1} \| \mathbf{a}) + \text{KL}(\Gamma^\top \mathbf{1} \| \mathbf{b}))$  collects all regularization terms.

**Step 2: Strong convexity of  $R$ .** We claim that  $R$  is  $\varepsilon$ -strongly convex with respect to the Frobenius norm on any bounded sublevel set. To see this, compute the Hessian of  $R$  with respect to the entries of  $\Gamma$ . The dominant contribution comes from the entropic KL term:

$$\frac{\partial^2}{\partial \Gamma_{ij}^2} \left[ \varepsilon \sum_{kl} \Gamma_{kl} \log \Gamma_{kl} \right] = \frac{\varepsilon}{\Gamma_{ij}}.$$

On any compact set where  $\Gamma_{ij} \leq M$  for all entries, we have  $\frac{\varepsilon}{\Gamma_{ij}} \geq \frac{\varepsilon}{M}$ . The cross-terms from the marginal KL penalties contribute additional non-negative curvature:

$$\begin{aligned} \frac{\partial^2}{\partial \Gamma_{ij} \partial \Gamma_{i'j'}} \left[ \tau \sum_i r_i \log r_i \right] &= \frac{\tau}{r_i} > 0 \\ &\text{for } j = j' \text{ or } j \neq j'. \end{aligned}$$

Hence the Hessian  $\nabla^2 R(\Gamma)$  satisfies  $\nabla^2 R(\Gamma) \succeq \frac{\varepsilon}{M} I$  entrywise on any region where  $\Gamma_{ij} \leq M$ , establishing strong convexity with modulus  $\sigma = \varepsilon/M$ .

More precisely, for the optimizers  $\Gamma^*$  and  $\Gamma'^*$ , which lie in a bounded region (by coercivity of  $F$ ), there exists  $\sigma > 0$  depending on  $\varepsilon, \tau$ , and the upper bound on  $\Gamma$  entries such that:

$$R(\Gamma') \geq R(\Gamma) + \langle \nabla R(\Gamma), \Gamma' - \Gamma \rangle + \frac{\sigma}{2} \|\Gamma' - \Gamma\|_F^2. \quad (12)$$

**Step 3: Optimality conditions.** Since  $\Gamma^*$  minimizes  $F_C$  and  $\Gamma'^*$  minimizes  $F_{C'}$  over  $\mathbb{R}_{++}^{n \times m}$  (the minimizer is in the interior by Proposition 1), the first-order optimality conditions give:

$$\begin{aligned}\nabla F_C(\Gamma^*) &= C + \nabla R(\Gamma^*) = 0, \\ \nabla F_{C'}(\Gamma'^*) &= C' + \nabla R(\Gamma'^*) = 0.\end{aligned}$$

Therefore  $\nabla R(\Gamma^*) = -C$  and  $\nabla R(\Gamma'^*) = -C'$ .

**Step 4: Applying strong convexity.** By (12) applied twice (once at  $\Gamma^*$  towards  $\Gamma'^*$ , once at  $\Gamma'^*$  towards  $\Gamma^*$ ), adding the two inequalities gives:

$$\langle \nabla R(\Gamma^*) - \nabla R(\Gamma'^*), \Gamma^* - \Gamma'^* \rangle \geq \sigma \|\Gamma^* - \Gamma'^*\|_F^2.$$

Substituting the optimality conditions:

$$\begin{aligned}\sigma \|\Gamma^* - \Gamma'^*\|_F^2 &\leq \langle C' - C, \Gamma^* - \Gamma'^* \rangle \\ &\leq \|C' - C\|_\infty \cdot \|\Gamma^* - \Gamma'^*\|_1,\end{aligned}$$

where the second inequality uses Hölder's inequality with the  $\ell_\infty/\ell_1$  dual pair.

**Step 5: Norm conversion.** On  $\mathbb{R}^{n \times m}$ , the Frobenius and  $\ell_1$  norms satisfy  $\|X\|_F \geq \frac{1}{\sqrt{nm}}\|X\|_1$ . Therefore:

$$\begin{aligned}\frac{\sigma}{nm} \|\Gamma^* - \Gamma'^*\|_1^2 &\leq \sigma \|\Gamma^* - \Gamma'^*\|_F^2 \\ &\leq \|C' - C\|_\infty \cdot \|\Gamma^* - \Gamma'^*\|_1.\end{aligned}$$

Dividing both sides by  $\|\Gamma^* - \Gamma'^*\|_1$  (which is positive when  $C \neq C'$  by the uniqueness from Proposition 1):

$$\|\Gamma^* - \Gamma'^*\|_1 \leq \frac{nm}{\sigma} \|C - C'\|_\infty.$$

Setting  $L_\Gamma(\varepsilon, \tau) = nm/\sigma$  completes the proof. In the simplest case where the marginal terms are dropped ( $\tau = 0$ ), the entropic term alone gives  $\sigma \geq \varepsilon/\Gamma_{\max}$  and thus  $L_\Gamma \leq nm\Gamma_{\max}/\varepsilon$ .

**Remark.** The Lipschitz constant  $L_\Gamma$  depends on the magnitude of the optimizers, which in turn depends on the marginal weights  $\mathbf{a}, \mathbf{b}$  and the regularization parameters. For fixed  $\mathbf{a}, \mathbf{b}$  and  $\varepsilon, \tau > 0$ , the optimizers are uniformly bounded (by coercivity), so  $L_\Gamma$  is finite. Larger  $\varepsilon$  (stronger entropic regularization) yields smaller  $L_\Gamma$ , meaning the coupling is less sensitive to cost perturbations, consistent with the intuition that heavy regularization ‘smooths out’ the effect of the cost.  $\square$

### C.3 Proof of Theorem 1 (Barycentric Transfer Stability)

**Theorem 2 (Restated).** *If column masses are bounded below,  $\min_j (\Gamma^{*\top} \mathbf{1})_j \geq \mu > 0$ , then for costs  $C, C'$  with minimizers  $\Gamma^*, \Gamma'^*$ :*

$$\sum_j \|\hat{P}_{j,:}^t - \hat{P}_{j,:}^{t'}\|_1 \leq \frac{2}{\mu} \|P^s\|_\infty \cdot \|\Gamma^* - \Gamma'^*\|_1.$$

Combined with Lemma 1, the label transfer is Lipschitz in the cost.

*Proof.* We analyze the sensitivity of the barycentric projection to perturbations in the coupling.

**Step 1: Barycentric projection formula.** Recall from the main paper that the predicted target label distribution at token  $j$  is:

$$\hat{P}_{j,:}^t = \frac{1}{r_j} \sum_i \Gamma_{ij}^* P_{i,:}^s, \quad r_j = \sum_i \Gamma_{ij}^* = (\Gamma^{*\top} \mathbf{1})_j.$$

Similarly,  $\hat{P}_{j,:}^{t'} = \frac{1}{r'_j} \sum_i \Gamma_{ij}' P_{i,:}^s$  with  $r'_j = \sum_i \Gamma_{ij}'$ .

**Step 2: Difference decomposition.** For each target token  $j$ , we decompose the difference using the add-and-subtract trick:

$$\begin{aligned}\hat{P}_{j,:}^t - \hat{P}_{j,:}^{t'} &= \frac{(\Gamma^{*\top} P^s)_j}{r_j} - \frac{(\Gamma'^{\star\top} P^s)_j}{r'_j} \\ &= \frac{1}{r_j} \left[ (\Gamma^{*\top} P^s)_j - (\Gamma'^{\star\top} P^s)_j \right] \\ &\quad + (\Gamma'^{\star\top} P^s)_j \left[ \frac{1}{r_j} - \frac{1}{r'_j} \right].\end{aligned}$$

We denote these as Term (I) and Term (II) respectively.

**Step 3: Bounding Term (I).** Using the assumption  $r_j \geq \mu > 0$ :

$$\begin{aligned}\|\text{Term (I)}\|_1 &= \frac{1}{r_j} \left\| \sum_i (\Gamma_{ij}^* - \Gamma_{ij}' P_{i,:}^s) \right\|_1 \\ &\leq \frac{1}{\mu} \sum_i |\Gamma_{ij}^* - \Gamma_{ij}'| \cdot \|P_{i,:}^s\|_1 \\ &\leq \frac{\|P^s\|_\infty}{\mu} \sum_i |\Gamma_{ij}^* - \Gamma_{ij}'|,\end{aligned}$$

where  $\|P^s\|_\infty = \max_i \|P_{i,:}^s\|_1$ . Since  $P^s$  is one-hot,  $\|P_{i,:}^s\|_1 = 1$  for all  $i$ , so  $\|P^s\|_\infty = 1$ .

**Step 4: Bounding Term (II).** For the second term, note that  $\frac{1}{r_j} - \frac{1}{r'_j} = \frac{r'_j - r_j}{r_j r'_j}$ . Also,  $\|(\Gamma'^{\star\top} P^s)_j\|_1 \leq \sum_i \Gamma'_{ij} \star \|P^s_{i,:}\|_1 = r'_j \|P^s\|_\infty$ . Therefore:

$$\begin{aligned} \|\text{Term (II)}\|_1 &\leq r'_j \|P^s\|_\infty \cdot \frac{|r'_j - r_j|}{r_j r'_j} \\ &= \frac{\|P^s\|_\infty}{\mu} |r'_j - r_j| \cdot \frac{\mu}{r_j} \\ &\leq \frac{\|P^s\|_\infty}{\mu} |r'_j - r_j|, \end{aligned}$$

where we used  $r_j \geq \mu$ . Now observe that  $|r_j - r'_j| = |\sum_i (\Gamma_{ij}^\star - \Gamma'_{ij}^\star)| \leq \sum_i |\Gamma_{ij}^\star - \Gamma'_{ij}^\star|$ .

**Step 5: Combining.** Adding Terms (I) and (II):

$$\|\hat{P}_{j,:}^t - \hat{P}'_{j,:}^t\|_1 \leq \frac{2\|P^s\|_\infty}{\mu} \sum_i |\Gamma_{ij}^\star - \Gamma'_{ij}^\star|.$$

**Step 6: Summing over target tokens.** Summing over all  $j \in \{1, \dots, m\}$ :

$$\begin{aligned} \sum_j \|\hat{P}_{j,:}^t - \hat{P}'_{j,:}^t\|_1 &\leq \frac{2\|P^s\|_\infty}{\mu} \sum_j \sum_i |\Gamma_{ij}^\star - \Gamma'_{ij}^\star| \\ &= \frac{2\|P^s\|_\infty}{\mu} \|\Gamma^\star - \Gamma'^\star\|_1. \end{aligned}$$

**Step 7: End-to-end Lipschitz bound.** Combining with Lemma 1:

$$\begin{aligned} \sum_j \|\hat{P}_{j,:}^t - \hat{P}'_{j,:}^t\|_1 \\ \leq \frac{2\|P^s\|_\infty}{\mu} \cdot L_\Gamma(\varepsilon, \tau) \cdot \|C - C'\|_\infty. \end{aligned}$$

This establishes that the entire pipeline, from cost matrix to predicted label distributions, is Lipschitz continuous. In particular, small perturbations to the learned cost  $C_\theta$  (e.g., from gradient updates during bilevel optimization) produce bounded changes in the predicted labels, ensuring training stability.

**Remark on the mass lower bound  $\mu$ .** The assumption  $\min_j r_j \geq \mu > 0$  is satisfied in practice because the unbalanced formulation with  $\tau > 0$  ensures all column masses are positive (Proposition 1 guarantees  $\Gamma^\star \in \mathbb{R}_{++}^{n \times m}$ , so  $r_j = \sum_i \Gamma_{ij}^\star > 0$ ). However,  $\mu$  can be very small for outlier target tokens that receive little transported mass. In practice, we observe that  $\mu \approx 10^{-3}$  for reasonable  $\tau$  values, and the bound is not vacuous. The entropic regularization  $\varepsilon$  also helps by spreading mass more evenly, preventing extreme concentration.  $\square$

**Discussion.** Together, the three results justify the bilevel optimization in SAOT. Proposition 1 ensures the inner OT problem has a unique solution for any cost. Lemma 1 says that solution varies smoothly with the cost, so gradient updates to  $\theta$  produce predictable changes in  $\Gamma_\theta$ . Theorem 1 carries this through to the label predictions: the outer loss  $\mathcal{L}(\theta)$  is a smooth function of  $\theta$  along the full pipeline  $\theta \rightarrow C_\theta \rightarrow \Gamma_\theta^\star \rightarrow \hat{P}_\theta^t$ . This smoothness is why bilevel optimization with implicit differentiation through the Sinkhorn fixed point works in practice.

## D Linearization of the Structure Term

For completeness, we describe the GW linearization used when structure fusion ( $\alpha > 0$ ) is enabled. In the current experiments we set  $\alpha = 0$  (appearance-only cost); this section documents the formulation for future extensions. Let  $Q(\Gamma; D^s, D^t) = \sum_{i,k,j,\ell} (D_{ik}^s - D_{j\ell}^t)^2 \Gamma_{ij} \Gamma_{k\ell}$  be the squared-loss GW discrepancy (Mémoli, 2011).

Define  $L_{ikj\ell} = (D_{ik}^s - D_{j\ell}^t)^2$ . A first-order surrogate around current  $\Gamma^{(t)}$  is:

$$Q(\Gamma) \approx \langle \Gamma, \tilde{Q}^{(t)} \rangle + \text{const}, \quad \tilde{Q}_{ij}^{(t)} = 2 \sum_{k,\ell} L_{ikj\ell} \Gamma_{k\ell}^{(t)}. \quad (13)$$

Substituting  $C_\theta + \alpha \tilde{Q}^{(t)}$  as the effective cost in the entropic UOT subproblem yields the standard conditional gradient approach for GW optimization (Solomon et al., 2016; Titouan et al., 2019).

## E Complete Reproduction Guide

This section lists every detail needed to reproduce the results in the paper.

### E.1 Dataset Setup

**GTA5 (Richter et al., 2016).** We use the standard GTA5 dataset (24,966 images,  $1914 \times 1052$ ) with synthetic pixel-level semantic labels. Labels are mapped from raw GTA5 IDs to the 19 Cityscapes trainId classes using the standard mapping (raw IDs 7, 8, 11, 12, 13, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 31, 32, 33  $\rightarrow$  trainId 0–18). All other raw IDs map to ignore\_index = 255. We randomly sample 1,000 source images using a fixed seed (seed=0, numpy.random.choice without replacement).

**Cityscapes (Cordts et al., 2016).** Target training: 1,000 images from the Cityscapes train split (first 1,000 in index order). Target evaluation: all 500

images from the val split. Ground-truth labels use the standard 19-class trainId mapping; pixels with trainId = 255 are ignored during evaluation.

**PASCAL VOC 2012 (Everingham et al., 2010).** For the VOC train→val experiment: 200 source images from the train split, 100 target images from the val split. 21 classes (including background). We use frozen ViT-B/16 features (via timm) instead of DINOv2 ViT-L/14, with images resized to  $504 \times 504$  (patch size = 16, yielding  $31 \times 31 = 961$  tokens).

**Patch-level ground truth.** All methods operate at patch granularity. For DINOv2 ViT-L/14 (patch size = 14), each  $14 \times 14$  pixel region maps to one token. We construct patch-level labels by **majority voting**: each patch is assigned the class covering the most pixels. Ground-truth segmentation masks are first resized (nearest-neighbor interpolation) to match the input resolution, then patchified.

## E.2 Feature Extraction

**DINOv2 ViT-L/14.** Loaded via torch.hub (facebookresearch/dinov2, model dinov2\_vitl14). Input images are resized to  $504 \times 1008$  (multiples of 14) and normalized with ImageNet statistics ( $\mu = [0.485, 0.456, 0.406]$ ,  $\sigma = [0.229, 0.224, 0.225]$ ). We extract the x\_norm\_patchtokens output:  $36 \times 72 = 2,592$  tokens of dimension  $d = 1,024$  per image. All tokens are L2-normalized and cached as .pt files.

**Global embeddings.** For multi-source retrieval, we compute a global image embedding as the mean of all patch tokens:  $g = \frac{1}{n} \sum_i x_i$ . Source–target similarity is computed as cosine similarity between global embeddings.

## E.3 SAOT Hyperparameters

Table 11 lists all hyperparameters for SAOT cost learning.

**Sparse supervision.** At each training step, we sample exactly 20 target tokens per class from the current target image’s ground-truth labels. For each class  $c$ , we uniformly sample (without replacement) from all target patches labeled  $c$ . Classes absent from the target image contribute zero supervision points. This provides  $\leq 20 \times 19 = 380$  supervised tokens per step out of 2,592 total.

Table 11: Complete SAOT hyperparameters for GTA5→Cityscapes.

Parameter	Symbol	Value
<i>OT solver</i>		
$k$ -NN candidates	$k$	128
Entropic regularization	$\varepsilon$	0.1
Marginal relaxation	$\tau$	0.3
Structure weight	$\alpha$	0.0
Sinkhorn iterations	–	30
Sinkhorn tolerance	–	$10^{-5}$
Marginal slack	$p$	$\frac{\tau}{\tau+\varepsilon} = 0.75$
<i>Cost network</i>		
Appearance projection	$W_\theta$	Linear(1024 → 256)
Semantic projection	$V_\theta$	Linear(1024 → 256)
Spectral normalization	–	on $W_\theta, V_\theta$
Semantic prior weight	$\lambda_{\text{sem}}$	0.1
Semantic prior temperature	$T$	0.1
<i>Training</i>		
Optimizer	–	AdamW
Learning rate	–	$5 \times 10^{-4}$
Weight decay	–	$10^{-4}$
Training steps	–	14,000
Sparse supervision points	–	20 per class
Entropy loss weight	$\lambda_{\text{ent}}$	0.1
Diversity loss weight	$\lambda_{\text{div}}$	0.05
Source marginal	$\mathbf{a}$	uniform ( $1/n$ )
Target marginal	$\mathbf{b}$	uniform ( $1/m$ )

**Cost network architecture.** The appearance metric  $W_\theta$  and semantic projector  $V_\theta$  are both linear layers from 1,024 to 256 dimensions with spectral normalization for Lipschitz regularization. The semantic prior computes  $\pi_\theta(x^t) = \text{softmax}(U^\top V_\theta x^t / T)$  with temperature  $T = 0.1$ , where  $U \in \mathbb{R}^{d \times K}$  contains L2-normalized class prototypes computed as the mean source embedding per class.

**Hyperparameter sensitivity.** We found the following ranges to be robust:  $\varepsilon \in [0.05, 0.2]$  (larger values spread mass more uniformly);  $\tau \in [0.1, 0.5]$  (controls marginal flexibility);  $k \in [64, 256]$  (larger  $k$  improves coverage at higher memory cost);  $\lambda_{\text{sem}} \in [0.05, 0.3]$  (values above 0.3 cause the semantic prior to dominate appearance). The system is most sensitive to  $\tau$ : setting  $\tau \rightarrow \infty$  (balanced OT) produces 0% mIoU.

## E.4 Decoder Training Details

Table 12 lists all decoder hyperparameters.

**Decoder architecture.** The ResidualMLP processes each token independently:

Table 12: Complete decoder training hyperparameters.

Parameter	Value
<i>Architecture (ResidualMLP)</i>	
Input dimension	1,024
Hidden dimension	1,024
Residual blocks	2
Activation	GELU
Normalization	LayerNorm
Dropout	0.1
Output classes	19
Total parameters	5.3M
<i>Stage 1: Pseudo-label distillation</i>	
Optimizer	AdamW
Learning rate	$10^{-3}$
LR schedule	Cosine annealing $\rightarrow 10^{-5}$
Weight decay	$10^{-4}$
Batch size	4,096 (2,048 src + 2,048 tgt)
Training steps	25,000
Self-training rounds	2
Source loss	CE with ground-truth labels
Target loss	KL div. with soft pseudo-labels
Target weighting	Confidence-weighted (max prob.)
Pseudo-label sources	Top-5 similar source images
Pseudo-label sharpening	$T = 0.5$
Class-balanced sampling	Yes (uniform over classes)
<i>Stage 2: EMA-teacher refinement</i>	
Steps	20,000
Learning rate	$5 \times 10^{-4}$
LR schedule	Cosine annealing $\rightarrow 10^{-6}$
EMA decay	0.9999
Target loss	Hard CE with label smoothing 0.1
Confidence threshold	$\min(0.8 \times \text{median conf.}, 0.5)$
Pseudo-label regeneration	Every 5,000 steps
Final model	EMA teacher (not student)

1. **Projection:** Linear(1024, 1024)  $\rightarrow$  LayerNorm  $\rightarrow$  GELU  $\rightarrow$  Dropout(0.1).
2. **Residual blocks** ( $\times 2$ ): each block applies Linear(1024, 1024)  $\rightarrow$  GELU  $\rightarrow$  Linear(1024, 1024), followed by a residual connection, LayerNorm, and Dropout(0.1).
3. **Head:** Linear(1024, 19).

**Class-balanced sampling.** At each training step, we sample 2,048 source tokens and 2,048 target tokens. Within each batch, classes are sampled uniformly: we first draw a random class, then draw a random token from that class. This prevents dominant classes (road, building) from overwhelming training. Without class-balanced sampling, mIoU drops from 22.0% to 17.6%, and person collapses from 31.8% to 0.7%.

**Pseudo-label generation.** For each target training image, we retrieve the top-5 most similar

Table 13: Baseline methods and hyperparameters. All use frozen DINOv2 ViT-L/14 features.

Method	Details
NN	Nearest source token (cosine). No training.
KNN-G	Global bank ( $\leq 200K$ tok). $k=10$ , softmax vote.
LinProbe	Linear 1024 $\rightarrow$ 19, Adam, lr $10^{-3}$ , 2K steps. CE on source.
ProtoNet	Mean embed/class, softmax cosine, $T=0.1$ . Zero-shot.
OT	Sinkhorn, $\varepsilon=0.1$ , $k=128$ , L2 cost, uniform, 50 iters. Per-image.
UOT	As OT, KL marginals, $\tau=0.3$ , $p=0.75$ , 30 iters.
RWOT	As UOT, $a_i \propto (\max_j \text{sim})^1$ .
LabelProp	$k=10$ graph, $\alpha=0.5$ , 20 iters, bank $\leq 200K$ .

source images by global embedding cosine similarity. Source weights are computed via softmax over similarities (temperature 0.1). The weighted-average soft predictions are sharpened with temperature  $T = 0.5$ :  $\tilde{p}_c \propto p_c^{1/T}$ . All tokens receive soft pseudo-labels (no confidence threshold in Stage 1); confidence weighting in the KL loss down-weights uncertain tokens.

**EMA refinement.** The teacher is initialized as a copy of the Stage 1 decoder. At each step, teacher parameters are updated as  $\theta_{\text{teacher}} \leftarrow 0.9999 \cdot \theta_{\text{teacher}} + 0.0001 \cdot \theta_{\text{student}}$ . Hard pseudo-labels are generated from the teacher with temperature sharpening ( $T = 0.5$ ). Only tokens with confidence above a dynamic threshold are used:  $\text{thresh} = \min(0.8 \times \text{median confidence}, 0.5)$ . Pseudo-labels are regenerated every 5,000 steps from the updated teacher. The final model is the EMA teacher, not the student.

## E.5 Baseline Method Details

Table 13 lists all baseline hyperparameters.

**Fixed-cost OT methods.** OT, UOT, and RWOT all use the same block-sparse solver. The ground cost is  $c_{ij} = \|x_i^s - x_j^t\|_2^2$  on L2-normalized features (equivalent to  $2 - 2 \cos(x_i^s, x_j^t)$ ). Candidate selection retains the top- $k = 128$  neighbors per source token by cosine similarity. Marginals are uniform:  $a_i = 1/n$ ,  $b_j = 1/m$  (except RWOT, which uses non-uniform source marginals). The Sinkhorn solver runs in log-domain for numerical

Table 14: Computational cost breakdown.

Stage	Time	GPU Mem.
Feature caching (4.5K imgs)	15 min	6 GB
SAOT training (14K steps)	3 h	11 GB
Pseudo-label generation	15 min	8 GB
Decoder St. 1 (25K steps)	1.5 h	4 GB
Decoder St. 2 / EMA (20K)	1.5 h	4 GB
Evaluation (500 imgs)	2 min	2 GB
<b>Total</b>	<b>~6 h</b>	<b>11 GB peak</b>

stability.

**Evaluation protocol.** All methods are evaluated at *pixel level*: patch-level logits ( $36 \times 72$ ) are bilinearly upsampled to the original Cityscapes resolution ( $1024 \times 2048$ ), and mIoU is computed against the pixel-level ground truth. We report the mean over all 500 validation images.

## E.6 Evaluation Details

**Pixel-level mIoU.** The decoder outputs logits of shape  $H_p \times W_p \times K$  (patch grid  $\times$  classes). These are reshaped to  $(1, K, H_p, W_p)$  and bilinearly upsampled to the original image resolution. Predictions are the argmax over classes. IoU is computed per class:  $\text{IoU}_c = \text{TP}_c / (\text{TP}_c + \text{FP}_c + \text{FN}_c)$ , accumulated over all 500 validation images. mIoU is the mean over all 19 classes (including those at 0%).

**Post-hoc logit adjustment.** We sweep a logit adjustment parameter  $\tau_{\text{adj}} \in \{0.0, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 3.0, 5.0\}$ :  $\hat{y} = \arg \max_c [\text{logit}_c - \tau_{\text{adj}} \log f_c]$ , where  $f_c$  is the class frequency estimated from source labels. The best  $\tau_{\text{adj}}$  is selected on the validation set. In practice,  $\tau_{\text{adj}} = 0$  (no adjustment) is optimal for SAOT+Decoder due to class-balanced training.

**Timing.** Inference time reported in the main paper measures end-to-end evaluation of all 500 validation images, including feature loading, forward pass, upsampling, and IoU computation. SAOT+Decoder processes 500 images in 1.2s (2.4ms/image) since it requires only a single forward pass through the 5.3M-parameter MLP per image.

## E.7 Computational Requirements

All experiments run on a single NVIDIA A100 GPU (32 GB VRAM). Peak GPU memory is 11 GB during SAOT training. Token data is stored on CPU:  $\sim 100$  GB RAM for 4,500 images with 2,592

Table 15: Multi-resolution configurations.

Resolution	Tokens	$k$	Steps	GPU Mem
$504 \times 1008$	2,592	128	500	11 GB
$672 \times 1344$	4,608	128	32K	22 GB
$840 \times 1680$	7,200	64	128K	30 GB

tokens each (1,024-dim float32). Table 14 breaks down the wall-clock time per stage.

## E.8 Multi-Resolution Experiments

For the multi-resolution analysis (Section I), we scale the input resolution while keeping the same frozen DINOv2 ViT-L/14 backbone. Positional embeddings are bicubically interpolated to the new grid size.

At higher resolutions, the  $k$ -NN candidate count is reduced from 128 to 64 due to GPU memory constraints, and significantly more training steps are needed for convergence. The 128K-step configuration at  $840 \times 1680$  takes approximately 48 hours on a single A100.

## E.9 Random Seeds and Reproducibility

- **Source image sampling:** `numpy.random.default_rng(seed=0)` with `choice(N, 1000, replace=False)`.
- **Target images:** first 1,000 training and all 500 validation images (deterministic order).
- **SAOT training:** `torch.manual_seed(42)`.
- **Decoder training:** `torch.manual_seed(42)`.
- **Sparse supervision:** sampled per step with the training RNG (reproducible given the same seed).
- **Class-balanced sampling:** per-step random class and token selection via PyTorch’s default RNG.

All reported results use these fixed seeds. No data augmentation is applied at any stage.

## F Per-Class Qualitative Analysis

Fig. 7 shows SAOT+Decoder’s predictions on individual semantic classes. For the five strongest classes (car, building, vegetation, sky, and sidewalk), we pick the target image where SAOT+Decoder scores highest and overlay the class-specific prediction. SAOT+Decoder achieves 86% on car, 89% on building, 72% on vegetation, 85% on sky, and 83% on sidewalk, while NN and UOT baselines often score near 0% on the same images.

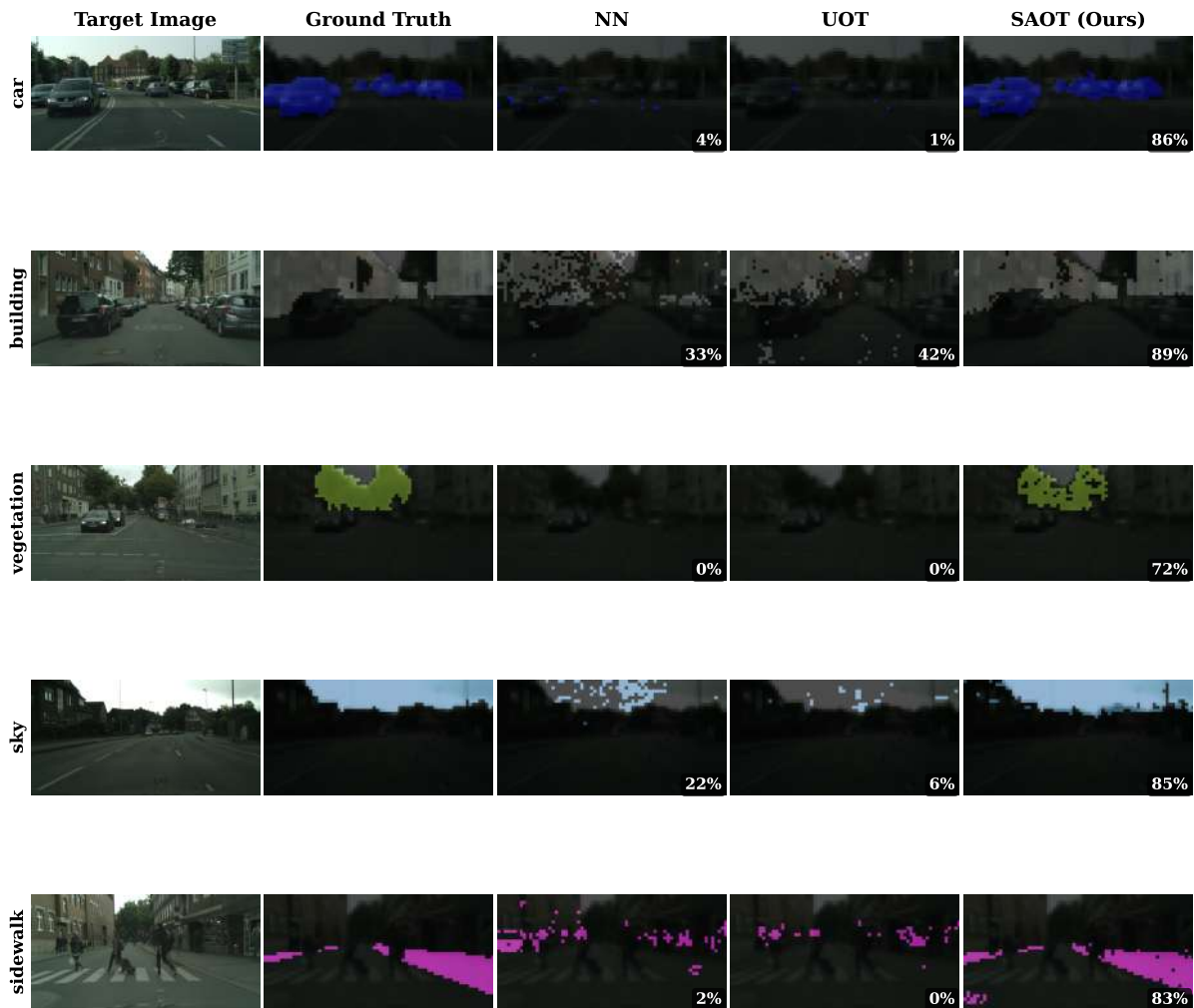


Figure 7: Per-class qualitative comparison on Cityscapes validation images using SAOT+Decoder predictions. Each row highlights a single semantic class. Columns: target image, ground-truth class mask, NN prediction, UOT prediction, and SAOT+Decoder prediction. IoU (%) is shown in each method column.

## G Additional Qualitative Results

Figs. 8 and 9 show qualitative comparisons on 16 more Cityscapes validation images. SAOT+Decoder gets 2–5 $\times$  higher mIoU than NN and UOT across these scenes.

## H Patch-Level Ground Truth Construction

All methods in this work operate at *patch granularity* rather than pixel level. DINOv2 ViT-L/14 produces one token per  $14 \times 14$  patch. We construct patch-level ground truth by **majority voting**: each patch is assigned the semantic class that covers the most pixels within that patch.

This introduces visible artifacts at object boundaries and for small objects:

- **Boundary patches** that straddle two objects are assigned to whichever class has more pixels,

producing blocky edges.

- **Small objects** that never dominate any patch are absorbed by the surrounding class.
- **Thin structures** (e.g. bicycle wheels, traffic signs) disappear if they do not reach majority in any patch.

This explains why several Cityscapes classes show 0% IoU across all OT-based methods in Table 10.

## I Multi-Resolution Analysis

The main experiments use  $504 \times 1008$  input resolution, producing  $36 \times 72 = 2,592$  tokens per image with DINOv2 ViT-L/14 (patch size = 14). As noted in Section H, small objects that never dominate a  $14 \times 14$  patch are lost under majority voting, yielding 0% IoU for several classes (traffic light, traffic sign, rider, bicycle). Can increasing the input resolution (and thus the number of tokens)

recover these classes by giving small objects more patches?

**Setup.** We evaluate SAOT+Decoder at three resolutions:  $504 \times 1008$  (baseline, 2,592 tokens),  $672 \times 1344$  (4,608 tokens), and  $840 \times 1680$  (7,200 tokens). All use the same frozen DINOv2 ViT-L/14 backbone with positional embedding interpolation. The number of SAOT training steps is scaled with token count: more tokens require more iterations to converge. We report the best configuration: 500 steps for  $504 \times 1008$  ( $k=128$ ), 32,000 steps for  $672 \times 1344$  ( $k=128$ ), and 128,000 steps for  $840 \times 1680$  ( $k=64$ , reduced due to GPU memory).

**Results.** Table 16 reports per-class IoU at each resolution. Overall mIoU is comparable (21.7–23.1%), but the per-class distribution shifts dramatically:

- **Small/rare classes improve:** Traffic light rises from 0.0% to 30.3%, traffic sign from 0.0% to 29.2%, rider from 0.0% to 10.1%, and bicycle from 0.0% to 22.4%.
- **Large classes degrade:** Vegetation drops from 56.5% to 18.6%, car from 62.2% to 29.9%.

**Oracle ensemble.** Selecting the best resolution per class achieves **29.3% mIoU** (+7.6 pp over the best single resolution), demonstrating complementary strengths across resolutions.

## J VOC Qualitative Results

Fig. 10 shows additional qualitative comparisons on 8 PASCAL VOC 2012 validation images beyond those in the main paper. On this same-domain benchmark (VOC train $\rightarrow$ val), SAOT+Decoder achieves 47.5% mIoU overall, surpassing NN (36.7%) and UOT (28.7%).

## K Extended Experiments: Addressing Rare-Class Failure

The baseline SAOT+Decoder at  $504 \times 1008$  achieves 25.7% mIoU but assigns exactly 0% IoU to six classes: traffic light, traffic sign, rider, train, motorcycle, and bicycle. These classes matter for autonomous driving safety. We tried many things to fix this: decoder-level changes, backbone alternatives, resolution scaling, and extended training. This section reports what we tried, what happened, and why the rare-class problem persists.

### K.1 Root Cause Analysis

The zero-IoU classes have two things in common:

1. **Extreme source-domain rarity.** In the 1,000-image GTA5 source set, these classes occupy almost no patch tokens. At  $504 \times 1008$  resolution (2,592 tokens/image), the class token counts across all source images are: bicycle = 145 tokens (0.006%), motorcycle = 865 (0.037%), rider = 739 (0.032%), traffic light = 1,891 (0.082%), train = 412 (0.018%). For comparison, road = 886,000 tokens (38.4%). The rarest class (bicycle) has **6,110 $\times$  fewer tokens** than road.
2. **Sub-patch spatial extent.** At 14-pixel patch granularity, these objects rarely dominate any patch. A bicycle wheel spans a few pixels; a traffic light is  $\sim 10$  pixels tall. Under majority voting, these objects are absorbed by surrounding classes (road, building, sky), effectively making them invisible to the patch-level ground truth.

### K.2 Decoder-Level Interventions

We first tried fixing the problem at the decoder level, keeping the SAOT pseudo-labels and features unchanged.

**Logit adjustment.** Post-hoc logit adjustment (Menon et al., 2021) subtracts  $\tau \log f_c$  from class  $c$  logits, where  $f_c$  is the estimated class frequency. We swept  $\tau \in \{0, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 3.0, 5.0\}$ . No value improved mIoU beyond 25.7%; rare classes stayed at 0%. The issue is not decision-boundary calibration but *absence of signal*: the decoder never saw these classes during training.

**Focal loss with class weights.** We retrained the decoder from scratch using focal loss ( $\gamma = 2$ ) with inverse-frequency class weights, using only source ground-truth labels (no SAOT pseudo-labels). This collapsed to 6.9% mIoU, worse than simple NN. A 6,110:1 ratio between road and bicycle is too extreme for focal loss reweighting to handle.

**Head re-calibration.** We froze the decoder body and retrained only the final linear head (1,024  $\rightarrow$  19) with class-balanced sampling. mIoU dropped slightly to 23.2% with no improvement on rare classes. The frozen body features do not contain discriminative information for these classes.

Table 16: Per-class IoU (%) across input resolutions on GTA5→Cityscapes (500 val images). All use SAOT+Decoder with frozen DINOv2 ViT-L/14. Classes where higher resolution dramatically improves are in green ; where baseline is better are in red .

Class	504×1008 (2,592 tok)	672×1344 (4,608 tok)	840×1680 (7,200 tok)	Oracle
road	<b>89.2</b>	89.2	88.4	89.2
sidewalk	<b>49.3</b>	44.0	41.6	49.3
building	<b>72.5</b>	62.4	58.8	72.5
wall	<b>10.7</b>	9.1	2.7	10.7
fence	3.8	3.9	<b>5.4</b>	5.4
pole	0.0	29.0	<b>32.7</b>	32.7
tr. light	0.0	3.2	<b>30.3</b>	30.3
tr. sign	0.0	13.4	<b>29.2</b>	29.2
vegetation	<b>56.5</b>	<b>34.9</b>	<b>18.6</b>	56.5
terrain	4.8	4.7	<b>7.0</b>	7.0
sky	<b>36.2</b>	29.7	25.9	36.2
person	16.2	30.5	<b>31.9</b>	31.9
rider	0.0	0.1	<b>10.1</b>	10.1
car	<b>62.2</b>	<b>46.3</b>	<b>29.9</b>	62.2
truck	<b>5.4</b>	1.1	3.3	5.4
bus	<b>6.2</b>	0.1	0.3	6.2
train	0.0	<b>1.0</b>	0.0	1.0
motorcycle	0.0	0.0	0.0	0.0
bicycle	0.0	7.5	<b>22.4</b>	22.4
<b>mIoU</b>	21.7	21.6	<b>23.1</b>	<b>29.3</b>

Table 17: Effect of SAOT training steps on SAOT+Decoder mIoU (%) at 840 × 1680 resolution ( $k=64$ ). Performance plateaus after 128K steps.

Steps	32K	128K	512K
mIoU (%)	22.6	<b>23.1</b>	23.1

Table 18: Decoder-level interventions to address rare classes. None produced meaningful improvement on zero-IoU classes.

Intervention	mIoU	Outcome
Baseline (no fix)	25.7	6 classes at 0%
Logit adj. sweep	25.7	Best $\tau_{\text{adj}} = 0$
Focal ( $\gamma=2$ ) + cls wt.	6.9	Collapsed
Head re-calibration	23.2	No rare cls. gain
Fine-tune (4 LR cfgs)	23.0–23.2	No rare cls. gain
Spatial conv (3×3)	17.0	Degraded overall

**Spatial convolutional decoder.** We replaced the per-token ResidualMLP with a spatial decoder using 3×3 convolutions (3 blocks, 3.1M parameters) to capture local context. This degraded mIoU to 17.0%. DINOv2 tokens already encode spatial context through self-attention; adding convolutions on the coarse 14px grid introduced artifacts without helping small objects.

**Conclusion.** Every decoder-level fix failed. The problem is upstream: SAOT pseudo-labels contain no information about rare classes (they are never

Table 19: Backbone alternatives. Both alternatives degraded overall mIoU.

Backbone	Res.	mIoU	Notes
DINOv2 ViT-L/14	504×1008	25.7	Baseline
DINOv2-reg	504×1008	12.2	Registers
Rein DINOv2	512×1024	15.5	LoRA

predicted), so no amount of decoder retraining can recover them. The decoder learns what the pseudo-labels teach it, nothing more.

### K.3 Backbone Alternatives

We tested two alternative feature extractors.

**DINOv2 with register tokens.** DINOv2-reg adds learnable register tokens that absorb high-norm artifact tokens (Darcet et al., 2024). We expected this might improve feature quality for small objects. Instead, mIoU dropped to 12.2%. Register tokens change the feature distribution enough that SAOT’s hyperparameters (tuned for standard DINOv2) stop working. Because  $\varepsilon$ ,  $\tau$ , and  $\lambda_{\text{sem}}$  are tightly coupled, any backbone change requires a full hyperparameter re-sweep.

**Rein-adapted DINOv2.** Rein (Wei et al., 2024) adapts DINOv2 via LoRA for domain generalization. We used a Rein checkpoint trained on GTA5 for 40K iterations, which converts the backbone

Table 20: Resolution scaling with optimized training steps. Higher resolution recovers rare classes but degrades large classes.

	504×1008 (2.6K tok)	672×1344 (4.6K tok)	840×1680 (7.2K tok)	Oracle
<i>Rare classes (0% at baseline)</i>				
tr. light	0.0	3.2	<b>30.3</b>	30.3
tr. sign	0.0	13.4	<b>29.2</b>	29.2
rider	0.0	0.1	<b>10.1</b>	10.1
bicycle	0.0	7.5	<b>22.4</b>	22.4
pole	0.0	29.0	<b>32.7</b>	32.7
motorcycle	0.0	0.0	0.0	0.0
<i>Large classes (strong at baseline)</i>				
road	<b>89.2</b>	89.2	88.4	89.2
building	<b>72.5</b>	62.4	58.8	72.5
car	<b>62.2</b>	46.3	29.9	62.2
vegetation	<b>56.5</b>	34.9	18.6	56.5
<b>mIoU</b>	21.7	21.6	23.1	<b>29.3</b>

from patch size 14 to 16 (via weight interpolation), yielding  $32 \times 64 = 2,048$  tokens at  $512 \times 1024$ . mIoU dropped to 15.5%. Rein features work well for direct segmentation (78.4% mIoU with an end-to-end decoder) but are worse for *transport-based* label transfer. Rein optimizes features for discriminative segmentation, not for cross-image matching. These are different objectives. The patch size change and reduced token count may also have degraded the OT solver’s ability to form fine-grained correspondences.

#### K.4 Resolution Scaling: The Key Lever

The most effective change was increasing input resolution. More tokens per image means small objects get more patches and more chances to dominate under majority voting.

**Step scaling.** Higher-resolution experiments require proportionally more SAOT training steps. At  $504 \times 1008$  (2,592 tokens), 500 steps suffice. At  $672 \times 1344$  (4,608 tokens), optimal performance requires 32,000 steps. At  $840 \times 1680$  (7,200 tokens), 128,000 steps are needed (performance plateaus; 512K steps give the same 23.1% mIoU). With more tokens, the cost network must learn from a larger set of correspondences, and the Sinkhorn solver converges more slowly on larger coupling matrices.

**The resolution–accuracy tradeoff.** Table 20 shows the tradeoff clearly: higher resolution improves rare/small classes but degrades large/common ones. At  $840 \times 1680$ :

Table 21: Step scaling at  $672 \times 1344$  resolution. Performance improves steadily before plateauing at 32K steps.

Steps	500	1K	2K	4K	8K	32K
mIoU (%)	13.8	15.1	17.4	19.1	20.1	21.6
pole (%)	–	–	9.1	24.6	27.9	29.0
tr.sign (%)	4.4	8.5	15.2	–	–	13.4
bicycle (%)	–	–	–	–	5.5	7.5

- Traffic light goes from 0% to 30.3%, traffic sign from 0% to 29.2%, bicycle from 0% to 22.4%.
- But vegetation drops from 56.5% to 18.6%, car from 62.2% to 29.9%.

Large classes degrade because: (1) the  $k$ -NN budget drops from 128 to 64 at high resolution due to GPU memory, reducing matching coverage; (2) the same supervision budget (20 points/class) is spread thinner across more tokens; and (3) the OT solver converges more slowly on larger coupling matrices.

**Oracle ensemble.** Selecting the best resolution per class gives 29.3% mIoU (+7.6 pp over the best single resolution). The two regimes are complementary, which makes multi-scale transport a natural next step (Section P).

**Motorcycle: the persistent failure.** Motorcycle is the only class that remains at 0% IoU across all resolutions, all backbone variants, and all decoder configurations. It is both very rare (865 source tokens, 0.037%) and visually similar to bicycle and rider at patch granularity. Even at  $840 \times 1680$ , motorcycles do not occupy enough patches to form reliable prototypes, and the learned cost cannot tell them apart from bicycles.

#### K.5 Overtraining at Low Resolution

At the baseline  $504 \times 1008$  resolution, more SAOT training steps actually make things worse. Running 4,000 steps instead of 500 drops mIoU from 23.3% to 16.6%. With only 2,592 tokens per image, the cost network overfits to specific source–target correspondences instead of learning generalizable semantic structure. At higher resolutions, by contrast, 32K–128K steps are needed. The relationship between token count and optimal step count is worth keeping in mind when configuring the pipeline.

## L Why Do Other Methods Handle Rare Classes Better?

End-to-end UDA methods (DAFormer, HRDA, MIC) reach 68–76% mIoU on GTA5→Cityscapes with non-zero IoU on all 19 classes. What are they doing differently?

**Pixel-level operation.** UDA methods operate at full pixel resolution ( $1024 \times 2048$ ), not patch level ( $36 \times 72$ ). A traffic light that occupies 50 pixels has 50 independent prediction sites in a pixel-level model but *zero* dedicated patches in our  $14 \times 14$  grid. This matters more than anything else.

**Learned features.** DAFormer, HRDA, and MIC fine-tune the entire backbone (Swin or MiT-B5) on source data with pixel-level supervision, including losses and augmentations designed for rare classes. Their features are optimized end-to-end for segmentation. We use frozen DINOv2 features trained for general visual representation, not for distinguishing motorcycles from bicycles at patch level.

**Multi-scale context.** HRDA uses multi-resolution inputs ( $512 \times 512$  and  $1024 \times 1024$  crops) and fuses predictions at different scales, giving small objects coverage at high resolution while keeping global context at low resolution. Our transport operates at one scale.

**Self-training with full target access.** UDA methods perform iterative self-training on the full target set with pixel-level pseudo-labels, augmentation (color jitter, mixup, ClassMix), and consistency regularization (MIC’s masked image consistency). Our decoder trains on SAOT pseudo-labels with no augmentation.

**The Rein comparison.** Rein (Wei et al., 2024) achieves 78.4% mIoU by adapting DINOv2 via LoRA with pixel-level supervision on GTA5. It operates at full pixel resolution with a Mask2Former decoder. When we plugged Rein features into our OT framework, performance dropped to 15.5% because Rein features are optimized for discriminative segmentation, not for the cross-image matching that OT needs. Features good for within-image classification are not necessarily good for between-image transport.

## M The Case for SAOT: When 25.7% mIoU Is Enough

Despite the rare-class limitations, SAOT+Decoder has practical advantages worth considering.

**Speed.** At 2.4ms per image (417 FPS), SAOT+Decoder is far faster than end-to-end UDA methods, which need multi-second inference with large backbones. For applications where only the main scene elements matter (road, building, car, vegetation, sky, sidewalk), this is fast enough for real-time use.

**No training infrastructure.** SAOT does not train the backbone. The full pipeline (feature caching + OT learning + decoder training) runs in 6 hours on a single GPU. DAFormer/HRDA/MIC need multi-GPU training for days. If you only have one GPU, SAOT is within reach.

**Strong performance on safety-critical large classes.** For autonomous driving, the classes that SAOT+Decoder handles well (road 90.3%, car 76.2%, building 71.5%, vegetation 59.7%, sidewalk 56.1%, sky 36.9%) are the ones needed for basic drivable-area detection and obstacle avoidance. A vehicle needs to know where the road is, where buildings and vegetation bound the scene, and where other cars are. These classes cover >85% of pixel area in typical Cityscapes images.

**Frozen backbone flexibility.** With a frozen backbone, SAOT can be applied to any new source-target pair by re-running OT learning (~3 hours) and decoder training (~3 hours). No catastrophic forgetting, no feature extractor retraining, and the same cached features can be reused across experiments. The method is also complementary to backbone adaptation: Rein-adapted or fine-tuned features could serve as input to SAOT.

**Theoretical guarantees.** SAOT’s bilevel optimization has provable smoothness properties (Section A.6), unlike heuristic self-training. The transport plan responds predictably to cost changes, and the pipeline is Lipschitz continuous. When predictions are wrong, the transport plan can be inspected to understand why.

## N Limitations of Token-Space Label Transfer

Working with ViT patch tokens introduces hard constraints that better optimization or more data

cannot fix.

**Fixed spatial granularity.** DINOv2 ViT-L/14 has a fixed patch size of 14 pixels. Objects smaller than  $14 \times 14$  pixels disappear under majority voting. At  $504 \times 1008$  input resolution, the spatial grid is  $36 \times 72$ , where each cell covers roughly  $28 \times 14$  pixels in the original Cityscapes image ( $1024 \times 2048$ ). Traffic lights ( $\sim 10$  pixels wide), thin poles ( $\sim 5$  pixels), and distant pedestrians ( $\sim 20$  pixels tall) cannot be resolved at this granularity.

Higher resolution helps (our  $840 \times 1680$  experiments reach 30% IoU on traffic lights), but the cost grows fast:  $2.8\times$  more tokens means  $2.8\times$  larger coupling matrices,  $2.8\times$  more memory, and roughly  $10\times$  more training steps.

**No sub-patch structure.** Each patch token is a single 1,024-dimensional vector summarizing a  $14 \times 14$  region. If a patch contains both road and a traffic light, the token blends both. The transport framework cannot “unblend” this; the token is matched as a whole. Pixel-level methods can assign different labels to different pixels within the same patch.

**Majority voting artifacts.** Ground truth labels are also at patch level via majority voting. A patch that is 45% bicycle and 55% road is labeled “road.” This means the training signal actively teaches the model to ignore minority classes within patches. Even a perfect transport plan would assign “road” to this patch, because that is what the ground truth says.

**Limited receptive field for cross-image matching.** Each token’s feature comes from its local receptive field in the ViT ( $\sim 5$  layers of self-attention give moderate spatial context, but less than a full-image feature). Two tokens in different images can look similar locally but differ in global context. OT operates on individual token features without modeling the global scene structure.

## O Limitations of OT-Based Label Transfer

The OT formulation itself also has limitations for dense label transfer, separate from the token-space constraints above.

**Pairwise matching is not classification.** OT finds correspondences between source and target tokens based on pairwise costs. Classification learns decision boundaries from many examples; OT sees

each source–target pair in isolation (or a few pairs in multi-source aggregation) and must infer labels from a single matching. The decoder aggregates pseudo-labels across many pairs, but the quality ceiling is set by the transport plan.

**Class frequency mismatch.** Unbalanced OT handles mass imbalance at the *image level* (different total mass per image), but not at the *class level*. If road occupies 40% of the source and 50% of the target, UOT can adjust the total mass, but it cannot ensure that the extra 10% target road tokens are matched to road sources rather than sidewalk sources. The semantic prior helps, but it operates on per-class prototypes that may themselves be noisy for rare classes.

**Quadratic scaling.** Even with block-sparse solvers ( $O(nk)$  instead of  $O(nm)$ ), the coupling size grows linearly with  $n$  and  $k$ . At  $840 \times 1680$  resolution ( $n = 7,200$ ) with  $k = 64$ , the coupling has 460K entries per image pair. This is still manageable, but the Sinkhorn solver’s convergence slows with larger problems. Truly high-resolution transport ( $n > 50,000$ ) would need hierarchical or approximate solvers.

**Sensitivity to regularization.** The entropic regularization  $\varepsilon$  controls the trade-off between matching quality and smoothness. Too small: the solver is numerically unstable and slow to converge. Too large: the coupling is too diffuse, and transport mass is spread uniformly rather than concentrating on good matches. The optimal  $\varepsilon$  depends on the scale of the cost function, which shifts as  $\theta$  is learned. Similarly,  $\tau$  controls mass flexibility: too small allows too much mass to be discarded (sparse couplings), too large forces mass conservation (degenerate couplings for mismatched class distributions).

**Bilevel optimization challenges.** The outer loss  $\mathcal{L}(\theta)$  is optimized by differentiating through the Sinkhorn solver. Lemma 1 guarantees Lipschitz smoothness, but in practice the gradients can be noisy: (1) they depend on sparse supervision (20 points per class), (2) truncating Sinkhorn to 30 iterations introduces bias, and (3) the cost network has limited capacity (256-dimensional projections).

## P Future Directions

Several directions follow from these experiments.

**Multi-scale transport.** The oracle ensemble (29.3% mIoU, Section I) shows that baseline and high-resolution predictions are complementary. One could perform transport at multiple resolutions and fuse the results: run SAOT at 2–3 resolutions, upsample all predictions to a common grid, and learn per-class fusion weights. A lightweight MLP trained on validation data would suffice for the fusion.

**Hierarchical or coarse-to-fine solvers.** Rather than a flat  $n \times m$  coupling, a hierarchical solver could first match regions (superpixels or ViT blocks) and then refine within each region. This would shrink the effective problem size while keeping fine-grained matching where it matters, particularly around object boundaries and small objects. Multi-scale OT (Schmitzer, 2019) provides a starting point.

**Pixel-level refinement.** A hybrid approach: use SAOT for coarse patch-level predictions, then refine to pixel level with a lightweight upsampling network (learned bilinear interpolation or a small U-Net). DINOv2’s intermediate features at multiple ViT layers could guide the refinement, keeping patch-level transport efficient while recovering sub-patch detail.

**Adapter-compatible features.** Our Rein experiments showed that features optimized for discriminative segmentation are suboptimal for transport. Training a lightweight adapter (LoRA or prompt tuning) that optimizes DINOv2 features for cross-image matching, rather than within-image classification, could close this gap. The adapter could be trained jointly with SAOT or as a preprocessing step.

**Class-conditional transport.** SAOT currently uses a single transport plan for all classes. Learning separate cost functions per class or per class group (“vehicles,” “infrastructure,” “nature”) would let the solver specialize. Rare classes would benefit from costs tuned to their own feature characteristics instead of being dominated by the frequent classes.

**Active learning for sparse supervision.** We sample 20 random points per class for sparse supervision. An active learning strategy could instead select the most informative target tokens, for example those where the transport plan is most uncertain or where different source images disagree. This

should improve learning efficiency, especially for rare classes where every supervised token counts.

**Temporal consistency in video.** For video segmentation, transport plans from consecutive frames should be temporally consistent. ASOT (Xu and Gould, 2024) explored this for point tracking; extending it to dense label transfer could enable efficient video segmentation from sparse frame-level annotations.

**Beyond synthetic-to-real.** Our evaluation focuses on GTA5→Cityscapes and VOC train→val. Testing on other domain pairs (Synthia→Cityscapes, simulation→real for robotics, medical imaging) would show how well this generalizes. The frozen-backbone approach fits medical imaging well, where labeled data is scarce but foundation models (e.g., BiomedCLIP) exist.

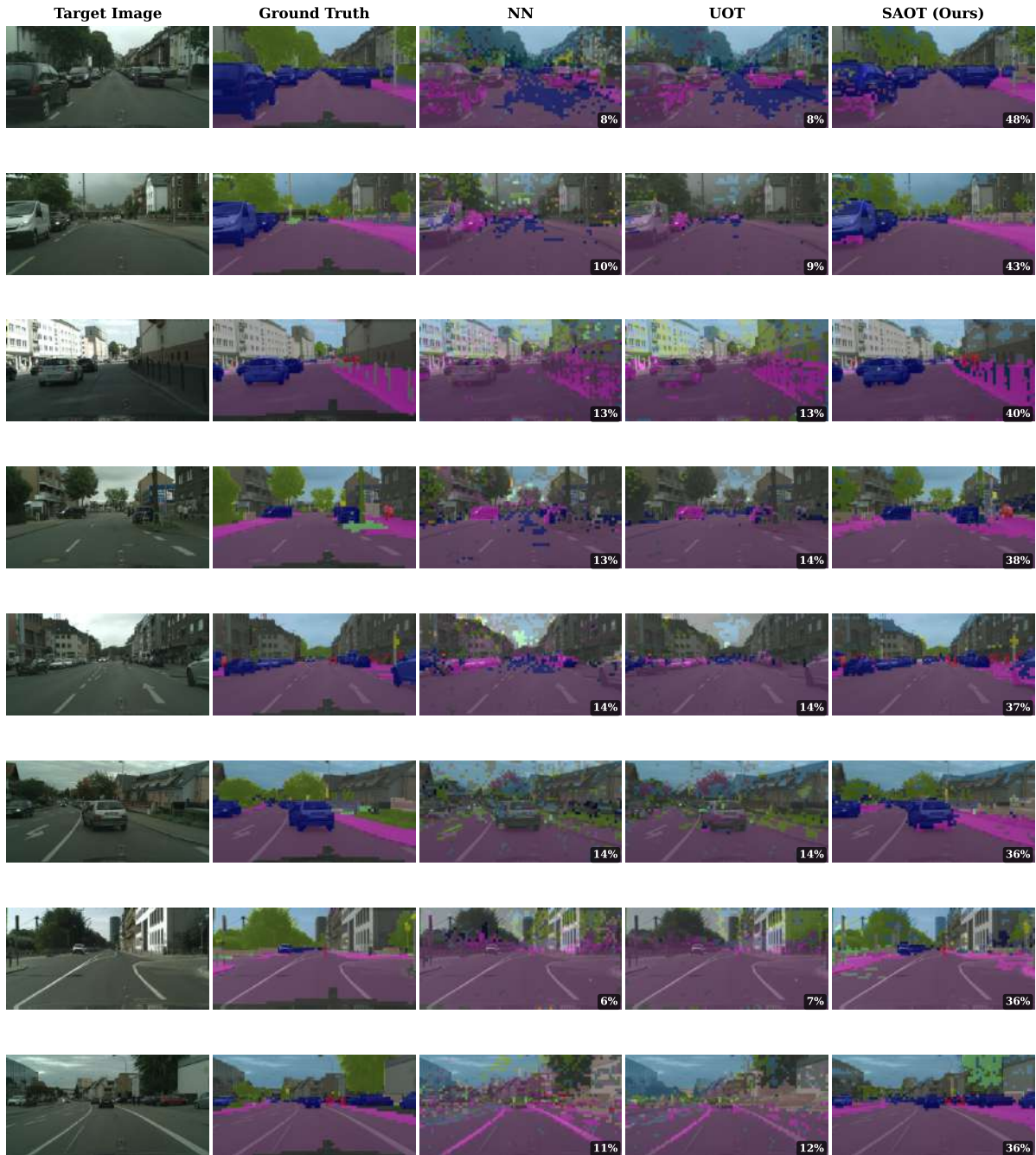


Figure 8: Additional qualitative results (part 1 of 2) on Cityscapes validation images. Columns: target image, ground truth, NN, UOT, SAOT+Decoder. Per-image mIoU (%) is annotated.

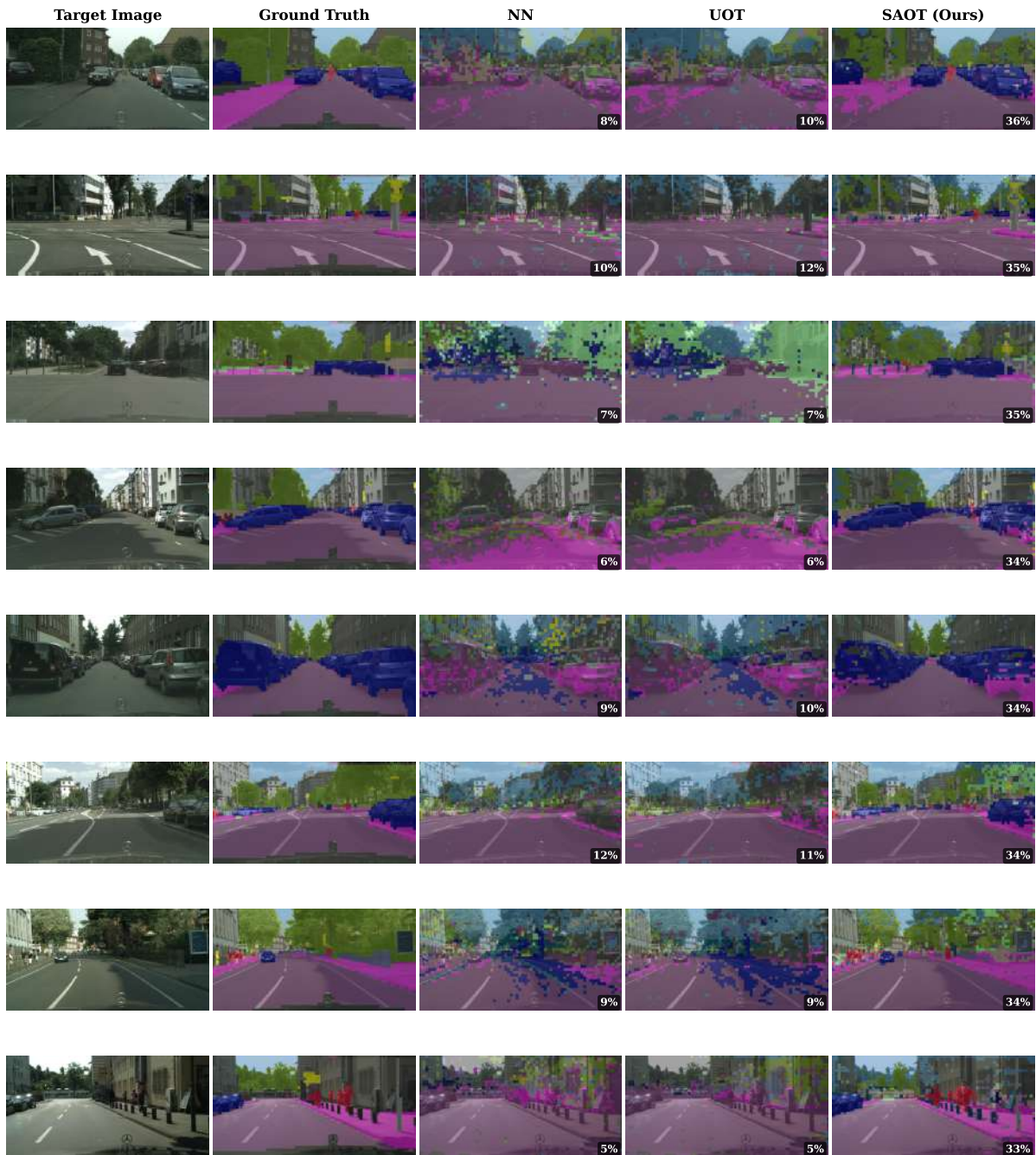


Figure 9: Additional qualitative results (part 2 of 2) on Cityscapes validation images. Same layout as Fig. 8.

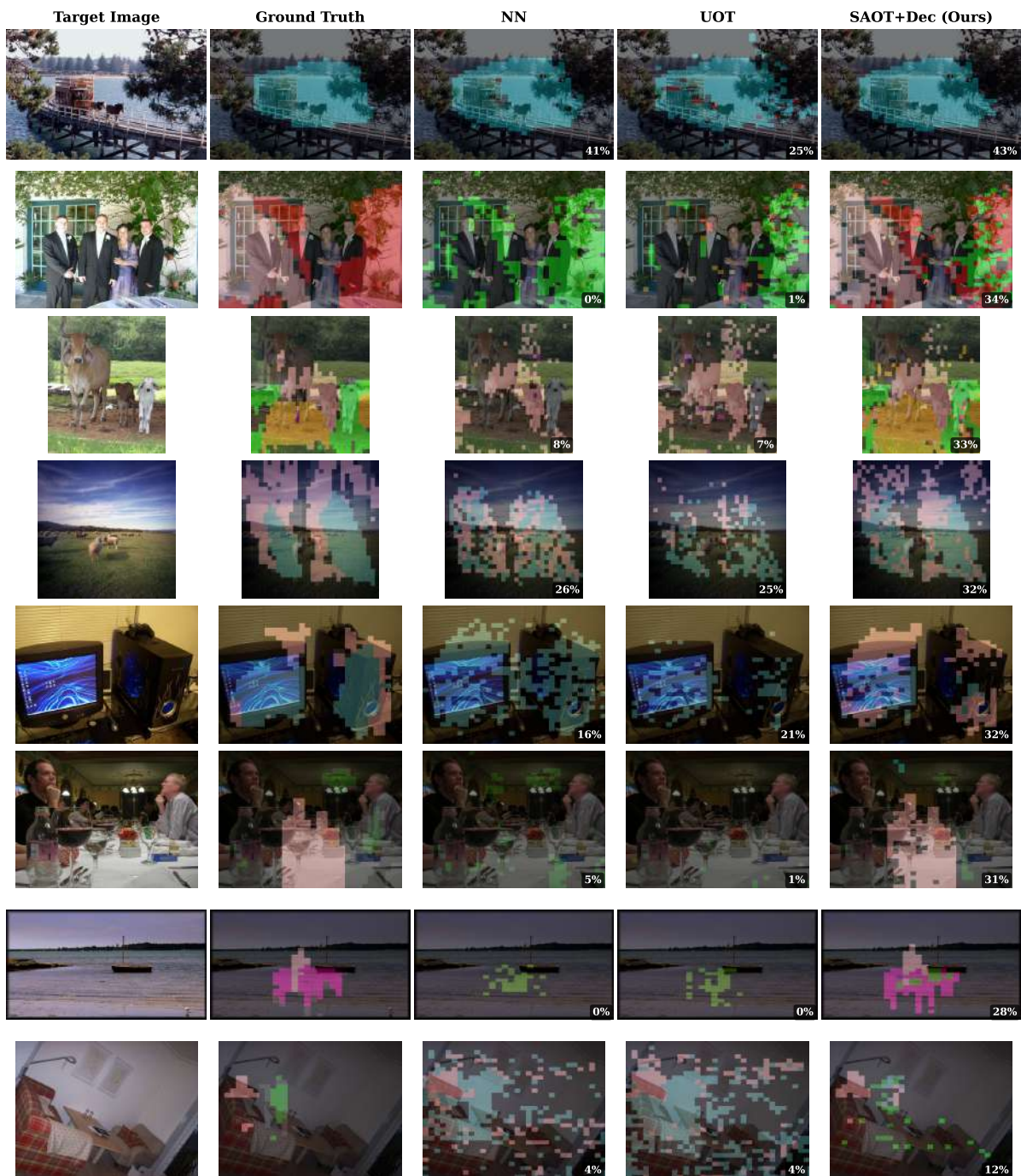


Figure 10: Additional qualitative results on PASCAL VOC 2012 validation images. Columns: target image, ground truth, NN, UOT, SAOT+Decoder. Per-image mIoU (%) is annotated.

# CoSMoEs: Compact Sparse Mixture of Experts

Patrick Huber, Akshat Shrivastava, Ernie Chang, Chinnadhurai Sankar,  
Ahmed Aly and Adithya Sagar  
Meta Reality Labs

## Abstract

Sparse Mixture of Expert (MoE) models are widely used foundation architectures at large scale, yet remain under-explored at smaller sizes. In this work, we introduce Compact Sparse Mixture of Experts (CoSMoEs) for on-device inference, addressing three key challenges: Quality, Memory, and Latency. On the quality front, we conduct a fair evaluation (removing confounding factors) and show that MoE architectures outperform dense models at on-device scale. We further propose weight-decomposed experts, which improve MoE performance beyond the standard formulation. On the memory and latency front, we address the large parameter count of MoE models by improving expert offloading efficiency through a novel training-time loss, reducing inference latency for on-device deployment.

## 1 Introduction

Mixture of Experts (MoEs) have become a popular extension of the transformer architecture (Vaswani et al., 2023). The core idea is that each token in the input sequence is routed through a set of sub-networks, or “experts”, whose outputs are combined via a gating mechanism that determines each expert’s contribution.

While all experts are activated in the most general MoE formulation (Jacobs et al., 1991; Jordan and Jacobs, 1993), *sparse* Mixture of Expert models select only a subset of experts per token (Cai et al., 2024), as used in Qwen (Bai et al., 2023; Yang et al., 2024), OLMoE (Muennighoff et al., 2024), Mixtral (Jiang et al., 2024), and DeepSeek (DeepSeek-AI, 2024). These large-scale MoE models are optimized for highly parallelized server-side inference.

In contrast, this work focuses on small-scale MoEs for edge devices, which face a distinct set of challenges around Quality, Memory, and Latency.

**Quality.** Unlike prior work (e.g., Jiang et al. (2024)), we establish a fair comparison between MoEs and dense models by aligning on active parameters (FLOP-aligned, *FA*) and total parameters (parameter-aligned, *PA*). We further reduce confounding factors by normalizing training data, recipes, and architectures wherever possible. Our evaluation shows that MoE architectures improve average language modeling performance by over 2.3% across model sizes. Building on these results, we propose a novel MoE extension following the intuition of “expert specialization,” yielding up to an additional 1.1% improvement.

**Memory and Latency.** As shown in Figure 1, models trained in server environments face additional constraints when deployed on edge devices. Despite these restrictions being largely architecture-independent, the high total parameter count of modern MoE models severely affect their ability to be deployed on the edge. While the sparsity property of MoE architectures can partially offset the high parameter count through expert offloading, this incurs a large (4-20×) inference latency increase (Xue et al., 2024). To relax this trade-off, we propose a “block-wise expert selection” loss that reduces expert offloads by 6× and improves inference latency by 50%.

## 2 CoSMoEs Models

### 2.1 Sparse Mixture-of-Experts

At the core of this work is the sparse Mixture-of-Expert (MoE) architecture<sup>1</sup>, popularized by GShard (Lepikhin et al., 2020) and Switch Transformers (Fedus et al., 2022). While MoEs can be applied to different parts of the architecture, the most common approach replaces the dense feed-forward layer with a router component and multiple experts. By selecting a discrete subset of experts

<sup>1</sup>When referring to MoE throughout this paper, we assume sparsity.

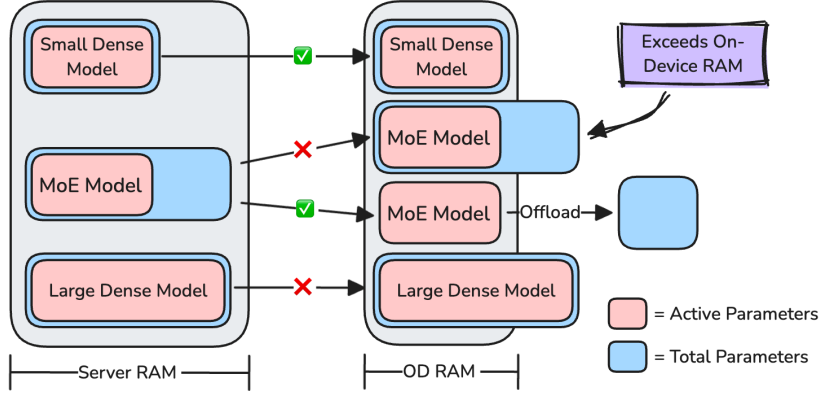


Figure 1: Server-side training environment (left) compared to the memory-constrained inference environment (right), showing deployment restrictions for parameter-heavy MoEs and large dense models on edge devices.

at each step, sparse MoE models are characterized by two quantities: their *active parameters* (FLOPs) and their *total parameters* (model size in memory). For expert selection, different routing paradigms have been proposed, either selecting experts per token (token choice) (Shazeer et al., 2017) or per expert (expert choice) (Zhou et al., 2022). We use token-choice routing, following the findings for text-only models in Muennighoff et al. (2024).

## 2.2 Weight-Decomposed Experts

Standard MoE models have a naturally large total parameter count because each expert maintains a full copy of the feed-forward layer weights. To reduce this overhead, we propose a lightweight expert formulation using matrix weight decomposition (“WD”), inspired by Low-Rank (“LoRA”) adapters (Hu et al., 2021). The key intuition is that each expert is intended to “specialize” on a subset of input tokens, so each expert need not have full representational capacity. We therefore replace the full expert matrices of shape  $n \times m$  with their decomposition into two smaller matrices of shape  $n \times r$  and  $r \times m$ :

$$M_{n \times m} \approx L_{n \times r} \times R_{r \times m} \quad (1)$$

with  $r \ll n$  and  $r \ll m$  (see also Figure 2). In preliminary experiments, we find that setting  $r$  to half the hidden dimension yields the best trade-off between parameter reduction and model performance. To ensure a parameter-aligned comparison, we adjust the number of heads and layers for weight-decomposed models (prefixed with *WD*), as detailed in Section 3.1.

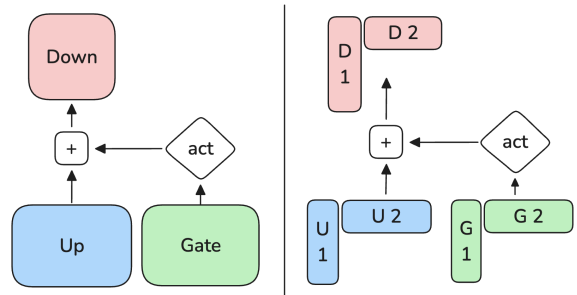


Figure 2: Feed Forward Layer: Standard (left) and Weight-Decomposed (right).

## 2.3 Block-wise Expert Selection

A central challenge for on-device MoE inference is expert offloading: when only active experts are kept in accelerator memory, each change in expert assignments requires transferring parameters between storage and compute, incurring substantial latency. Multiple lines of research have explored inference-time solutions such as predictive offloading and bitwidth adaptations (Yi et al., 2023; Eliseev and Mazur, 2023; Aminabadi et al., 2022). We approach this problem from a different angle: rather than optimizing the offloading mechanism, we reduce the number of offloading events by encouraging *temporally coherent* expert assignments during training.

To this end, we propose a “Block-wise Expert Selection” (BIES) loss term that penalizes frequent expert switches between consecutive tokens, closely related to the expert load balancing loss in Fedus et al. (2022):

Let  $R$  be a router logits tensor with shape  $(B, T, E)$ , where  $B$  is the batch dimension,  $T$  is the sequence length, and  $E$  is the expert dimension. We compute routing weights  $W$  by applying

the softmax function to  $R$ , scaled by a temperature parameter  $\tau$ :

$$W = \text{softmax}(\tau R) \quad (2)$$

In a non-differentiable step, we select the top- $k$  experts  $K$  for each token based on  $W$ . Let  $S$  be the selected experts tensor with shape  $(B, T, K)$ :

$$S = \text{top}_k(W, K) \quad (3)$$

We then count the hard expert replacements  $H$  by comparing consecutive tokens’ expert assignments:

$$H_e = \sum_{b=1}^B \sum_{t=1}^{T-1} |(S_{[b,t+1]} = e) - (S_{[b,t]} = e)| \quad (4)$$

$$H = \sum_{e=1}^E H_e$$

where  $e$  is the expert index and  $S_{[b,t]} = e$  equals 1 if expert  $e$  is among the top- $k$  candidates for token  $t$ . We normalize  $H$  by the batch size, top- $k$ , and number of tokens:

$$H_{norm} = \frac{\lfloor \frac{H}{2} \rfloor}{B \cdot K \cdot (T - 1)} \quad (5)$$

Since  $H$  is non-differentiable, we introduce a continuous surrogate  $L$  that sums the per-expert probability differences between consecutive tokens:

$$L = \sum_{b=1}^B \sum_{t=1}^{T-1} \sum_{e=1}^E |W_{b,t+1,e} - W_{b,t,e}| \quad (6)$$

$$L_{norm} = \frac{L}{B \cdot T}$$

The final loss is the product of the hard and soft expert selection terms:

$$\text{loss} = H_{norm} \cdot L_{norm} \quad (7)$$

To match the sequence-level computation of the BIES loss, we replace the standard (model-level) load balancing loss (Fedus et al., 2022) with **layer-wise load balancing** (following Lin et al. (2024)).

**Layer-wise load balancing.** An important design choice is to compute the load balancing loss *per layer* rather than aggregating across layers. When the loss is computed at the model level, it can be trivially minimized by consistently selecting a single expert per layer—for example, with 3

experts and 3 layers, always selecting expert 0 in layer 0, expert 1 in layer 1 and expert 2 in layer 2 achieves a perfect balance while also minimizing the BIES term. Figure 3 illustrates this failure mode with 3 layers and 3 experts.

		Expert Sequence →				
Layers ↑	L0	0	0	0	1	
	L1	1	1	2	1	
	L2	0	2	2	2	

Layer Avg:					Model Avg:
L0	3/4	1/4	0/4		1/3
L1	0/4	3/4	1/4		1/3
L2	1/4	0/4	3/4		1/3

Figure 3: Example expert selection (for simplicity,  $k=1$ ) for individual layers and the complete model.

## 3 Experimental Setup

### 3.1 Model Configuration

We compare two on-device size categories: “Phone-sized” (1–3B parameters) and “Wearable-sized” (100–300M parameters), across three architectures: Dense, MoE, and WD MoE, as described in Table 1<sup>2</sup>. All models are based on the Llama3 architecture, with an MoE component consisting of eight total experts, two of which are active per token. We follow the standard expert implementation from the Hugging Face codebase (Wolf et al., 2020). Model hyperparameters are kept consistent while aligning on active and total parameters. When trade-offs are necessary, we favor depth over breadth, following Liu et al. (2024).

### 3.2 Training Details

We pre-train all models on the FineWeb Education dataset (FW-edu, Penedo et al. (2024)), a 1.4 trillion token text corpus provided by Hugging Face (Wolf et al., 2020). FW-edu represents a high-quality, general-purpose language dataset.

### 3.3 Evaluation Metrics

We evaluate along three axes. For **language modeling performance**, we use the EleutherAI

<sup>2</sup>The BIES extension uses the MoE architecture and is therefore not listed separately.

Model	Params	L	H	Hid	Seq	Steps	Bsz
<b>Phone-sized models, 1B-3B Parameters</b>							
Dense	1.50B	16	32	2048	2048	310k	2048
MoE	1.37B (3.75B)	24	18	1440	2048	310k	2048
+ WD	1.42B (3.65B)	26	20	1600	2048	310k	2048
Dense	3.61B	28	24	3072	2048	310k	2048
<b>Wearable-sized models, 100-300M Parameters</b>							
Dense	189M	19	8	512	2048	310k	2048
MoE	188M (377M)	19	8	432	2048	310k	2048
+ WD	188M (377M)	32	10	400	2048	310k	2048
Dense	380M	29	12	768	2048	310k	2048

Table 1: On-device model candidates. Params = #Active (#Total) Parameters, L = Layers, H = Self-Attention Heads, Hid = Hidden size, Seq = Sequence length, Bsz = effective batch size.

LM eval harness with nine benchmarks (Gao et al., 2024), following the evaluation protocols of Llama3 (Grattafiori et al., 2024) and MobileLLM (Liu et al., 2024). For **offloading efficiency**, we report the Expert Replacement Ratio (ExRep), which measures the percentage of realized expert replacements, and the optimal expert balance, which computes the average per-layer deviation from a uniform distribution. For **memory and latency**, we report per-token generation speed and peak memory usage.

## 4 Results

### 4.1 Language Modeling Performance

Table 2 presents our language modeling results. A random baseline is shown at the top, followed by our MoE results at phone and wearable scale, with public baselines at the bottom for context.

**Phone-sized models.** All MoE candidates outperform the random baseline by a large margin and consistently improve over the FA dense model by at least 2%. For MMLU and AGI-English, all models show only minor gains, indicating room for further improvement. On the remaining benchmarks, clear improvements are observed. Among MoE variants, the weight-decomposed model performs best overall. We observe a minor performance regression when using the BIES loss. Compared to the PA dense model, MoE candidates perform better on 3 out of 10 metrics, falling only half a percentage point short on average. In the context of previously published models, our MoE candidates outperform the FA Llama 3.2 1B and OLMoE models but do not reach the PA Llama 3.2 3B performance.

**Wearable-sized models.** The wearable-sized evaluation shows similar trends. The weight-decomposed model again achieves the best MoE

performance, here even surpassing the PA dense model. At wearable scale, at least one MoE model outperforms the PA dense model on 6 of 10 tasks. Compared to the published MobileLLM models, we observe improvements at both the 125M and 350M parameter scales. The BIES model again shows a slight performance drop relative to the standard MoE, consistent with the phone-scale findings.

### 4.2 Offload Efficiency

As illustrated in Figure 1, running MoE models on-device often requires offloading experts to stay within memory constraints, at the cost of significant latency increases. Since expert offloading frequency is data-dependent, we use a 100-sample subset of the C4 dataset (Raffel et al., 2020) as a proxy for general text data. Table 3 presents results along three dimensions: the expert replacement percentage (ExRep), the realized inference speed<sup>3</sup>, and expert balancing quality (deviation from uniform expert balance,  $\Delta$ Uniform). The BIES extension achieves over  $6\times$  fewer expert switches than the standard MoE model. This directly translates to a  $1.5\times$  improvement in generation speed, with only a minor regression of less than 1% relative in expert balancing<sup>4</sup>.

**Qualitative analysis.** Figure 4 visualizes expert assignments across 35 tokens for a single layer, comparing the BIES model (top) and the standard MoE model (bottom). The BIES model reduces expert replacements from 21 to 11 while preserving expert diversity—both models actively use 6 out of 8 experts. This illustrates how the BIES loss encourages temporally coherent expert assignments without collapsing to a single expert.

**Layer-wise expert balance.** To understand the per-layer impact of the BIES loss, Figure 5 plots the layer-wise expert balance for both models. With BIES, greater expert divergence is observed in lower layers, whereas the standard MoE model exhibits higher divergence in upper layers. Higher expert diversity in later layers appears preferable, given the general intuition that lower layers encode more local, syntactic information while upper layers capture more global, semantic structures.

<sup>3</sup>Full on-device benchmarks are presented in Section 4.3.

<sup>4</sup>Inference latency improvements are batch-size dependent.

Model	Params	MMLU	AGI-E	Arc-C	Arc-E	BoolQ	PIQA	SIQA	HellaS	OBQA	WinoG	Avg
<b>Random Baseline</b>												
Random	–	24.53	16.07	21.08	25.25	51.07	51.74	33.11	26.31	29.40	50.83	32.94
<b>Phone-sized models, ~1B-3B Parameters</b>												
Dense	1.50B	24.78	17.99	36.95	74.03	59.08	74.54	41.76	59.88	41.20	57.54	48.78
MoE	1.37B (3.75B)	<u>25.96</u>	17.65	42.58	76.77	60.89	75.52	42.12	65.07	42.40	<u>62.35</u>	51.13
+ BIES	1.37B (3.75B)	25.40	17.50	41.55	<u>77.02</u>	62.81	76.06	41.91	63.14	42.60	59.04	50.70
+ WD	1.42B (3.65B)	23.90	<b>18.20</b>	43.69	76.81	<b>66.76</b>	76.39	<b>45.14</b>	66.51	42.80	62.04	52.22
Dense	3.61B	<b>26.41</b>	16.82	<b>44.54</b>	<b>77.9</b>	65.87	<b>77.48</b>	43.3	<b>67.18</b>	<b>45.00</b>	<b>63.46</b>	<b>52.80</b>
<b>Wearable-sized models, ~100-200M Parameters</b>												
Dense	189M	22.9	16.82	23.29	56.82	57.09	64.15	37.82	36.36	32.8	50.99	39.90
MoE	188M (377M)	<b>25.27</b>	17.37	27.9	<u>63.09</u>	<u>58.39</u>	69.04	39.61	44.09	<b>34.4</b>	53.03	43.22
+ BIES	188M (377M)	24.27	17.58	24.83	58.84	<b>59.82</b>	66.49	38.64	39.70	33.40	49.96	41.35
+ WD	188M (377M)	23.64	17.16	<u>28.58</u>	62.58	57.13	<b>69.31</b>	<b>40.28</b>	<u>46.15</u>	33.20	<b>54.38</b>	<b>43.24</b>
Dense	380M	24.79	<b>17.86</b>	<b>28.92</b>	<b>64.35</b>	52.02	69.21	39.97	<b>46.53</b>	33.80	51.62	42.91
<b>Public Baselines across Model Sizes</b>												
MobLLM (2024)	135M	23.02	17.45	19.97	46.38	60.34	64.96	38.08	38.17	28.40	52.57	38.93
MobLLM (2024)	350M	26.33	17.47	23.89	56.4	61.96	68.88	39.87	49.57	31.00	57.38	43.28
Llama3.2 (2024)	1.4B	36.92	18.80	31.31	65.40	63.61	74.54	42.84	47.74	26.20	60.06	46.70
Llama3.2 (2024)	3.6B	54.01	22.53	42.32	74.41	72.81	76.71	47.13	55.32	31.20	69.30	54.50
OLMoE (2024)	1.68B (6.92B)	25.74	17.19	40.87	74.20	60.52	74.70	44.37	60.38	38.40	58.72	49.50

Table 2: Model comparison on zero-shot LM evaluations. Params = #Active (#Total) Parameters, BIES = Block-wise Expert Selection, WD = Weight-Decomposed, MobLLM = MobileLLM. Public baselines are evaluated using the EleutherAI LM eval harness (2024).

Model	ExRep (↓)	Tok/s Gen (↑)	ΔUni (↓)
MoE	43.82	15.02	<b>9.60</b>
+ BIES	<b>6.55</b>	<b>23.10</b>	9.67

Table 3: Impact of the BIES loss on expert replacement ratio (in percent), generation speed (tokens/second), and deviation from the uniform expert distribution (in percent). ↓ = lower is better, ↑ = higher is better.

### 4.3 On-Device Benchmarks

Since on-device models often execute in CPU-based environments or on proprietary accelerators, we compare model latency on both CPU and GPU<sup>5</sup>. As this paper targets training-time improvements, we use standard inference code from the Hugging Face Transformers library (Wolf et al., 2020) and the gpt-fast codebase (PyTorch Labs, 2023) without inference-specific optimizations such as EdgeMoE (Yi et al., 2023). Table 4 presents results along four dimensions: (1) language modeling performance (from Table 2), (2) inference speed across 128 tokens on CPU and GPU, (3) peak memory after 128 token generations, and (4) suitability for on-device deployment. In addition to the previously described model candidates, we evaluate standard MoE offloading as “Offl,” where only active experts are kept on the accelerator.

**Latency.** On CPU, the FA dense model achieves the highest generation rate; MoE candidates are slightly slower, and the PA dense model is 2× slower. On GPU, MoE models generally produce

<sup>5</sup>Evaluations are performed in a server environment; actual on-device accelerator numbers may vary.

fewer tokens per second than dense models, primarily due to their deeper architecture (Table 1). We also observe the 1.5× speed-up between standard offloaded MoE models and BIES offloaded models, consistent with Table 3. For context, inference-based offloading strategies such as those in Eliseev and Mazur (2023) and Aminabadi et al. (2022) achieve 2–3× and 5.5× latency reductions, respectively, at comparable model sizes. However, unlike our training-time approach, inference-time offloading methods often require additional modeling components to predict future expert usage, which can be impractical for on-device scenarios.

**Peak Memory.** Without expert offloading, the peak generation memory of MoE candidates is comparable to the PA dense model. With offloading, peak memory drops to the level of the FA dense model, since only active parameters are kept in memory. This makes offloaded MoE models viable on-device candidates (see ✓).

### 4.4 Ablation Studies

Going beyond the standard MoE setup with two active and eight total experts, we ablate these dimensions and explore their impact on quality, latency, and memory. We evaluate eight model configurations spanning a range of active and total expert counts. Figure 6 summarizes findings along the active expert (left) and total expert (right) dimensions. For active expert ablations, we fix the total number of experts to 8; for total expert ablations, we fix the number of active experts to 2.

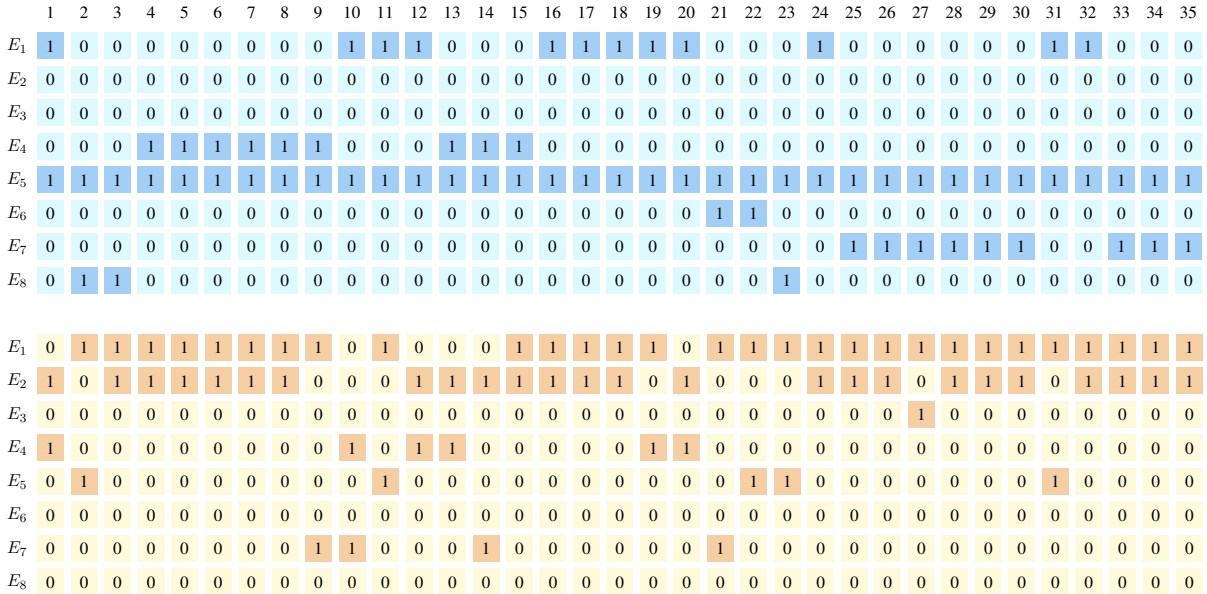


Figure 4: Expert assignments across 35 tokens for a single layer. 1 = Active, 0 = Inactive. Top: BIES (11 replacements), Bottom: Standard MoE (21 replacements).

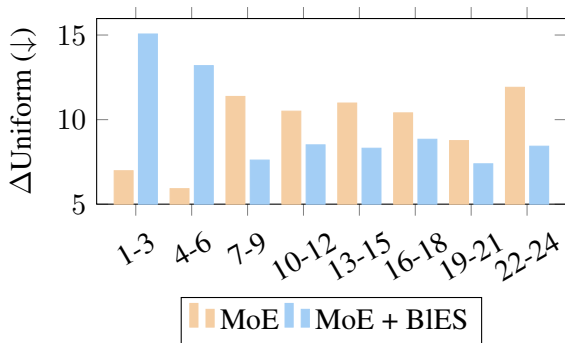


Figure 5: Per-layer divergence of expert routing from the uniform distribution. Large values indicate expert collapse toward a pseudo-dense layer.

**Active Expert Ablation.** Increasing the number of active experts improves model quality, though returns diminish as the count approaches 8. Generation speed decreases linearly with the number of active experts, while peak memory remains constant<sup>6</sup>.

**Total Expert Ablation.** Model quality increases roughly linearly with the number of total experts, though the improvement is less pronounced than for active experts. Generation speed is unaffected since FLOPs are held constant. However, total expert count significantly impacts peak memory<sup>7</sup>.

In summary, increasing either active or total ex-

<sup>6</sup>Peak memory would increase if experts were actively offloaded.

<sup>7</sup>Peak memory would remain constant with active offloading, at the cost of reduced generation speed.

Model	LM Eval	Latency		Mem	
Setup	Avg	Gen (tok/sec)		Gen	
Metric	%	CPU	GPU	GB	☐ / ✗
Dense	48.78	4.47	73.10	5.8	☑
MoE	51.13	4.30	40.60	14.7	✗
+ WD	52.22	3.85	33.50	14.2	✗
+ Offl	51.13	4.30	15.02	5.4	☑
+ BIES	50.70	4.30	23.10	5.4	☑
Dense	52.80	1.77	42.60	14.0	✗

Table 4: On-device benchmarks. Gen = Generation of 128 tokens (1 token prefill), Offl = Offloaded, BIES = Block-wise Expert Selection. Mem = Peak GPU memory. ☐ = Phone-sized, assuming <6GB of RAM use (e.g., iPhone 12 Pro).

perts improves model quality but requires a trade-off in either latency or memory.

#### 4.5 Training Efficiency

Figure 7 compares the training dynamics of MoE and dense model candidates, aligned by dataset, training steps, and hyperparameters. We track average language modeling performance at checkpoints from 10k to the full 310k steps.

Comparing the FA and PA dense models with our best-performing MoE model, we corroborate the findings of Lin et al. (2024), observing a 5–10× training efficiency gain for MoE models over their FA dense counterparts. Our MoE candidate reaches the best performance of the 1.4B dense model at around 35k steps, while the larger 3.6B dense model achieves generally higher scores throughout training.

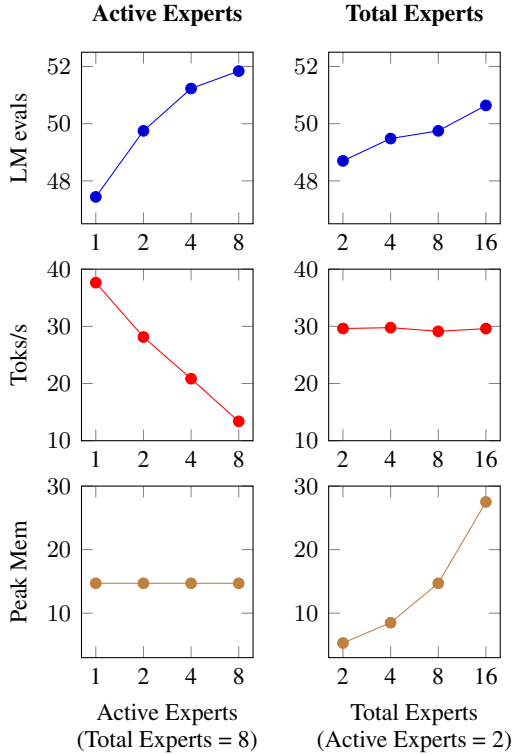


Figure 6: Active (left) and total (right) expert ablations of the 1.4B MoE model at 50k steps (~210B tokens).

## 5 Related Work

**Small-Scale Language Models.** As foundation models become increasingly expensive to train and deploy, two research directions have emerged. The first focuses on improving small-scale architectures, including MobileLLM (Liu et al., 2024), MobiLlama (Thawakar et al., 2024), and the BabyLlama series (Timiryasov and Tastet, 2023; Tastet and Timiryasov, 2024). The second targets training data quality through cleaner, more curated pipelines, as demonstrated by the Microsoft Phi series (Abdin et al., 2024) and Hugging Face efforts (Ben Allal et al., 2024; Lozhkov et al., 2024). For a comprehensive survey, see Nguyen et al. (2024).

**Sparse Mixture of Experts.** Sparse MoEs span a wide range of model sizes: Qwen (2023; 2024) and OLMoE (2024) at 1–3B active parameters, Mixtral (2024) and DeepSeek (2024) at around 7B, and DBRX (Databricks, 2023) and Grok-1 (x.ai, 2023) at 36B and 86B active parameters, respectively. The OLMoE paper (Muennighoff et al., 2024) is particularly relevant, presenting training insights and design decisions for MoE models at smaller scales; we follow many of its findings. Comparing the inner workings of large MoE models, Lo

et al. (2024) analyze Mixtral, Grok, and DeepSeek, finding initial similarities despite different training paradigms. We pursue similar comparisons but prioritize fairness by controlling confounding factors. For a detailed MoE survey, see Cai et al. (2024).

### Weight Decomposition for Mixture of Experts.

Along similar lines to our weight-decomposed experts, Dou et al. (2024) proposed a LoRA-style extension that converts dense networks into MoE models during supervised fine-tuning (SFT). By freezing the dense backbone and introducing a router at the SFT stage, their approach aims to reduce catastrophic forgetting of pre-training knowledge. In contrast, we apply weight decomposition directly during pre-training, training more parameter-efficient experts from the start.

**Inference Efficiency.** Our BIES loss is complementary to prior work on inference-time offloading optimization. Xue et al. (2024) improve expert pre-fetching and caching to reduce parameter transfers. EdgeMoE (Yi et al., 2023) enhances offloading through predictive strategies and bitwidth adaptations. Other frameworks include Mixtral Fast Inference (Eliseev and Mazur, 2023) and DeepSpeed Efficient Inference (Aminabadi et al., 2022). Unlike these approaches, our method reduces the number of offloading events during training rather than optimizing the offloading mechanism at inference time, making the two approaches orthogonal and potentially complementary.

## 6 Conclusion

We demonstrate how to enable sparse MoE architectures for on-device inference along three key dimensions: Quality, Memory, and Latency. On the quality front, a fair comparison shows that MoE models outperform their dense counterparts on language modeling tasks by over 2.35%. Our weight-decomposed experts yield further gains of up to 1.1% over standard MoE models. To make MoE models practical for on-device deployment, we address the offloading bottleneck by reducing expert switches during training. Our block-wise expert selection loss improves offloading efficiency by 6× and increases generation speed by 50% compared to standard offloaded MoE models. These results pave the way for deploying MoE architectures in on-device scenarios, supporting high-quality, privacy-preserving foundation models for edge devices.

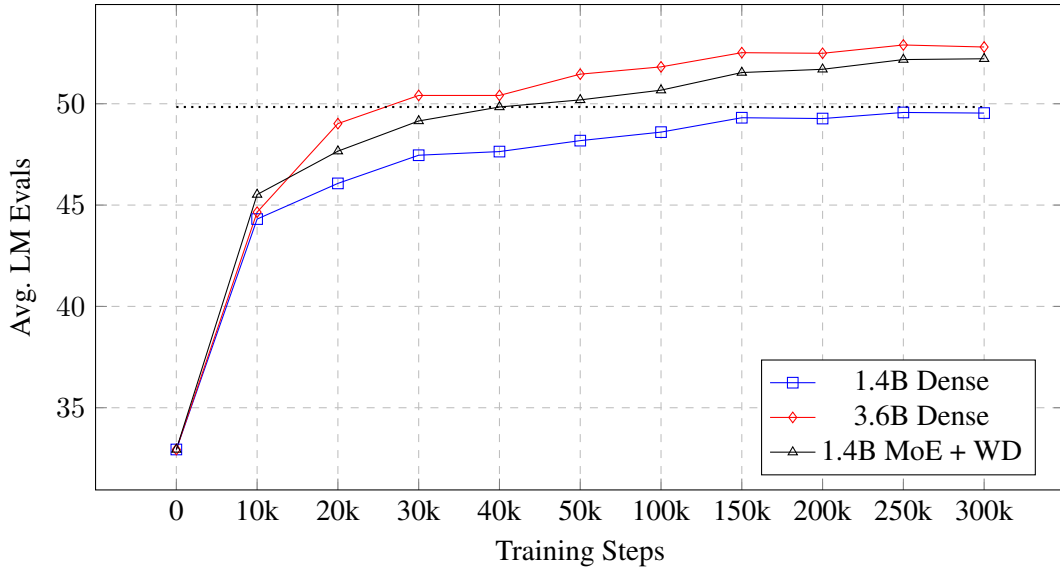


Figure 7: Training dynamics across model candidates. The dotted line marks the best 1.4B Dense checkpoint performance, reached by the MoE + WD model at  $\sim 35$ k steps.

## Limitations

We note two limitations of this work.

First, while we aim for a fair comparison between MoE and dense models, a perfectly balanced comparison remains elusive. Aligning active and total parameters necessarily introduces variation in model depth and width, which affects certain metrics (e.g., latency depends heavily on layer count). Additionally, keeping hyperparameters aligned may inadvertently favor one architecture. Thorough hyperparameter sweeps would help address this but would introduce their own confounding factors.

Second, our evaluations and benchmarks are performed on server hardware, which introduces a level of abstraction from actual edge devices. Although we aim to present a complete picture of MoE models for on-device use, this gap between server and edge environments is a limitation of our current evaluation.

## References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek

Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacrose, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. [Phi-3 technical report: A highly capable language model locally on your phone.](#)

Reza Yazdani Aminabadi, Samyam Rajbhandari, Minjia Zhang, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Jeff Rasley, Shaden Smith, Olatunji Ruwase, and Yuxiong He. 2022. [DeepSpeed inference: Enabling efficient inference of transformer models at unprecedented scale.](#)

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei

- Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. [Qwen technical report](#).
- Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. [Smolm-corus](#).
- Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. 2024. [A survey on mixture of experts](#).
- Databricks. 2023. [Introducing dbrx: A new state-of-the-art open llm](#). Blog post.
- DeepSeek-AI. 2024. [Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model](#).
- Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. [Loramoe: Alleviate world knowledge forgetting in large language models via moe-style plugin](#).
- Artyom Eliseev and Denis Mazur. 2023. [Fast inference of mixture-of-experts language models with offloading](#).
- William Fedus, Barret Zoph, and Noam Shazeer. 2022. [Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity](#).
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. [A framework for few-shot language model evaluation](#).
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Al-lonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimppoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kam-badur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Rapparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Del-pierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand,

- Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khan-delwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelen, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiao Cheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. [The llama 3 herd of models](#).
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#).
- Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. [Adaptive mixtures of local experts](#). *Neural Computation*, 3(1):79–87.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L el io Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Th eophile Gervet, Thibaut Lavril, Thomas Wang, Timoth e Lacroix, and William El Sayed. 2024. [Mixtral of experts](#).
- M.I. Jordan and R.A. Jacobs. 1993. [Hierarchical mixtures of experts and the em algorithm](#). In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, volume 2, pages 1339–1344 vol.2.
- Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020.

- Gshard: Scaling giant models with conditional computation and automatic sharding.
- Xi Victoria Lin, Akshat Shrivastava, Liang Luo, Srinivasan Iyer, Mike Lewis, Gargi Ghosh, Luke Zettlemoyer, and Armen Aghajanyan. 2024. [Moma: Efficient early-fusion pre-training with mixture of modality-aware experts.](#)
- Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. 2024. [Mobilellm: Optimizing sub-billion parameter language models for on-device use cases.](#)
- Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. 2024. [A closer look into mixture-of-experts in large language models.](#)
- Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. [Fineweb-edu: the finest collection of educational content.](#)
- Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. 2024. [Olmoe: Open mixture-of-experts language models.](#)
- Chien Van Nguyen, Xuan Shen, Ryan Aponte, Yu Xia, Samyadeep Basu, Zhengmian Hu, Jian Chen, Mihir Parmar, Sasidhar Kunapuli, Joe Barrow, Junda Wu, Ashish Singh, Yu Wang, Jiuxiang Gu, Franck Dernoncourt, Nesreen K. Ahmed, Nedim Lipka, Ruiyi Zhang, Xiang Chen, Tong Yu, Sungchul Kim, Hanieh Deilamsalehy, Namyong Park, Mike Rimer, Zhehao Zhang, Huanrui Yang, Ryan A. Rossi, and Thien Huu Nguyen. 2024. [A survey of small language models.](#)
- Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. [The fineweb datasets: Decanting the web for the finest text data at scale.](#)
- PyTorch Labs. 2023. [Gpt-fast.](#) GitHub repository.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer.](#) *Journal of Machine Learning Research*, 21(140):1–67.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. [Outrageously large neural networks: The sparsely-gated mixture-of-experts layer.](#)
- Jean-Loup Tastet and Inar Timiryasov. 2024. [Babyllama-2: Ensemble-distilled models consistently outperform teachers with limited data.](#)
- Omkar Thawakar, Ashmal Vayani, Salman Khan, Hisham Cholakkal, Rao M. Anwer, Michael Felsberg, Tim Baldwin, Eric P. Xing, and Fahad Shahbaz Khan. 2024. [Mobillama: Towards accurate and lightweight fully transparent gpt.](#)
- Inar Timiryasov and Jean-Loup Tastet. 2023. [Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty.](#)
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. [Attention is all you need.](#)
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Huggingface’s transformers: State-of-the-art natural language processing.](#)
- x.ai. 2023. [Grok os.](#) Blog post.
- Leyang Xue, Yao Fu, Zhan Lu, Luo Mai, and Mahesh Marina. 2024. [Moe-infinity: Offloading-efficient moe model serving.](#)
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. [Qwen2 technical report.](#)
- Rongjie Yi, Liwei Guo, Shiyun Wei, Ao Zhou, Shang-guang Wang, and Mengwei Xu. 2023. [Edgemoe: Fast on-device inference of moe-based large language models.](#)
- Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. 2022. [Mixture-of-experts with expert choice routing.](#)

# GraphicWeaver: Benchmarking Agentic Planning for Graphic Design Generation

Dayeon Ki ✨\* Tianyi Zhou ✨ Marine Carpuat ✨  
Gang Wu 🍃 Puneet Mathur 🍃 Viswanathan Swaminathan 🍃

University of Maryland ✨ Adobe Research 🍃  
dayeonki@umd.edu

## Abstract

Vision-language model (VLM)-powered agents are increasingly enabling new forms of automation across various human tasks. While prior work has primarily focused on well-defined problems with explicit goals, the capabilities of agents in creative graphic design, where goals are inherently open-ended and subjective, remain largely underexplored. To bridge this gap, we introduce GraphicWeaver, a planning benchmark for graphic design comprising 1,079 diverse user queries and associated images spanning four design categories. Comprehensive experiments with six models reveal that current VLM-based agents struggle to handle such complex planning tasks, which require taking into account both explicit design constraints specified in queries and implicit commonsense design principles. We attribute these failures to challenges in (1) retrieving appropriate parameters for tool usage, (2) understanding spatial relationships across design components, and (3) coordinating dependencies across agents. We envision GraphicWeaver as a challenging yet valuable testbed for advancing VLM agent planning in creative design contexts.<sup>1</sup>

## 1 Introduction

Recent advancements in Vision-Language Models (VLMs) have expanded their potential as general-purpose agents capable of automating a wide range of human tasks. Prior work has evaluated VLM agents in diverse domains, including web navigation and interaction (Zheng et al., 2024; He et al., 2024a; Tian et al., 2025), travel planning (Xie et al., 2024; Jandial et al., 2025), item detection (Kelly et al., 2024), embodied scenarios (Zheng et al., 2023), and online shopping (Koh et al., 2024).

On the other hand, research on the planning capabilities of VLM agents for *creative* design tasks

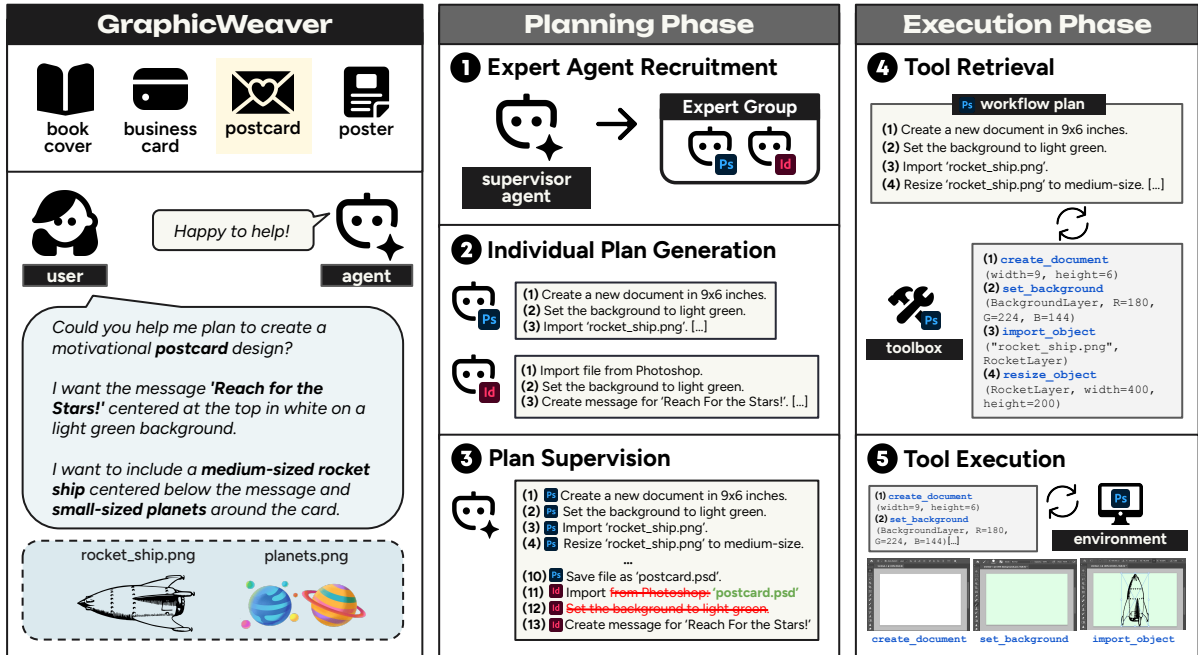
remains limited, primarily due to underspecified open-ended goals from users (Guo et al., 2024; Ge et al., 2025). They require delicate planning that translates a high-level user request into a structured plan composed of executable sub-tasks that collectively produce the final design. This is inherently complex, posing multiple challenges: (1) A complex design often requires collaboration involving multiple agents; (2) Design planning is usually *long-horizon*, involving a sequence of decisions for expert selection, tool calls, and tool uses, with an expansive tool space to explore (Xie et al., 2024); (3) A design plan must accommodate both explicit constraints from user queries (e.g., “the title text color must be white”) and implicit constraints inferred through commonsense reasoning (e.g., “the background should contrast with the color of text elements”) since user queries are often incomplete with unspecified details (Qian et al., 2024b); (4) Assessing design outcomes is inherently subjective, as the notion of what constitutes a *better* design vary across individuals. These challenges raise a key question: To what extent can VLM agents generate cohesive plans for creative design tasks when provided only with open-ended user queries?

We focus on graphic design, a task that remains challenging even for humans as it demands specialized knowledge of professional design tools, often requiring substantial time and effort to learn (Bedford et al., 2006). In this work, we introduce **GraphicWeaver**, a planning benchmark comprising 1,079 realistic user queries paired with associated images. The dataset spans four representative design categories: book cover, business card, postcard, and poster, chosen to capture a broad range of design goals varying in layout composition, textual arrangement, and overall visual organization (§3).

We comprehensively evaluate six VLM-based agents, ranging from smaller open-weight models to larger closed-source ones (§4). As shown in Figure 1, each agent is assessed on its ability to reason

\*Work done during internship at Adobe Research.

<sup>1</sup><https://github.com/dayeonki/graphicweaver>



**Figure 1: Overview of GraphicWeaver.** Given a user query and associated image(s), vision-language agents collaboratively *plan* and *execute* the requested task. They generate a plan that satisfies the user’s requirements expressed in the query while adhering to implicit commonsense design principles. Based on the finalized plan, each agent retrieves the appropriate tools and executes them within its respective environment.

about task requirements and devise actionable plans (Planning; §5.2) and to retrieve and execute appropriate tools to achieve the desired goal (Execution; §5.3). Our key findings are summarized as follows:









- All evaluated VLM-based agents struggle with complex planning tasks in GraphicWeaver, particularly in integrating both explicit user-specified requirements and implicit commonsense design principles into the generated plans.
- Execution success rates are generally low, with GPT-4.1 (the highest performing model) successfully executing only 62.5% of its generated plans. The resulting design outcomes receive low scores for both alignment with user queries and images, as well as for overall creativity.
- Further analyses reveal three recurring failure modes: (1) retrieval of invalid parameters for tool usage, (2) difficulty in reasoning about spatial relationships of design components, and (3) inadequate coordination across agents.

## 2 Related Work

**VLM-Based Agents.** Leveraging the strengths of Large Vision-Language Models (LVLMs), VLM-based agents have shown strong performance in automating human tasks through tool use (Schick et al., 2023; Qin et al., 2023) and advanced reasoning strategies (Yao et al., 2022; Shinn et al., 2023).

Further inspired by human collaboration and its role in improving work efficiency (O’Reilly et al., 1997; Woolley et al., 2015), recent research has proposed multi-agent frameworks in which multiple agents coordinate to solve a shared task (Ding et al., 2023; Shen et al., 2023; Dong et al., 2024; Chen et al., 2024). In particular, studies suggest that assigning specialized roles to agents improves their effectiveness on complex problems (Li et al., 2023; Talebirad and Nadiri, 2023; Du et al., 2024; Hong et al., 2024; Qian et al., 2024a). Similarly, we evaluate VLM agents in a collaborative setting, but in the context of an underexplored problem in this space: graphic design generation.

**Graphic Design Generation.** Graphic design is a form of visual art that combines multimodal elements (e.g., images, texts, and vector symbols) to create aesthetic compositions that effectively communicate user’s intent (Cheng et al., 2024). Most prior work has examined specific design *sub-tasks*—including layout generation (Li et al., 2019; Gupta et al., 2021; Jiang et al., 2023), typography generation (Zhao et al., 2018; Jiang et al., 2019), attribute recognition (Lin et al., 2024), and colorization (Yuan et al., 2021; Qiu et al., 2023)—as well as *single-shot* generation (Hsu et al., 2023; Seol et al., 2024; Yang et al., 2024b). In contrast,





Category	Example User Query	Example Image(s)	# Train/Test
 Book Cover	We need to create a cooking book cover design titled 'Quick & Easy Weekday Meals' with the author's name 'Sarah James' at the top center, above the title, and a subtle green background. The title should be in white, top center, below the author's name, and feature a delicious pasta plate centered in the middle, below the title text. Include the tagline 'Delicious Recipes for Busy Lives' in white, below the title.		5/260
 Business Card	Create a business card design in teal background for a software company named 'OceanSoft'. Please include the company name in huge white font at the top center. I want to include the tagline 'Sailing to Success' in medium white font placed below the company name and a large wave icon at the bottom center.		5/203
 Postcard	Could you draft a plan to create a wedding announcement postcard design with the message 'Save the Date' centered at the top in white on a navy blue background? I want to include medium-sized golden wedding rings centered below the message and small elegant florals in the corners.		5/260
 Poster	I need to create a poster design for a music festival named 'Rhythm Beats' on a green background, featuring a large illustration of a colorful guitar in the center, a catchy title 'Rhythm Beats' in huge white font at the top center, and a tagline 'Feel the Music in Your Soul' in large white font at the bottom center.		5/336

**Table 1: Actual examples of user queries and images in GraphicWeaver.** Each user query is paired with one or more relevant images that serve as inputs to the design planning process. Train split examples are used as in-context examples.

GraphicWeaver evaluates VLM agents on their ability to plan end-to-end designs through multi-step generation across multiple agents and web-based environments, more closely mirroring real-world graphic design practices (Inoue et al., 2024).

### 3 GraphicWeaver




#### 3.1 Overview

We introduce GraphicWeaver, a benchmark for evaluating VLM agents on complex planning and tool use across multiple web-based design environments. To reflect real-world design planning, GraphicWeaver incorporates diverse constraints, including those explicitly specified in the user queries (Figure 1) and those arising from commonsense design principles, such as choosing background colors that contrast with text elements. The benchmark comprises 1,079 query-image pairs across four graphic design categories:  book cover,  business card,  postcard, and  poster. It is split into training and test sets: the training set contains 5 queries per category with human-annotated reference plans (20 in total), and the test set contains the remaining 1,059 queries, with detailed examples and statistics reported in Table 1.

#### 3.2 Benchmark Construction Pipeline

This section outlines the construction pipeline of GraphicWeaver, which consists of the following steps: environment setting, diverse user query design, image pairing, and human quality checking. All prompts are outlined in Appendix D.1.

**Environment Setting.** In GraphicWeaver, we construct a static, closed sandbox environment for

evaluation, ensuring that all VLM agents operate under the same fixed set of resources and eliminate any variability for fair comparison. It also avoids the overhead of building and maintaining custom environments, which is particularly challenging in graphic design domains. We consider three complementary design environments:  Adobe Photoshop,  Illustrator, and  InDesign, each offering distinct capabilities. For each environment, we build a database with 46 tools in which every tool is linked to an executable JavaScript code corresponding to a single mouse or keyboard operation (e.g., creating a new document) (He et al., 2024b) and parameterized only by predefined input fields. As summarized in Table 2, we group tools into four categories: basic operations, drawing, text-related, and object manipulation functions.<sup>2</sup> For additional details on the toolset and environment configuration, refer to Appendix B.1.

**User Query Construction.** To construct diverse queries for GraphicWeaver, we identify key design components and sub-components for each design category, as summarized in Table 3. We begin by collecting 20 screenshots of design projects (5 per category) shared by practitioners on the Behance platform,<sup>3</sup> ensuring that the benchmark reflects authentic, real-world design needs. Three graduate students familiar with Adobe software then collaboratively write realistic user queries and corresponding execution plans, and carry out these plans within our sandbox environment to produce final designs that closely resemble the reference screen-

<sup>2</sup>We derive the tools from Adobe’s official tutorials, reflecting commonly used operations among design practitioners.

<sup>3</sup><https://www.behance.net/>

Category	Tool	Input Parameters	Description	Env.
Basic	SetBackgroundColor	red, green, blue	Set the background color to desired RGB color.	Ps Ai Id
	SaveDocument	fileName, format	Save the current document into desired format.	Ps Ai Id
Drawing	OpacityDrawing	layerName, opacity	Adjust opacity of a drawing.	Ai
	ResizeDrawing	layerName, width, height	Resize a drawing to desired width and height.	Ai
Text	ApplyFont	layerName, fontName	Apply font to text.	Ps Ai Id
	RotateText	layerName, angle	Rotate text to desired angle.	Ps Ai Id
Object	ImportObject	fileName, layerName	Import an image or object from file path.	Ps Ai Id
	PhotoFilter	layerName, filterType, density	Apply a photo filter to an object with desired density.	Ps

**Table 2: Subset of available tools in GraphicWeaver.** Each tool is parameterized by predefined input fields for execution. **Env.:** Environment which supports the execution of a specific tool. Full list of available tools is in Appendix Table 6.

Category	Design Components	Required?
📖 Book Cover	Background color	✓
	Title (content, size, color, position)	✓
	Author Name (content, size, color, position)	✓
	Subtitle (content, size, color, position)	✗
	Tagline (content, size, color, position)	✗
	Image (size, position, image URL, caption)	✓
📄 Business Card	Background color	✓
	Brand Name (content, size, color, position)	✓
	Tagline (content, size, color, position)	✗
	Contact Details (content, size, color, position)	✗
	Image (size, position, image URL, caption)	✓
✉ Postcard	Background color	✓
	Message (content, size, color, position)	✓
	Image (size, position, image URL, caption)	✓
📜 Poster	Background color	✓
	Title (content, size, color, position)	✓
	Tagline (content, size, color, position)	✗
	Image (size, position, image URL, caption)	✓

**Table 3: Key design components and sub-components for each design category.** Sub-components are listed in parentheses. **Required?:** Whether the component is required in the user query during query construction process.

shots (see Appendix B.2 for more details).

For each design category, the identified design components are used as placeholders to construct query skeletons, which serve as prompt templates (Qian et al., 2024b; Xie et al., 2024; Yoran et al., 2024). We then prompt GPT-4 (Achiam et al., 2023) to randomly fill these placeholders and manually introduce additional variation in query openers (e.g., “Please help me create a design [...]”, “Could you provide me a design [...]”) to better reflect the range of natural phrasing in real user queries, as illustrated in Table 1.

Directly using model-generated queries often results in many with highly similar design concepts (i.e., multiple postcards themed around “Birthday”). To diversify this, we remove near-duplicates by (1) discarding queries with overlapping bi-grams in any design components, and (2) filtering out semantically similar pairs with SentenceBERT similarity scores above 0.8 (Reimers and Gurevych, 2019).

**Image Pairing.** Each validated user query includes a brief description of the image(s) required for the design (see Figure 1). To ground these descriptions in concrete visual assets, we construct an image retrieval pool by collecting vector illustrations from OpenCLIPArt<sup>4</sup> and Public Domain Vectors.<sup>5</sup> In total, we gather 274K caption-image URL pairs as our retrieval pool. For each image, we then retrieve the top-3 candidates whose captions have the highest SentenceBERT similarity scores with the query’s image description.<sup>6</sup>

**Quality Control.** We first conduct an automatic evaluation to assess the quality of the user queries and the top-3 retrieved images. For each query, we prompt GPT-o1<sup>7</sup> to: (Q1) identify the key design components and rate how well each contributes to the coherence of the final design on a five-point Likert scale (1:Not at all, 5:Completely), and (Q2) rank the three retrieved image candidates from 1 (best fit) to 3 (least fit) based on their relevance.

To validate these automatic annotations, we conduct a human verification study on a stratified random sample of 200 user queries (50 per design category) using the same criteria. We observe substantial agreement between GPT-o1 and human judgments (Cohen’s  $\kappa=0.586$  for Q1 and Kendall’s  $\tau=0.671$  for Q2). We discard queries in which any design component receives a rating of 1 or 2 and retain only the image ranked as the best fit. Further details are provided in Appendix B.3.

### 3.3 Evaluation

We evaluate both the plans and the execution outcomes offered by agents along multiple dimensions.

<sup>4</sup><https://openclipart.org/>

<sup>5</sup><https://publicdomainvectors.org/>

<sup>6</sup>All images will be released under the Creative Commons Zero (CC0) license. The average text/image counts per query are: book covers (3.05/1.00), business cards (2.15/1.33), postcards (1.03/1.28), and posters (1.99/1.04).

<sup>7</sup><https://openai.com/o1/>

Detailed prompts are provided in Appendix D.2.

### (1) Planning Evaluation

- **Delivery Rate:** This assesses whether agents can successfully deliver a final plan within a limited number of steps, determined by difficulty: Easy (1 expert, 10 steps), Medium (2 experts, 20 steps), Hard (3 experts, 30 steps).<sup>8</sup> Plans that exceed the limit are counted as failures (Xie et al., 2024).
- **Design Pass Rate:** This measures if the plan correctly reflects both *explicit* user-specified constraints and *implicit* commonsense principles. We prompt GPT-5<sup>9</sup> to score color, text, and imagery alignment on a five-point Likert scale.
- **Step Efficiency:** This metric represents the proportion of unique (non-duplicate) to total steps.
- **Expert Use Efficiency:** This metric captures how effectively a plan minimizes switching between expert agents. For a plan  $p$  with  $N$  steps and  $E$  unique experts:

$$\text{Eff.}(p) = \frac{E - 1}{\max(1, \sum_{i=1}^N \mathbb{1}(\text{expert}_i \neq \text{expert}_{i-1}))} \quad (1)$$

### (2) Execution Evaluation

- **Execution Success Rate:** This measures the proportion of plans executable without errors.
- **Fidelity:** This metric captures whether the required user images appear in the final outcome, measured via template matching (opencv).
- **Content Similarity:** Semantic alignment between the user query and the final outcome, measured using CLIPScore (Hessel et al., 2021).
- **VQA Pass Rate:** This metric measures how well the final design outcome reflects the components specified in the user query, using Visual Question-Answering (VQA) (Agrawal et al., 2016). For each query, we use GPT-4 to generate component-based questions,<sup>10</sup> and then answer them with LLAVA-1.5 13B (Liu et al., 2024). Pass rate is the average Yes/No accuracy (Zhao et al., 2024).
- **Creativity:** Following Torrance (1966); Runco and Jaeger (2012), we assess Originality (uniqueness) and Elaboration (extent to which the design expands on the user query by adding meaningful details) on a five-point Likert scale using GPT-5.

<sup>8</sup>Step limits are based on human-annotated plans.

<sup>9</sup><https://openai.com/gpt-5/>

<sup>10</sup>On average, 9.07, 10.0, 7.89, 8.70 questions are generated per user query for book covers, business cards, postcards, and posters, respectively. Examples are in Appendix B.4.

## 4 Experiment Setup

**Models.** We focus on VLMs with input context lengths of at least 32K tokens due to the extensive information required for planning. We evaluate four open-weight models of varying sizes and families: QWEN-2.5-VL 7B and 32B (Bai et al., 2025), GEMMA-3 12B and 27B (Team et al., 2025), and two closed-source models: GPT-O4-MINI<sup>11</sup> and GPT-4.1.<sup>12</sup> We use temperature of 0.0.<sup>13</sup>

**Planning Strategies.** To study the impact of agentic planning, we compare two strategies: **direct** and **agentic**. In the **direct** mode, a single VLM agent generates the entire design plan without invoking the multi-step planning process illustrated in Figure 1. In the **agentic** mode, we adopt a hierarchical framework where a supervisor VLM agent  $a_s$  coordinates a group of expert agents  $a_i \in \mathcal{A}$  for planning (Fourney et al., 2024; Zhang et al., 2025).

Specifically, for each user query, the supervisor agent  $a_s$  assembles an expert group  $\mathcal{A}$  based on predefined role descriptions and assigns a high-level goal to each expert agent  $a_i \in \mathcal{A}$ . We instantiate three design experts, each aligned with one of our three design environments (§3.2) and prompted with distinct expertise and responsibilities:

- **Photo Editor:** An agent with an expertise in **Ps** Adobe Photoshop, responsible for image editing, color correction, and applying filters.
- **Vector Graphic Editor:** An agent with an expertise in **Ai** Adobe Illustrator, focused on creating and editing vector illustrations.
- **Layout Designer:** An agent with an expertise in **Id** Adobe InDesign, responsible for customizing layout templates, exporting files, and integrating text with visual elements.

Each expert VLM agent  $a_i$  then proposes its own plan  $p_i$  conditioned on its assigned goal. To emulate human problem-solving process (Zhu et al., 2023), each  $a_i$  is instructed to plan with a sequence of actionable steps (Yang et al., 2024a; Wu et al., 2024b; Zheng et al., 2025), which further facilitates accurate tool retrieval (Huang et al., 2024). The supervisor then overlooks these individual plans in terms of the overall goal and merges them into a single cohesive plan  $p_s$ , which is then executed sequentially, yielding the final design outcome.

<sup>11</sup><https://openai.com/o3-and-o4>

<sup>12</sup><https://openai.com/index/gpt-4-1/>

<sup>13</sup>HuggingFace model names are in Appendix A.

Model	Planning				Execution					
	Delivery Rate (%)	Design Pass Rate (%)	Step Eff.	Expert Use Eff.	Success Rate (%)	Fidelity	Content Similarity	VQA Pass Rate (%)	Creative (O)	Creative (E)
<i>Direct mode</i>										
QWEN-2.5-VL 7B	27.1	22.5	45.5	0.98	10.3	0.01	5.70	16.0	1.05	0.64
QWEN-2.5-VL 32B	<u>57.3</u>	19.7	56.0	<b>1.00</b>	17.5	0.04	6.62	19.0	<u>1.18</u>	1.07
GEMMA-3 12B	37.7	19.0	63.8	0.99	14.9	0.03	6.93	13.6	<u>1.18</u>	0.76
GEMMA-3 27B	<b>67.8</b>	<u>24.7</u>	70.9	<b>1.00</b>	22.3	0.09	<u>9.30</u>	17.7	1.09	1.10
GPT-04-MINI	46.2	23.3	<u>79.5</u>	<b>1.00</b>	28.0	<u>0.11</u>	8.70	<u>21.5</u>	1.14	<u>1.13</u>
GPT-4.1	30.3	<b>30.1</b>	<b>82.5</b>	<b>1.00</b>	<b>29.5</b>	<b>0.13</b>	<b>13.1</b>	<b>23.3</b>	<b>1.39</b>	<b>1.36</b>
<i>Agentic mode</i>										
QWEN-2.5-VL 7B	15.2	51.8	92.0	<b>1.00</b>	39.4	0.15	22.5	37.6	1.77	1.59
QWEN-2.5-VL 32B	<u>39.6</u>	49.7	92.4	<b>1.00</b>	61.3	<u>0.20</u>	22.3	35.5	1.89	1.65
GEMMA-3 12B	27.3	<u>54.5</u>	96.5	<b>1.00</b>	58.1	0.17	21.0	36.2	<u>2.01</u>	1.68
GEMMA-3 27B	<b>51.4</b>	54.4	93.1	<b>1.00</b>	55.4	<u>0.20</u>	26.7	<u>44.7</u>	1.98	<u>2.04</u>
GPT-04-MINI	20.8	51.1	<u>97.1</u>	<b>1.00</b>	56.5	<u>0.20</u>	28.3	<b>44.9</b>	1.78	1.99
GPT-4.1	13.9	<b>56.8</b>	<b>94.7</b>	<b>1.00</b>	<b>62.5</b>	<b>0.21</b>	<b>29.3</b>	<u>44.7</u>	<b>2.02</b>	<b>2.06</b>

**Table 4: Planning and execution results of different VLMs and planning strategies on GraphicWeaver.** For each planning mode, column-wise best scores are **bolded** and second-best scores are underlined. All metrics are interpreted as higher values for better performance ( $\uparrow$ ). Scores are aggregated over the four design categories; per category results are provided in Appendix C.1.

## 5 Results

We discuss the performance of various VLM agents across planning strategies (§5.1) on GraphicWeaver in terms of planning (§5.2) and execution (§5.3).

### 5.1 Direct vs. Agentic Mode

As shown in Table 4, all VLM agents perform worse in the **direct** mode than in the **agentic** mode on nearly all metrics, with the exception of delivery rate. We attribute the higher delivery rate in **direct** mode to the fact that directly generated plans are typically shorter, and thus more easily satisfy the maximum number of steps constraint. Expert use efficiency also remains close to perfect in the **direct** mode, since most models tend to persist with a single expert agent throughout the entire plan rather than switching between experts.

Other planning and execution metrics are substantially lower in the **direct** mode, with the largest gap reaching over 30%. For instance, design pass rate drops from 53.1% (agentic) to 23.2% (direct) on average, step efficiency from 94.3% to 66.3%, and execution success rate from 55.5% to 20.4%. Together, these results highlight the importance of an agentic framework for complex design planning, suggesting that decomposing the task and coordinating specialized experts is more effective than asking VLMs to plan in one-shot.

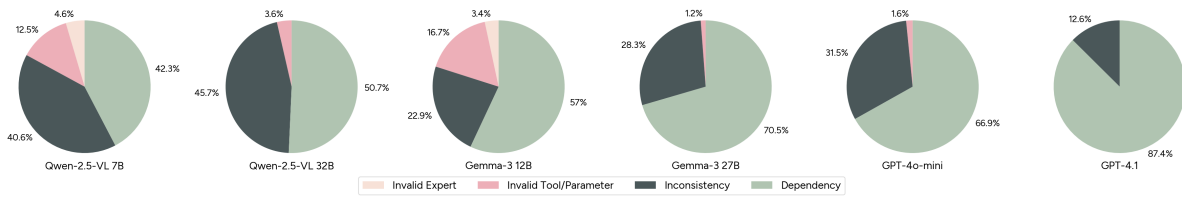
### 5.2 Planning

In the **agentic** mode, all evaluated VLM agents still struggle to deliver plans within the step limit on GraphicWeaver. GEMMA-3 27B achieves the highest rate at only 51.4%, with all other models remaining below 40%. Design pass rates are also modest, indicating difficulty in jointly satisfying explicit user-specified requirements and implicit commonsense design constraints: GPT-4.1 achieves the highest design pass rate at 56.8%, while QWEN-2.5-VL 32B scores the lowest (49.7%). In contrast, step efficiency and expert use efficiency remain relatively high, suggesting that the generated plans contain only a few redundant steps and avoid unnecessary switching between expert agents.

Across the four design categories, poster plans achieve the highest design pass rate (54.8%), whereas business card plans exhibit the lowest delivery and design pass rates. This is likely due to the higher number of text elements and, for some queries, the added complexity of planning designs for both the front and back of the card. Detailed per-category results are reported in Appendix C.1.

### 5.3 Execution

Execution metrics are also low across all evaluated VLM agents. The proportion of plans that fully execute ranges from 39.4% for QWEN-2.5-VL 32B to at most 62.5% for GPT-4.1. Even among successfully executed plans, the resulting design outcomes



**Figure 2: Error distribution for each VLM.** For all tested models, majority of the errors stem from inconsistent planning across design components (Inconsistency) or failing to resolve dependencies across steps in design plans (Dependency).

receive low scores on measures of alignment with both user queries and images. For example, fidelity to the user images remains around 0.10-0.20 across models, with the best score of 0.21 achieved by GPT-4.1. Semantic similarity between user queries and final designs is likewise limited, ranging from 21.0% for GEMMA-3 12B to a maximum of 29.3% for GPT-4.1. None of the VLMs achieve a VQA pass rate above random chance (50% for Yes/No accuracy), with the highest score being 44.9% for GPT-4o-MINI. Additionally, the overall creativity of the generated design outcomes, as judged by GPT-5, remains low on both originality (O) and elaboration (E), with all models scoring roughly between 1.5 and 2.0 on these scales.

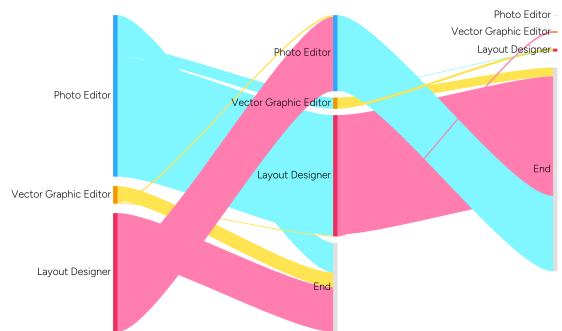
Taken together, these results show that GraphicWeaver presents a substantial challenge for current VLM agents: even the state-of-the-art VLMs, when equipped with agentic planning strategy, still fall short of planning and successful execution in complex graphic design tasks.

## 6 Analysis

We present further analysis on the **agentic** mode.

### 6.1 Expert & Tool-Use Analysis

We visualize the distribution and flow of expert agents in the generated plans in Figure 3. On average, 2.11 expert agents are recruited per user query. Across models, the Photo Editor and Layout Designer are most frequently paired, reflecting their complementary roles, whereas the Vector Graphic Editor is used less often since most queries already specify images to incorporate into the planning. The Layout Designer is responsible for the largest share of steps (15.2 on average), followed by the Vector Graphic Editor (10.4) and the Photo Editor (8.24). In terms of agent ordering, models exhibit clear preferences: the most common transition is Photo Editor → Layout Designer (43.3% of transitions), followed by Layout Designer → Photo Editor (21.5%) and plans that use only the Layout



**Figure 3: Aggregated expert agent usage ordering.** Most common order is Photo Editor → Layout Designer.

Designer accounting for 17.8% of cases.

As further detailed in Appendix C.2, expert agents consistently rely on only a small subset of their available tools: the Photo Editor primarily performs object manipulation, while Layout Designer focuses on text operations. The most frequent tool usage sequences closely resemble workflows in human-annotated references (e.g., document creation → set background color → import and manipulate images → text), indicating that previously observed human-like reasoning patterns (Wei et al., 2023) possibly extend to graphic design planning.

### 6.2 Error Analysis

We categorize step-level errors in the generated plans into four types, as illustrated in Figure 2: (1) **Invalid Expert**, where a non-existent agent is assigned; (2) **Invalid Tool/Parameter**, where tools outside the defined toolset or non-defined parameters are selected; (3) **Inconsistency**, where a step’s assumptions conflict with design components, and (4) **Dependency**, where an operation references an object that is unavailable or uninitialized. We summarize our main findings as follows:

1. **Dependency errors dominate**, accounting for 62.5% of all errors. These include both *local* (within-agent) and *global* (across-agent) dependencies. All models particularly struggle with

global dependencies, often failing to correctly reference objects instantiated by other agents. We also observe that the proportion of dependency errors among all errors tends to increase for larger, closed-source models.

2. **Inconsistency errors are also common**, accounting for 30.3% of errors across all tested models. These typically occur when a plan contains mutually incompatible assumptions about the target design—for example, first specifying that an element should be centered, then later introducing steps that place other elements in the same position. This is consistent with previously reported spatial reasoning limitations of VLMs (Yamada et al., 2024; Wu et al., 2024a).
3. **Invalid tool or parameter errors** are particularly common in smaller, open-weight models (the QWEN-2.5-VL and GEMMA-3 families). These errors often stem from hallucinated tools (e.g., `CreateTextAndResize`) or incorrect tool choices (e.g., mapping “*Move the title to the center.*” to `RepositionObject` instead of `RepositionText`), even when the full toolset is provided, underscoring the need for more reliable tool retrieval mechanisms.

Overall, VLM agents primarily struggle with global dependency handling and spatial reasoning, highlighting the need for more robust reasoning over design components and improved dependency resolution approaches.

## 7 Conclusion

We introduce GraphicWeaver, a planning benchmark grounded in real-world graphic design needs, to assess the complex design planning and tool-use capabilities of current vision-language agents. Our comprehensive evaluation of six VLMs show that GraphicWeaver remains highly challenging even for state-of-the-art models with an agentic planning strategy: agents struggle both to reason over user queries and to devise actionable plans, as well as to retrieve and execute appropriate tools to achieve the target design outcomes. Error analysis further reveals systematic weaknesses in tool selection, understanding spatial relationships across design components, and, most critically, recognizing and handling global dependencies across agents.

We envision GraphicWeaver as a stepping stone toward building more capable graphic design assistants, grounded in realistic tasks and constraints.

We hope our work spurs future research on advancing planning in more open-ended creative tasks.

## 8 Limitation

Our study comes with certain limitations:

- GraphicWeaver assumes a scenario in which user queries explicitly specify the text and image elements, as well as the precise attributes such as color and text position. However, in realistic settings, users may not always specify or even know exactly what images to include in a design, or they may express their requests at a very high-level (Ge et al., 2025). Future works can explore scenarios where user input is limited, requiring models to seek clarification through interactions with users (Qian et al., 2024b; Li et al., 2024).
- The number of tools available in GraphicWeaver is currently limited to a fixed set of 46, as each corresponding JavaScript code was manually written by the authors. This set is not exhaustive of all possible tools within the three design environments. Future works can investigate automated methods for dynamically generating and retrieving tools (Yuan et al., 2024) or integrate a retrieval-augmented generation module (Lewis et al., 2021) into tool retrieval pipeline to enable agents make more informed decisions. Additionally, GraphicWeaver is currently limited to four design categories (book cover, business card, postcard, poster) and only considers three design expert agents, which does not cover the full range of design variants in graphic design tasks.
- Our experiments primarily focus on evaluating the performance of VLMs’ capabilities in design planning. Supporting Large Language Models (LLMs) would require a different setup, including prompting with image captions instead of raw images, which is available in GraphicWeaver. Moreover, our agents operate in a stateless planning environment, relying only on the user query and the provided toolset. This allows us to isolate and evaluate the VLMs’ planning capabilities, independent of environmental feedback. We view extending our work to support LLMs and perceptual tools as a valuable future direction.
- The scope of our experiments is constrained by computational and API budgets: we focus on open-weight models that can be run locally and closed-source models within our cost limits. Consequently, our findings may not fully generalize to other model families or larger-scale models.

## Acknowledgments

We would like to thank our collaborators at Adobe Research for their valuable feedback, including Alexa Siu, Tong Sun, Saayan Mitra, and Stefano Petrangeli. Dayeon is especially grateful for the Adobe Research intern cohort for making the internship experience memorable, including Nishant Balepur, Dang Nguyen, Vishakh Padmakumar, Paiheng Xu, Hyunji Lee, and Yoonjoo Lee.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. 2016. *Vqa: Visual question answering*. *Preprint*, arXiv:1505.00468.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. *Qwen2.5-vl technical report*. *Preprint*, arXiv:2502.13923.
- Tim Bedford, John Quigley, and Lesley Walls. 2006. *Expert elicitation for reliable system design*. *Statistical Science*, 21(4).
- Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje Karlsson, Jie Fu, and Yemin Shi. 2024. *Autoagents: a framework for automatic agent generation*. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI '24*.
- Yutao Cheng, Zhao Zhang, Maoke Yang, Hui Nie, Chunyuan Li, Xinglong Wu, and Jie Shao. 2024. *Graphic design with large multimodal model*. *Preprint*, arXiv:2404.14368.
- Shiyang Ding, Xinyi Chen, Yan Fang, Wenrui Liu, Yiwu Qiu, and Chunlei Chai. 2023. *Designgpt: Multi-agent collaboration in design*. *Preprint*, arXiv:2311.11591.
- Yubo Dong, Xukun Zhu, Zhengzhe Pan, Linchao Zhu, and Yi Yang. 2024. *Villageragent: A graph-based multi-agent framework for coordinating complex task dependencies in minecraft*. *Preprint*, arXiv:2406.05720.
- Zhuoyun Du, Chen Qian, Wei Liu, Zihao Xie, Yifei Wang, Yufan Dang, Weize Chen, and Cheng Yang. 2024. *Multi-agent software development through cross-team collaboration*. *Preprint*, arXiv:2406.08979.
- Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. 2024. *Magentic-one: A generalist multi-agent system for solving complex tasks*. *Preprint*, arXiv:2411.04468.
- Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and Trevor Darrell. 2025. *Autopresent: Designing structured visuals from scratch*. *Preprint*, arXiv:2501.00912.
- Yuxuan Guo, Shaohui Peng, Jiaming Guo, Di Huang, Xishan Zhang, Rui Zhang, Yifan Hao, Ling Li, Zikang Tian, Mingju Gao, Yutai Li, Yiming Gan, Shuai Liang, Zihao Zhang, Zidong Du, Qi Guo, Xing Hu, and Yunji Chen. 2024. *Luban: Building open-ended creative agents via autonomous embodied verification*. *Preprint*, arXiv:2405.15414.
- Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. 2021. *Layouttransformer: Layout generation and completion with self-attention*. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1004–1014.
- Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024a. *Webvoyager: Building an end-to-end web agent with large multimodal models*. *arXiv preprint arXiv:2401.13919*.
- Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024b. *WebVoyager: Building an end-to-end web agent with large multimodal models*. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890, Bangkok, Thailand. Association for Computational Linguistics.
- Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. *CLIPScore: A reference-free evaluation metric for image captioning*. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. *Metagpt: Meta programming for a multi-agent collaborative framework*. *Preprint*, arXiv:2308.00352.

- HsiaoYuan Hsu, Xiangteng He, Yuxin Peng, Hao Kong, and Qing Zhang. 2023. Posterlayout: A new benchmark and approach for content-aware visual-textual presentation layout. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6018–6026.
- Tenghao Huang, Dongwon Jung, Vaibhav Kumar, Mohammad Kachuee, Xiang Li, Puyang Xu, and Muhao Chen. 2024. [Planning and editing what you retrieve for enhanced tool learning](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 975–988, Mexico City, Mexico. Association for Computational Linguistics.
- Naoto Inoue, Kento Masui, Wataru Shimoda, and Kota Yamaguchi. 2024. [Opencole: Towards reproducible automatic graphic design generation](#). *Preprint*, arXiv:2406.08232.
- Surgan Jandial, Yinong Oliver Wang, Andrea Bajcsy, and Fernando De la Torre. 2025. [On the fine-grained planning abilities of VLM web agents](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25347–25380, Suzhou, China. Association for Computational Linguistics.
- Shuhui Jiang, Zhaowen Wang, Aaron Hertzmann, Hailin Jin, and Yun Fu. 2019. Visual font pairing. *IEEE Transactions on Multimedia*, 22(8):2086–2097.
- Zhaoyun Jiang, Jiaqi Guo, Shizhao Sun, Huayu Deng, Zhongkai Wu, Vuksan Mijovic, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. 2023. [Layoutformer++: Conditional graphic layout generation via constraint serialization and decoding space restriction](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18403–18412.
- Chris Kelly, Luhui Hu, Bang Yang, Yu Tian, Deshun Yang, Cindy Yang, Zaoshan Huang, Zihao Li, Jiayin Hu, and Yuexian Zou. 2024. [Visiongpt: Vision-language understanding agent using generalized multimodal framework](#). *Preprint*, arXiv:2403.09027.
- Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. 2024. [VisualWebArena: Evaluating multimodal agents on realistic visual web tasks](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905, Bangkok, Thailand. Association for Computational Linguistics.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). *Preprint*, arXiv:2005.11401.
- Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. [Camel: communicative agents for "mind" exploration of large language model society](#). In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.
- Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. 2019. [Layoutgan: Generating graphic layouts with wireframe discriminators](#). *arXiv preprint arXiv:1901.06767*.
- Shuyue Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S. Ilgen, Emma Pierson, Pang Wei Koh, and Yulia Tsvetkov. 2024. [Mediq: Question-asking LLMs and a benchmark for reliable interactive clinical reasoning](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Jieru Lin, Danqing Huang, Tiejun Zhao, Dechen Zhan, and Chin-Yew Lin. 2024. [Designprobe: A graphic design benchmark for multimodal large language models](#). *Preprint*, arXiv:2404.14801.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. [Improved baselines with visual instruction tuning](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306.
- Charles O'Reilly, Katherine Phillips, and Sigal Barsade. 1997. [Group demography and innovation: Does diversity help?](#) *Research on managing groups and teams*, 1.
- Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024a. [ChatDev: Communicative agents for software development](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand. Association for Computational Linguistics.
- Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024b. [Tell me more! towards implicit user intention understanding of language model driven agents](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1113, Bangkok, Thailand. Association for Computational Linguistics.
- Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. [Toollm: Facilitating large language models to master 16000+ real-world apis](#). *arXiv preprint arXiv:2307.16789*.
- Qianru Qiu, Xueting Wang, and Mayu Otani. 2023. [Multimodal color recommendation in vector graphic documents](#). In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4003–4011.

- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Mark A Runco and Garrett J Jaeger. 2012. The standard definition of creativity. *Creativity research journal*, 24(1):92–96.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.
- Jaeyung Seol, Seojun Kim, and Jaeyun Yoo. 2024. [Posterllama: Bridging design ability of language model to contents-aware layout generation](#). In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXXII*, page 451–468, Berlin, Heidelberg. Springer-Verlag.
- Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. [Hugging-gpt: Solving ai tasks with chatgpt and its friends in hugging face](#). *Preprint*, arXiv:2303.17580.
- Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: Language agents with verbal reinforcement learning](#). *Preprint*, arXiv:2303.11366.
- Yashar Talebirad and Amirhossein Nadiri. 2023. [Multi-agent collaboration: Harnessing the power of intelligent llm agents](#). *Preprint*, arXiv:2306.03314.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenaly, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Pettrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szepktor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shrivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Pöder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.
- Shulin Tian, Ziniu Zhang, Liangyu Chen, and Ziwei Liu. 2025. [MMInA: Benchmarking multihop multimodal Internet agents](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13682–13697, Vienna, Austria. Association for Computational Linguistics.
- E Paul Torrance. 1966. Torrance tests of creative thinking. *Educational and psychological measurement*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. [Chain-of-thought prompting elicits reasoning in large language models](#). *Preprint*, arXiv:2201.11903.
- Anita Williams Woolley, Christopher F. Chabris, Alex

- Pentland, Nada Hashmi, and Thomas W. Malone. 2015. [Collective intelligence and group performance](#). *Current Directions in Psychological Science*, 24(6):420–424.
- Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024a. [Mind’s eye of LLMs: Visualization-of-thought elicits spatial reasoning in large language models](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Zhiyong Wu, Chengcheng Han, Zichen Ding, Zhenmin Weng, Zhoumianze Liu, Shunyu Yao, Tao Yu, and Lingpeng Kong. 2024b. [Os-copilot: Towards generalist computer agents with self-improvement](#). *Preprint*, arXiv:2402.07456.
- Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. [Travelplanner: A benchmark for real-world planning with language agents](#). In *Forty-first International Conference on Machine Learning*.
- Yutaro Yamada, Yihan Bao, Andrew Kyle Lampinen, Jungo Kasai, and Ilker Yildirim. 2024. [Evaluating spatial understanding of large language models](#). *Transactions on Machine Learning Research*.
- Ruihan Yang, Jiangjie Chen, Yikai Zhang, Siyu Yuan, Aili Chen, Kyle Richardson, Yanghua Xiao, and Deqing Yang. 2024a. [Selfgoal: Your language agents already know how to achieve high-level goals](#). *Preprint*, arXiv:2406.04784.
- Tao Yang, Yingmin Luo, Zhongang Qi, Yang Wu, Ying Shan, and Chang Wen Chen. 2024b. [Posterllava: Constructing a unified multi-modal layout generator with llm](#). *Preprint*, arXiv:2406.02884.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. [React: Synergizing reasoning and acting in language models](#). *arXiv preprint arXiv:2210.03629*.
- Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. 2024. [Assistantbench: Can web agents solve realistic and time-consuming tasks?](#) *Preprint*, arXiv:2407.15711.
- Lifan Yuan, Yangyi Chen, Xingyao Wang, Yi R. Fung, Hao Peng, and Heng Ji. 2024. [Craft: Customizing llms by creating and retrieving from specialized toolsets](#). *Preprint*, arXiv:2309.17428.
- Lin-Ping Yuan, Ziqi Zhou, Jian Zhao, Yiqiu Guo, Fan Du, and Huamin Qu. 2021. [Infocolorizer: Interactive recommendation of color palettes for infographics](#). *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4252–4266.
- Cong Zhang, Xin Deik Goh, Dexun Li, Hao Zhang, and Yong Liu. 2025. [Planning with multi-constraints via collaborative language agents](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10054–10082, Abu Dhabi, UAE. Association for Computational Linguistics.
- Hengyuan Zhao, Pan Zhou, Difei Gao, Zechen Bai, and Mike Zheng Shou. 2024. [LOVA3: Learning to visual question answering, asking and assessment](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Nanxuan Zhao, Ying Cao, and Rynson WH Lau. 2018. [Modeling fonts in context: Font prediction on web designs](#). In *Computer Graphics Forum*, volume 37, pages 385–395. Wiley Online Library.
- Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. [Gpt-4v\(ision\) is a generalist web agent, if grounded](#). In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org.
- Sipeng Zheng, Jiazheng Liu, Yicheng Feng, and Zongqing Lu. 2023. [Steve-eye: Equipping llm-based embodied agents with visual perception in open worlds](#). *arXiv preprint arXiv:2310.13255*.
- Xinyue Zheng, Haowei Lin, Kaichen He, Zihao Wang, Zilong Zheng, and Yitao Liang. 2025. [Mcu: An evaluation framework for open-ended game agents](#). *Preprint*, arXiv:2310.08367.
- Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, Yu Qiao, Zhaoxiang Zhang, and Jifeng Dai. 2023. [Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory](#). *Preprint*, arXiv:2305.17144.

## A HuggingFace Models

HuggingFace model names for open-weight models are listed in Table 5.

## B GraphicWeaver Construction Pipeline

### B.1 Environment Setting

The complete list of 46 available tools in GraphicWeaver is provided in Table 6. All tools are derived based on Adobe’s official tutorial videos, which provides a diverse range of commonly used operations by graphic design practitioners. In Figure 4, we further show an example of one executable JavaScript code for AdjustBC tool for the Photo Editor agent.

### B.2 Human Annotation

We present examples of human-annotated user queries for each design category in Table 7. Annotators are instructed to use the same design environment(s) as those employed in the original reference designs.

### B.3 Quality Control

We detail the human annotation process as part of constructing GraphicWeaver. We built our custom annotation interface as illustrated in Figure 5. We invited 8 students to participate and provide a compensation of 10 USD gift card each. Before the survey, we show examples of both successful and failed cases to provide some context of annotation standards to annotators.

As part of the pre-survey, annotators were asked two questions on a five-point Likert scale: **(1) Design tool usage:** How often do you use design tools in daily work and life? (1:Never, 5:Always) and **(2) Adobe software usage:** How familiar are you in using Adobe software (e.g., Photoshop, Illustrator)? (1:Not familiar at all, 5: Extremely familiar). Of the 8 annotators, for design tool usage, 3 responded “Never” (Never in the past month), 4 “Rarely” (Fewer than once a week), and 1 “Sometimes” (two or three times a week). For Adobe software usage, 3 were “Not familiar at all” (have never used it before), 2 “Slightly familiar” (have some basic knowledge but have rarely used it), and 3 “Moderately familiar” (can perform simple tasks but may need guidance for more complex features).

### B.4 VQA Examples

We present several examples of generated questions for measuring VQA pass rate as part of the

Model	HuggingFace Name
QWEN-2.5-VL 7B	Qwen/Qwen2.5-VL-7B-Instruct
QWEN-2.5-VL 32B	Qwen/Qwen2.5-VL-32B-Instruct
GEMMA-3 12B	google/gemma-3-12b-it
GEMMA-3 27B	google/gemma-3-27b-it

Table 5: HuggingFace model names for the tested open-weight models.

execution evaluation are detailed in Table 8.

## C Detailed Results

### C.1 Planning & Execution Evaluation

We provide the full numerical results by VLM and design category for all planning and execution metrics in Table 9 (direct) and Table 10 (agentic).

### C.2 Tool-Use Analysis

We present the top-3 most common tool-use sequences for each VLM in Table 11. We find that many of the sequences closely follow human-annotated workflows, typically starting with document creation (`CreateDocument` or `CreateDocumentCustom`), setting the background color (`SetBackgroundColor`), importing images as objects (`ImportObject`), manipulating the imported object (such as using `ResizeObject`, `RepositionObject`, etc.), saving the document (`SaveDocument`), and manipulating text elements (`CreateText`, `ApplyFont`, `ColorText`, etc.).

Category	Tool	Input Parameters	Description	Env.
Basic	CreateDocument	docType	Create new document with pre-defined dimensions.	PS AI Id
	CreateDocumentCustom	width, height	Create new document with desired width and height values.	PS AI Id
	SetBackgroundColor	red, green, blue	Set the background color to desired RGB color.	PS AI Id
	SaveDocument	fileName, format	Save the current document into desired format.	PS AI Id
Drawing	DrawCircle	layerName, radius, red, green, blue	Draw a circle of desired radius and RGB color.	AI
	DrawEllipse	layerName, majorRadius, minorRadius, red, green, blue	Draw an ellipse of desired radius and RGB color.	AI
	DrawLine	layerName, startX, startY, endX, endY, strokeWidth, red, green, blue	Draw a line of desired length, stroke, and RGB color.	AI
	DrawPolygon	layerName, sides, radius, red, green, blue	Draw a polygon of desired number of sides, radius, and RGB color.	AI
	DrawRectangle	layerName, width, height, red, green, blue	Draw a rectangle of desired size and RGB color.	AI
	DrawStar	layerName, numPoints, radius, red, green, blue	Draw a star of desired number of points, radius, and RGB color.	AI
	DrawTriangle	layerName, base, height, red, green, blue	Draw a triangle of desired size and RGB color.	AI
	OpacityDrawing	layerName, opacity	Adjust opacity of a drawing.	AI
	RemoveDrawing	layerName	Remove a drawing.	AI
	RepositionDrawing	layerName, posX, posY	Reposition a drawing to desired x and y-axis position.	AI
	ResizeDrawing	layerName, width, height	Resize a drawing to desired width and height.	AI
	RotateDrawing	layerName, angle	Rotate a drawing to desired angle.	AI
StrokeDrawing	layerName, strokeWidth, red, green, blue	Adjust stroke of a drawing with desired width and RGB color.	AI	
Text	AlignText	layerName, alignment	Align text to desired alignment (left, center, right).	PS AI Id
	ApplyFont	layerName, fontName	Apply font to text.	PS AI Id
	ArrangeText	layerName, arrangement	Arrange text to desired arrangement (front, forward, back, backward).	PS AI Id
	ColorText	layerName, red, green, blue	Color text to desired RGB color.	PS AI Id
	CreateText	layerName, textString	Create a new text (default to Arial font).	PS AI Id
	OpacityText	layerName, opacity	Adjust opacity of text.	PS AI Id
	RemoveText	layerName	Remove text.	PS AI Id
	RepositionText	layerName, posX, posY	Reposition text to desired x and y-axis position.	PS AI Id
	ResizeText	layerName, fontSize	Resize text to desired font size.	PS AI Id
	RotateText	layerName, angle	Rotate text to desired angle.	PS AI Id
StrokeText	layerName, strokeWidth, red, green, blue	Adjust stroke of text with desired width and RGB color.	AI Id	
Object	ImportObject	fileName, layerName	Import an image or object from file path.	PS AI Id
	OpacityObject	fileName, opacity	Adjust opacity of an object.	PS AI
	RemoveObject	fileName	Remove an object.	PS AI Id
	RepositionObject	fileName, posX, posY	Reposition an object to desired x and y-axis position.	PS AI Id
	ResizeObject	fileName, width, height	Resize an object to desired width and height.	PS AI Id
	RotateObject	fileName, angle	Rotate an object to desired angle.	PS AI Id
	GenerateQRObject	layerName, linkURL	Generate a QR code with desired URL embedded.	Id
	AdjustBC	layerName, brightness, contrast	Adjust brightness and contrast level of an object.	PS
	AdjustBW	layerName	Change an object to black & white.	PS
	AdjustHSL	layerName	Adjust hue, saturation, and lightness level of an object.	PS
	BlurObject	layerName, blurAmount	Blur an object to desired amount.	PS
	PhotoFilter	layerName, filterType, density	Apply a photo filter to an object with desired density.	PS
	GlassFilter	layerName, distortion, smoothness, scaling	Apply a glass filter to an object with the specified parameters.	PS
	GlowFilter	layerName, graininess, glowAmount, clearAmount	Apply a glow filter to an object with the specified parameters.	PS
	OceanRippleFilter	layerName, rippleSize, rippleMagnitude	Apply an ocean ripple filter to an object with the specified parameters.	PS
	StainedGlassFilter	layerName, cellSize, borderThickness, lightIntensity	Apply a stained glass filter to an object with the specified parameters.	PS
PatchWorkFilter	layerName, squareSize, relief	Apply a patchwork filter to an object with the specified parameters.	PS	
WatercolorFilter	layerName, brushDetail, shadowIntensity, texture	Apply a watercolor filter to an object with the specified parameters.	PS	

**Table 6: Complete list of available tools in GraphicWeaver.** Each tool requires specific parameters for execution. **Experts:** The expert agent(s) which supports the execution of a specific tool. For numerical parameters, we provide reference ranges (e.g., angle as [0, 360], brightness as [-150, +150]). For parameters in filter-related functions, we provide a short description.

#### JavaScript for AdjustBC action (Photo Editor agent)

```
function promptForLayerName() {
    var layerName = arguments[0];

    if (layerName == null || layerName == "") {
        throw new Error("Layer with the name '" + layerName + "' does not exist.");
    }
    return layerName;
}

function promptForAdjustmentValues() {
    var brightness = parseInt(arguments[1], 10);
    var contrast = parseInt(arguments[2], 10);

    if (isNaN(brightness) || isNaN(contrast)) {
        throw new Error("Invalid input provided. Please run the script again and provide valid numbers.");
    }
    return { brightness: brightness, contrast: contrast };
}

function layerExists(layerName) {
    var ref = new ActionReference();
    ref.putName(charIDToTypeID("Lyr "), layerName);
    try {
        var desc = executeActionGet(ref);
        return true;
    } catch (e) {
        return false;
    }
}

function selectLayerByName(layerName) {
    var idselect = charIDToTypeID("slct");
    var desc = new ActionDescriptor();
    var idnull = charIDToTypeID("null");
    var ref = new ActionReference();
    var idLyr = charIDToTypeID("Lyr ");
    ref.putName(idLyr, layerName);
    desc.putReference(idnull, ref);
    var idMkVs = charIDToTypeID("MkVs");
    desc.putBoolean(idMkVs, false);
    executeAction(idselect, desc, DialogModes.NO);
}

function applyBrightnessContrastAdjustment(brightness, contrast) {
    var idBrtC = charIDToTypeID("BrgC");
    var desc = new ActionDescriptor();
    desc.putUnitDouble(charIDToTypeID("Brgh"), charIDToTypeID("#Prc"), brightness);
    desc.putUnitDouble(charIDToTypeID("Cntr"), charIDToTypeID("#Prc"), contrast);
    executeAction(idBrtC, desc, DialogModes.NO);
}

function adjustBrightnessContrast() {
    if (!layerExists(layerName)) {
        throw new Error("Layer with the name '" + layerName + "' does not exist.");
    }

















    var layerName = promptForLayerName();
    if (layerName == null) {
        throw new Error("Layer name does not exist.");
    }

    var adjustments = promptForAdjustmentValues();
    if (adjustments == null) {
        throw new Error("Parameter values are not provided.");
    }





    selectLayerByName(layerName);
    applyBrightnessContrastAdjustment(adjustments.brightness, adjustments.contrast);
}

adjustBrightnessContrast();
```

**Figure 4: JavaScript code snippet for AdjustBC for the Photo Editor agent.** Each tool corresponds to an executable function in the design environment that takes predefined parameter values as input.

Category	Example User Query	Reference	Design	Env.
 Book Cover	Create a book cover design for a romance novel titled ‘Love \n Story’ featuring a silhouette illustration of a couple in a romantic pose against a pink moonlit background. The title should be at the top center, the author’s name ‘A Novel By \n Olivia Wilson’ below the title, and the tagline ‘Best Selling Book of the Year’ above the title, all in white.			
 Business Card	Create a one-sided business card design with a light yellow background for the bookstore ‘CACTUS’. Replace the ‘T’ in ‘CACTUS’ with a cactus-shaped illustration in green font, centered and add a tagline ‘Livros Novos e Usados’ in green font below the bookstore name.			
 Postcard	Create a postcard design with the message ‘Think Happy!’ in a red, curly font on a floral background featuring a mix of warm-toned roses. Place a semi-transparent white box behind the message.			
 Poster	Create a poster design with a light yellow background, featuring a large jellyfish illustration centered within a black rectangular box. Add a bold, black title ‘JELLYFISH’ at the top and place a brief informative sentence about jellyfish in white font at the bottom left corner.			

**Table 7: Examples of human-annotated user queries, reference designs, and resulting design outcomes for each design category. Env.:** Design environment(s) used for planning and execution.

Category	Example User Query	Example Questions
 Book Cover	Please create a self-help book cover design titled ‘Achieve Your Dreams’ with the author’s name ‘Nathan White’, featuring a person climbing a mountain centered in the lower half against a sunrise background. The title should be at the top center in white, the author’s name below the title in white, and the tagline ‘Climb Higher, Dream Bigger.’ below the author’s name in white.	[“Is there a text ‘Achieve Your Dreams?’”, “Is there a text ‘Nathan White?’”, “Is there a person climbing a mountain?”, “Is the background a sunrise?”, “Is the text ‘Achieve Your Dreams’ at the top center in white?”, “Is the text ‘Nathan White’ below the title in white?”, “Is the text ‘Climb Higher, Dream Bigger.’ below the author’s name in white?”, “Is the person climbing a mountain centered in the lower half?”]
 Business Card	I need to create a business card design for ‘Sparkle Jewelry’ with a royal blue background. Please include the company name in large gold font centered, and a medium-sized diamond icon placed above the company name. The contact details of ‘Phone: +1 987 654 3210 \n Email: info@sparklejewelry.com \n Address: 12 Gem St, Los Angeles, CA, USA’ should be in small gold font, placed bottom right.	[“Is the background of the business card royal blue?”, “Is the company name ‘Sparkle Jewelry’ in large gold font?”, “Is the company name centered?”, “Is there a medium-sized diamond icon?”, “Is the diamond icon placed above the company name?”, “Are the contact details in small gold font?”, “Are the contact details placed at the bottom right?”, “Is the phone number ‘+1 987 654 3210’ included in the contact details?”, “Is the email ‘info@sparklejewelry.com’ included in the contact details?”, “Is the address ‘12 Gem St, Los Angeles, CA, USA’ included in the contact details?”]
 Postcard	Please create a motivational postcard design with the message ‘Stay Positive, Work Hard’ at the top in red on a yellow background featuring a large lion roaring at the bottom.	[“Is there a text ‘Stay Positive, Work Hard?’”, “Is the text ‘Stay Positive, Work Hard’ at the top?”, “Is the text ‘Stay Positive, Work Hard’ in red?”, “Is the background yellow?”, “Is there an illustration of a lion?”, “Is the lion roaring?”, “Is the lion illustration large?”, “Is the lion illustration at the bottom?”]
 Poster	Could you help create a promotional poster design for a jazz festival on a deep blue background, featuring a large image of a saxophonist playing in the center, a huge bold title ‘JAZZFEST’ in gold at the top center, and event details ‘Jazz Festival   May 5-7, 2023   Central Park, New York’ in medium golden text at the bottom right?	[“Is there a deep blue background?”, “Is there a large image of a saxophonist playing in the center?”, “Is there a huge bold title ‘JAZZFEST’?”, “Is the title ‘JAZZFEST’ in gold?”, “Is the title ‘JAZZFEST’ at the top center?”, “Is there event details ‘Jazz Festival   May 5-7, 2023   Central Park, New York’?”, “Is the event details in medium golden text?”, “Is the event details at the bottom right?”]

**Table 8: Examples of the generated questions using GPT-4 for each design category. We use the questions to compute the VQA pass rate as part of execution evaluation.**

**User Query:**

We need to create a business card design for a travel agency named 'Adventure Awaits' with a sky blue background. Please include the agency name in huge white font at the top center. I want to include the tagline 'Your Journey Begins Here' in medium white font below the agency name, and a mountain icon of medium size centered below the tagline. Also, add the contact details 'Phone: +1 800-555-7890 Email: info@adventureawaits.com Website: www.adventureawaits.com' in small white font at the bottom left.

**Design Choices:**

- **Background Color:** sky blue
- **Text:**
  - **Agency\_name**
    - Content: "Adventure Awaits"
    - Size: large
    - Color: white
    - Position: top center
  - **Tagline**
    - Content: "Your Journey Begins Here"
    - Size: small
    - Color: white
    - Position: below the agency name
  - **Contact\_details**
    - Content: "Phone: +1 800-555-7890 Email: info@adventureawaits.com Website: www.adventureawaits.com"
    - Size: small
    - Color: white
    - Position: bottom left
- **Images:**
  - **Mountain icon**
    - Size: medium
    - Position: center below the tagline

**[Question 1]**

Is each design choice **aligned** with the user query?

(Here, "**aligned**" means that each design element fits the user's specifications in the query and contributes to the overall coherence of the final design.)

	<b>Not aligned at all</b> (key elements are missing)	<b>Slightly aligned</b> (some important elements misplaced/incorrectly implemented)	<b>Moderately aligned</b> (capture general intent but need adjustment to fully meet user's specifications)	<b>Aligned well</b> (only need minor adjustments)	<b>Completely aligned</b> (perfectly match user query)
Background color	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Agency_name	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tagline	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Contact_details	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**[Question 2]**

If you answered "Not aligned at all" or "Slightly aligned" to the previous question, please explain your reasoning in 1-2 sentences. You may also suggest changes to improve the alignment of the design choices with the user query.

Enter here

**[Question 3]**

Below are the three image candidates for "**mountain icon**". Please rank them based on how well they fit the user query. Assign a rank from 1 (best fit) to 3 (least fit).

**Image 1**



Rank:

**Image 2**



Rank:

**Image 3**



Rank:

**Figure 5: Screenshot of human annotation interface.** For each user query, annotators are asked to (Q1) evaluate how well each design component aligns with the query on a five-point Likert scale and (Q2) rank the three images from 1 to 3 based on their relevance to the query. Additionally, they have the option to provide free-form feedback.

Model	Planning				Execution					
	Delivery Rate (%)	Design Pass Rate (%)	Step Eff.	Expert Use Eff.	Success Rate (%)	Fidelity	Content Similarity	VQA Pass Rate (%)	Creative (O)	Creative (E)
<i>Direct mode (aggregated)</i>										
QWEN-2.5-VL 7B	27.1	22.5	45.5	0.98	10.3	0.01	5.70	16.0	1.05	0.64
QWEN-2.5-VL 32B	<u>57.2</u>	19.7	55.9	<b>1.00</b>	17.5	0.04	6.62	19.0	1.18	1.07
GEMMA-3 12B	37.7	19.0	63.8	0.99	14.9	0.03	6.93	13.6	1.18	0.76
GEMMA-3 27B	<b>67.8</b>	<u>24.8</u>	70.9	<b>1.00</b>	22.3	0.09	<u>9.36</u>	17.7	1.09	1.09
GPT-04-MINI	46.2	23.3	<u>79.5</u>	<b>1.00</b>	<u>27.9</u>	<u>0.10</u>	8.72	<u>21.5</u>	1.14	<u>1.12</u>
GPT-4.1	30.3	<b>30.1</b>	<b>82.5</b>	<b>1.00</b>	<b>29.5</b>	<b>0.13</b>	<b>13.1</b>	<b>23.3</b>	<b>1.39</b>	<b>1.35</b>
<i>Book Cover</i>										
QWEN-2.5-VL 7B	9.34	<u>25.5</u>	38.3	<u>0.98</u>	3.5	0.00	5.72	21.6	1.15	0.64
QWEN-2.5-VL 32B	<u>52.8</u>	21.1	65.1	<b>1.00</b>	16.8	0.00	5.12	<b>24.8</b>	<b>1.44</b>	1.02
GEMMA-3 12B	23.8	20.0	66.5	<b>1.00</b>	14.4	<u>0.09</u>	7.08	14.5	1.18	0.97
GEMMA-3 27B	<b>68.9</b>	<u>25.5</u>	75.7	<b>1.00</b>	25.9	<u>0.09</u>	9.50	19.9	1.06	<b>1.32</b>
GPT-04-MINI	45.6	23.1	<b>81.4</b>	<b>1.00</b>	<u>27.4</u>	<u>0.09</u>	<u>10.4</u>	20.7	1.08	<u>1.09</u>
GPT-4.1	30.5	<b>28.1</b>	<u>80.3</u>	<b>1.00</b>	<b>29.9</b>	<b>0.12</b>	<b>14.2</b>	<u>21.8</u>	<u>1.25</u>	<b>1.32</b>
<i>Business Card</i>										
QWEN-2.5-VL 7B	18.7	<b>27.4</b>	44.5	0.97	10.3	0.00	5.00	15.7	0.98	0.77
QWEN-2.5-VL 32B	30.6	16.1	46.3	<b>1.00</b>	12.5	0.00	4.67	16.2	1.02	0.94
GEMMA-3 12B	20.9	15.6	68.4	<u>0.98</u>	12.9	0.00	5.72	11.8	1.05	0.64
GEMMA-3 27B	<b>45.6</b>	19.3	65.4	<b>1.00</b>	22.8	0.05	7.45	15.6	<u>1.07</u>	<u>0.97</u>
GPT-04-MINI	37.8	21.1	<u>74.5</u>	<b>1.00</b>	24.4	<u>0.10</u>	<u>7.75</u>	19.3	1.04	<u>0.97</u>
GPT-4.1	18.9	<u>25.7</u>	<b>81.4</b>	<b>1.00</b>	<b>26.1</b>	<b>0.12</b>	<b>11.1</b>	<b>21.0</b>	<b>1.43</b>	<b>1.09</b>
<i>Postcard</i>										
QWEN-2.5-VL 7B	45.6	21.9	55.8	<u>0.98</u>	15.3	0.01	6.95	14.5	0.96	0.43
QWEN-2.5-VL 32B	66.7	23.4	55.6	<b>1.00</b>	18.9	0.08	8.01	19.4	1.13	<u>1.23</u>
GEMMA-3 12B	<u>72.4</u>	21.0	61.1	<b>1.00</b>	16.9	0.03	7.33	15.6	<u>1.25</u>	0.77
GEMMA-3 27B	<b>82.1</b>	<u>27.9</u>	70.9	<b>1.00</b>	19.2	<u>0.11</u>	<u>9.58</u>	17.8	1.09	0.93
GPT-04-MINI	40.6	23.8	<u>82.2</u>	<b>1.00</b>	<b>31.0</b>	<u>0.11</u>	8.51	<u>23.4</u>	1.18	1.12
GPT-4.1	20.7	<b>34.3</b>	<b>83.5</b>	<b>1.00</b>	<u>30.9</u>	<b>0.14</b>	<b>13.8</b>	<b>24.6</b>	<b>1.57</b>	<b>1.47</b>
<i>Poster</i>										
QWEN-2.5-VL 7B	34.9	15.2	43.2	<b>1.00</b>	12.0	0.02	5.12	12.2	1.12	0.71
QWEN-2.5-VL 32B	<b>78.9</b>	18.3	56.8	<b>1.00</b>	21.7	0.09	8.66	15.6	1.11	1.08
GEMMA-3 12B	33.6	19.4	59.2	<b>1.00</b>	15.5	0.02	7.58	12.6	<u>1.25</u>	0.67
GEMMA-3 27B	<u>74.5</u>	<u>26.3</u>	71.4	<b>1.00</b>	21.3	0.11	<u>10.9</u>	17.4	1.15	1.16
GPT-04-MINI	60.7	25.3	80.0	<b>1.00</b>	29.0	0.12	8.21	<u>22.6</u>	1.24	<u>1.32</u>
GPT-4.1	51.1	<b>32.3</b>	<b>84.7</b>	<b>1.00</b>	<b>31.2</b>	<b>0.13</b>	<b>13.3</b>	<b>25.7</b>	<b>1.32</b>	<b>1.54</b>

Table 9: Planning and execution results of different VLMs on GraphicWeaver for **direct** mode (aggregated and per category). For each section, column-wise best scores are **bolded** and second-best scores are underlined. All metrics are interpreted as higher values for better performance (↑).

Model	Planning				Execution					
	Delivery Rate (%)	Design Pass Rate (%)	Step Eff.	Expert Use Eff.	Success Rate (%)	Fidelity	Content Similarity	VQA Pass Rate (%)	Creative (O)	Creative (E)
<i>Agentic mode (aggregated)</i>										
QWEN-2.5-VL 7B	15.2	51.8	92.0	<b>1.00</b>	39.4	0.14	22.5	37.6	1.77	1.59
QWEN-2.5-VL 32B	<u>39.6</u>	49.7	92.4	<b>1.00</b>	<u>61.3</u>	<u>0.20</u>	22.3	35.4	1.89	1.65
GEMMA-3 12B	27.3	<u>54.5</u>	96.5	<b>1.00</b>	58.1	0.17	21.0	36.2	<u>2.01</u>	1.68
GEMMA-3 27B	<b>51.4</b>	54.4	93.1	<b>1.00</b>	55.4	<u>0.20</u>	26.7	<u>44.7</u>	1.98	<u>2.04</u>
GPT-O4-MINI	20.8	51.1	<b>97.1</b>	<b>1.00</b>	56.5	<u>0.20</u>	<u>28.3</u>	<b>44.8</b>	1.78	1.99
GPT-4.1	13.9	<b>56.8</b>	94.7	<b>1.00</b>	<b>62.5</b>	<b>0.21</b>	<b>29.3</b>	44.7	<b>2.02</b>	<b>2.06</b>
<i>Book Cover</i>										
QWEN-2.5-VL 7B	6.40	52.4	92.6	<b>1.00</b>	10.6	0.10	17.2	37.9	1.75	1.57
QWEN-2.5-VL 32B	<u>40.0</u>	<u>56.1</u>	95.4	<b>1.00</b>	<u>66.0</u>	0.18	17.4	34.1	1.99	1.73
GEMMA-3 12B	17.5	55.9	<b>97.2</b>	<b>1.00</b>	60.8	0.17	18.3	33.0	<b>2.22</b>	1.99
GEMMA-3 27B	<b>57.4</b>	53.6	90.7	<b>1.00</b>	53.5	<u>0.19</u>	23.6	<b>46.4</b>	<u>2.07</u>	<b>2.14</b>
GPT-O4-MINI	23.9	48.9	<u>97.0</u>	<b>1.00</b>	55.0	<u>0.18</u>	<b>27.3</b>	<u>41.0</u>	1.91	1.96
GPT-4.1	4.90	<b>56.9</b>	95.4	<b>1.00</b>	<b>66.4</b>	<b>0.22</b>	<u>24.0</u>	38.3	2.00	<u>2.07</u>
<i>Business Card</i>										
QWEN-2.5-VL 7B	<u>12.3</u>	48.8	94.3	<b>1.00</b>	56.0	0.12	19.4	30.7	1.61	1.53
QWEN-2.5-VL 32B	<b>46.8</b>	44.2	90.5	<b>1.00</b>	<b>62.1</b>	<b>0.20</b>	18.5	29.8	1.73	1.64
GEMMA-3 12B	1.70	53.3	<u>96.5</u>	<b>1.00</b>	58.1	0.15	19.2	34.5	1.81	1.64
GEMMA-3 27B	9.60	<u>53.4</u>	94.4	<b>1.00</b>	54.6	0.16	24.8	39.2	<u>1.91</u>	<u>1.98</u>
GPT-O4-MINI	2.60	51.2	<b>97.0</b>	<b>1.00</b>	56.1	<u>0.19</u>	<b>32.1</b>	<b>47.7</b>	1.85	1.79
GPT-4.1	0.00	<b>55.8</b>	93.8	<b>1.00</b>	<u>60.7</u>	<b>0.20</b>	<u>27.8</u>	<u>46.9</u>	<b>2.12</b>	<b>2.03</b>
<i>Postcard</i>										
QWEN-2.5-VL 7B	21.2	53.6	91.0	<b>1.00</b>	48.9	0.18	30.9	<u>48.7</u>	1.98	1.91
QWEN-2.5-VL 32B	61.1	43.1	90.9	<b>1.00</b>	<b>62.8</b>	<u>0.21</u>	31.0	46.9	2.06	1.75
GEMMA-3 12B	68.0	<u>55.2</u>	94.4	<b>1.00</b>	56.4	0.18	23.3	40.1	2.13	1.73
GEMMA-3 27B	<b>82.3</b>	54.8	<u>95.4</u>	<b>1.00</b>	59.5	<b>0.22</b>	<u>31.2</u>	<b>50.2</b>	<u>2.14</u>	<u>2.20</u>
GPT-O4-MINI	15.4	50.3	<b>96.8</b>	<b>1.00</b>	53.6	0.20	28.3	45.0	1.70	2.18
GPT-4.1	8.00	<b>56.5</b>	94.2	<b>1.00</b>	57.5	<u>0.21</u>	<b>33.6</b>	48.4	<b>2.20</b>	<b>2.22</b>
<i>Poster</i>										
QWEN-2.5-VL 7B	20.9	52.4	90.0	<b>1.00</b>	42.2	0.18	22.5	33.3	1.75	1.37
QWEN-2.5-VL 32B	10.6	55.4	92.9	<b>1.00</b>	54.3	0.19	22.1	30.9	1.77	1.47
GEMMA-3 12B	21.9	53.6	<b>97.9</b>	<b>1.00</b>	57.2	0.18	23.2	37.3	<b>1.88</b>	1.37
GEMMA-3 27B	<b>56.1</b>	<u>55.6</u>	91.7	<b>1.00</b>	53.9	<u>0.21</u>	<u>27.0</u>	43.1	<u>1.79</u>	1.85
GPT-O4-MINI	41.4	53.9	<u>97.5</u>	<b>1.00</b>	61.3	<u>0.21</u>	25.5	<b>45.6</b>	1.67	<b>2.04</b>
GPT-4.1	<u>42.7</u>	<b>57.9</b>	95.4	<b>1.00</b>	<b>65.3</b>	<b>0.22</b>	<b>31.8</b>	<u>45.3</u>	1.75	<u>1.91</u>

Table 10: Planning and execution results of different VLMs on GraphicWeaver for **agentic** mode (aggregated and per category). For each section, column-wise best scores are **bolded** and second-best scores are underlined. All metrics are interpreted as higher values for better performance (↑).

Model	Frequency	Tool-Use Sequence
QWEN-2.5-VL 7B	34	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ApplyFont → ColorText → AlignText → ResizeText → RepositionText → ExportDocument → AdjustHSL → SaveDocument
	22	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → AdjustHSL → SaveDocument → CreateText → ApplyFont → ColorText → AlignText → ResizeText → ExportDocument
	17	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → AdjustHSL → SaveDocument → CreateText → ResizeText → ColorText → AlignText → RepositionText → ExportDocument
QWEN-2.5-VL 32B	45	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → SaveDocument → CreateText → ApplyFont → ColorText → AlignText → ExportDocument
	22	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → SaveDocument → CreateText → ApplyFont → ColorText → AlignText → RepositionText → ExportDocument
	12	CreateDocumentCustom → SetBackgroundColor → CreateText → ColorText → AlignText → ImportObject → ResizeObject → RepositionObject → ExportDocument
GEMMA-3 12B	28	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → SaveDocument → CreateText → ResizeText → ColorText → AlignText → RepositionText → ExportDocument
	26	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → SaveDocument → CreateText → ResizeText → ColorText → AlignText → ExportDocument
	14	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ApplyFont → ColorText → AlignText → ResizeText → ExportDocument → AdjustHSL → SaveDocument
GEMMA-3 27B	59	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ColorText → AlignText → ExportDocument
	48	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ResizeText → ColorText → AlignText → ExportDocument
	44	CreateDocumentCustom → ImportObject → ResizeObject → RepositionObject → SaveDocument → SetBackgroundColor → CreateText → ApplyFont → ColorText → AlignText → ExportDocument
GPT-04-MINI	78	CreateDocument → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → AdjustHSL → SaveDocument → CreateText → ColorText → AlignText → ExportDocument
	53	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → AdjustHSL → SaveDocument → CreateDocument → CreateText → ColorText → AlignText → ExportDocument
	42	CreateDocumentCustom → SetBackgroundColor → CreateText → ColorText → AlignText → ImportObject → ResizeObject → RepositionObject → ExportDocument
GPT-4.1	37	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ColorText → AlignText → ExportDocument
	33	CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ResizeText → ColorText → AlignText → ExportDocument
	21	CreateDocument → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → AdjustHSL → SaveDocument → CreateText → AlignText → ExportDocument

Table 11: Top-3 most common tool-use sequences per model, ordered by frequency.

## D Prompts

### D.1 GraphicWeaver Prompts

We show prompt templates used for constructing user queries in GraphicWeaver and prompting VLMs for in the `agentic` mode below.

#### Prompt D.1.1: Query Construction (Book Cover)

**Task:** You are a design expert. Given the template for generating a book cover, fill in the placeholders with appropriate design components. Generate 5 diverse examples in a Python list of strings. Be as creative as possible.

**Template:** Create a book cover design with a [background color] background, featuring [images]. The title [title] should be placed at [position] in [color] and the author name [author name] at [position] in [color]. You may also include an optional [subtitle] or [tagline] if needed.

\*\*\*\* Example Starts \*\*\*

Create a book cover design for a romance novel titled 'Love \n Story' featuring a silhouette illustration of a couple in a romantic pose against a pink moonlit background. The title should be at the top center, the author's name 'A Novel By \n Olivia Wilson' below the title, and the tagline 'Best Selling Book of the Year' above the title, all in white.

\*\*\*\* Example Ends \*\*\*

**Examples:**

#### Prompt D.1.2: Query Construction (Business Card)

**Task:** You are a design expert. Given the template for generating a business card, fill in the placeholders with appropriate design components. Generate 5 diverse examples in a Python list of strings. Be as creative as possible.

**Template:** Create a [side]-sided business card design with a [background color] background, featuring [images]. The name [brand name] should be placed at [position] in [color]. You may also include an optional [contact details] or [tagline] if needed.

\*\*\*\* Example Starts \*\*\*

Create a one-sided business card design with a light yellow background for the bookstore 'CACTUS'. Replace the 'T' in 'CACTUS' with a cactus-shaped illustration in green font, centered and add a tagline 'Livros Novos e Usados' in green font below the bookstore name.

\*\*\*\* Example Ends \*\*\*

**Examples:**

#### Prompt D.1.3: Query Construction (Postcard)

**Task:** You are a design expert. Given the template for generating a postcard, fill in the placeholders with appropriate design components. Generate 5 diverse examples in a Python list of strings. Be as creative as possible.

**Template:** Create a postcard design with a [background color] background, featuring [images]. The message [message] should be placed at [position] in [color].

\*\*\*\* Example Starts \*\*\*

Create a postcard design with the message 'Think Happy!' in a red, curly font on a floral background featuring a mix of warm-toned roses. Place a semi-transparent white box behind the message.

\*\*\*\* Example Ends \*\*\*

**Examples:**

Prompt D.1.4: Query Construction (Poster)

**Task:** You are a design expert. Given the template for generating a poster, fill in the placeholders with appropriate design components. Generate 5 diverse examples in a Python list of strings. Be as creative as possible.

**Template:** Create a poster design with a [background color] background, featuring [images]. The title [title] should be placed at [position] in [color]. You may also include an optional [tagline] if needed.

\*\*\*\* Example Starts \*\*\*

Create a poster design with a light yellow background, featuring a large jellyfish illustration centered within a black rectangular box. Add a bold, black title 'JELLYFISH' at the top and place a brief informative sentence about jellyfish in white font at the bottom left corner.

\*\*\*\* Example Ends \*\*\*

**Examples:**

### Prompt D.1.5: Expert Agent Recruitment

**Task:** Your task is to recruit the necessary experts to complete a design outlined in the user query. Create a recruitment status in JSON list format.

**User query:** {user query}

**User image file(s):** {user image files}

You can recruit from the three experts with the following profiles:

- **Photo Editor**
  - Job Responsibilities:
    - \* Image editing: Cropping, adjusting composition, correcting lighting, and retouching images or illustrations.
    - \* Color correction: Adjusting brightness and contrast or adjusting hue and saturation.
    - \* Apply filters: Apply different filters (e.g., photo, glass, ocean ripple, watercolor) to images.
- **Vector Graphic Editor**
  - Job Responsibilities:
    - \* Draw shapes: Drawing simple shapes (circle, polygon, square, star) on canvas.
- **Layout Designer**
  - Job Responsibilities:
    - \* Customize layout templates: Create grid systems for books, brochures, cards, and magazines to organize the layout.
    - \* Export files: Export documents to any format, in print or digital.
    - \* Combine text and visual elements: Combine visual elements from other apps with text into a completed design.

\*\*\* Output Format \*\*\*

Each object in the JSON list should follow:

```
{
  "expert": "Name of the expert (Photo Editor, Vector Graphic Editor, Layout
    Desinger).",
  "task": "High-level task that can be performed by the expert."
}
```

\*\*\* Example Starts \*\*\*

```
[
  {"expert": "Photo Editor", "task": "Add the provided images to create a deep
    purple night sky background with a large dreamy moon centered,
    surrounded by small twinkling stars spread across the top half of the
    cover."},
  {"expert": "Layout Designer", "task": "Combine the edited image with the
    title 'Moonlit Fantasies', the author name 'J.K. Stellar', and the
    tagline 'A Journey Through the Night Sky.' to create the book cover
    design."}
]
```

\*\*\* Example Ends \*\*\*

\*\*\*\* Key Requirements \*\*\*\*

- Only recruit each expert one.
- The name of the expert must match those in the expert profiles.
- For task description, explain how the expert can contribute towards the final product. Summarize in one sentence.
- In order to achieve the task in the design outline, experts should work together and their task will be dependent to each other. Arrange in the order of which expert should finish first.
- Output should be in a list of JSON objects format.
- Do NOT include further explanation other than in the JSON list.
- Be as concise and brief as possible.

**Recruitment status:**

### Prompt D.1.6: Individual Plan Generation

**Task:** You are a proficient {expert}. You are recruited to collaborate on a design project with other experts.

You are assigned to complete the following task: {task}. Please plan a sequence of detailed, low-level sub-tasks required to accomplish this task and output them as a JSON list.

\*\*\*\* Output Format \*\*\*\*

Each object in the JSON list should follow:

```
{
  "id": "ID of the subtask, starting from 1.",
  "expert": "Name of the expert.",
  "description": "Description of the subtask in one sentence."
}
```

\*\*\* Example Starts \*\*\*

```
[
  {"id": 1, "expert": "Photo Editor", "description": "Create a new document with book cover dimensions."},
  {"id": 2, "expert": "Photo Editor", "description": "Set the background color to light pink."},
  {"id": 3, "expert": "Photo Editor", "description": "Import the pink moonlit image from 'static/pink_moonlit.png'."},
  {"id": 4, "expert": "Photo Editor", "description": "Resize the pink moonlit image to medium size, covering the bottom part of the document."},
  {"id": 5, "expert": "Photo Editor", "description": "Reposition the pink moonlit image to the bottom-center of the document."},
  {"id": 6, "expert": "Photo Editor", "description": "Import the couple silhouette illustration from 'static/couple_silhouette.png'."},
  {"id": 7, "expert": "Photo Editor", "description": "Resize the couple silhouette illustration to span across the lower half of the cover."},
  {"id": 8, "expert": "Photo Editor", "description": "Reposition the couple silhouette illustration to be centered in the bottom-middle part."},
  {"id": 9, "expert": "Photo Editor", "description": "Adjust the background colors to match the light pink moonlit theme."},
  {"id": 10, "expert": "Photo Editor", "description": "Save the document in a psd format suitable for further editing by the Layout Designer."}
]
```

\*\*\* Example Ends \*\*\*

\*\*\*\* Key Requirements \*\*\*\*

- First step should always be creating a new document and the last step should always be saving the document in appropriate file format.
- Use the exact image URLs the user provided when importing images.
- Do not include very basic operations such as opening the software or closing the software.
- Do not include new expert in the plan.
- Output should be in a list of JSON objects format.
- Do NOT include further explanation other than in the JSON list.
- Be as concise and brief as possible.

**Sequence of subtasks:**

### Prompt D.1.7: Plan Supervision

**Task:** You are the supervisor of a design project that requires collaboration among various design experts.

The following experts have been recruited for the project. Use as reference:

```
{recruitment status}
```

Each expert has submitted their proposed workflow plans:

```
{workflow plans}
```

Your task is to combine these proposed workflow plans into a cohesive sequence of tasks in a JSON list format.

\*\*\*\* Output Format \*\*\*\*

Each object in the JSON list should follow:

```
{
  "id": "ID of the subtask, starting from 1.",
  "expert": "Name of the expert.",
  "description": "Description of the subtask in one sentence."
}
```

\*\*\* Example Starts \*\*\*

```
[
  {"id": 1, "expert": "Photo Editor", "description": "Create a new document
  with book cover dimensions."},
  {"id": 2, "expert": "Photo Editor", "description": "Set the background color
  to light pink."},
  ...
  {"id": 11, "expert": "Layout Designer", "description": "Create a new
  document with book cover dimensions."},
  {"id": 12, "expert": "Layout Designer", "description": "Import the edited
  image from the Photo Editor: 'moonlit_illustration_edited.psd'."},
  {"id": 13, "expert": "Layout Designer", "description": "Resize the edited
  image to cover the entire document."},
  {"id": 14, "expert": "Layout Designer", "description": "Create text for the
  title 'LOVE\nSTORY'."},
  {"id": 15, "expert": "Layout Designer", "description": "Apply the Andale
  Mono font to the title text."},
  ...
  {"id": 31, "expert": "Layout Designer", "description": "Reposition the
  tagline text above the title."},
  {"id": 32, "expert": "Layout Designer", "description": "Export the final
  book cover design as a PDF file."}
]
```

\*\*\* Example Ends \*\*\*

\*\*\*\* Key Requirements \*\*\*\*

- Do NOT repeat any steps that are already completed in previous step.
- For each expert, first step should always be creating a new document and the last step should always be saving the document in appropriate file format.
- When switching experts, use the output from the previous expert as input for the next.
- Once an expert is used and switched to another expert, it should not be used again.
- You should output only one list of workflow plan.
- Start the id from 1 to the number of steps in the workflow.
- Arrange each subtask in a chronological order.
- Output should be in a list of JSON objects format.
- Do NOT include further explanation other than in the JSON list.
- Be as concise and brief as possible.

**Supervised sequence of subtasks:**

### Prompt D.1.8: Tool Retrieval

**Task:** You are a proficient {expert}. You are recruited to collaborate on a design project with other experts. Use your available list of tools to map each step in the sequence of subtasks to a tool.

**Sequence of subtasks:** {workflow plan}

Your available tools are as below:

{list of tools}

\*\*\*\* Output Format \*\*\*\*

Each object in the JSON list should follow:

```
{
  "id": "ID of the subtask, starting from 1.",
  "expert": "Name of the expert.",
  "description": "Description of the subtask in one sentence.",
  "tool": "Name of the mapped tool.",
  "parameters": "Dictionary of parameter keys and corresponding values."
}
```

\*\*\* Example Starts \*\*\*

```
[
  {"id": 1, "expert": "Photo Editor", "description": "Create a new document
  with book cover dimensions.", "skill": "CreateDocument", "parameters":
  {"docType": "book cover"}},
  {"id": 2, "expert": "Photo Editor", "description": "Set the background color
  to light pink.", "skill": "SetBackgroundColor", "parameters": {"red":
  255, "green": 179, "blue": 238}},
  ...
  {"id": 8, "expert": "Photo Editor", "description": "Reposition the couple
  silhouette illustration to be centered in the bottom-middle part.", "
  parameters": {"layerName": "SilhouetteLayer", "posX": 267, "posY":
  1052}},
  {"id": 9, "expert": "Photo Editor", "description": "Adjust the background
  colors to match the light pink moonlit theme.", "skill": "AdjustHSL", "
  parameters": {"layerName": "MoonlitLayer", "hue": 18, "saturation": -18,
  "light": 0}},
  {"id": 10, "expert": "Photo Editor", "description": "Save the document in a
  format suitable for further editing by the Layout Designer.", "skill": "
  SaveDocument", "parameters": {"fileName": "moonlit_illustration_edited",
  "format": "psd"}},
]
```

\*\*\* Example Ends \*\*\*

\*\*\*\* Key Requirements \*\*\*\*

- For any file name that appears in the design outline, use exact file names in your sequence of subtasks.
- Each step should only be mapped to one tool. If a step of the workflow is not able to be mapped to one tool, it means the step can be decomposed further into multiple steps. You can reformat, reorder, add, edit steps of the workflow if needed to be directly mapped to tools.
- Each step should have a tool and a dictionary of parameter values.
- For layerName, try to name it as to end as Layer (e.g., BackgroundLayer, TitleLayer).
- For detailed numeric values (e.g., height, width, x-axis position, y-axis position), consider the document's dimensions, imagine, and propose a likely value.
- Arrange each subtask in a chronological order.
- Output should be in a list of JSON objects format.
- Do NOT include further explanation other than in the JSON list.
- Be as concise and brief as possible.

**Sequence of subtasks:**

## D.2 Evaluation Prompts

We present the prompts used for evaluating both planning and execution, including design pass rate (color, text, image), VQA pass rate, and creativity (originality and elaboration).

Prompt D.2.1: Design Pass Rate Evaluation (Color)

**Task:** Evaluate if the workflow plan (1) correctly applies the background color and (2) the background and the text color are contrasting. Return a score between 1 to 5 according to the scoring rubric.

**Background color:** {background color}

**Text elements:** {text}

**Workflow plan:** {workflow plan}

\*\*\*\* Scoring Rubric \*\*\*

- 1: Workflow plan fails to reflect all of the color constraints specified.
- 3: Workflow plan reflects approximately half of the color constraints specified.
- 5: Workflow plan reflects all of the color constraints specified.

Score should strictly be a number between 1 to 5. Do not include any further explanation other than the score.

**Score:**

Prompt D.2.2: Design Pass Rate Evaluation (Text)

**Task:** Evaluate if the workflow plan adequately applies the text elements (e.g., title, tagline) specified. Return a score between 1 to 5 according to the scoring rubric.

**Text elements:** {text}

**Workflow plan:** {workflow plan}

\*\*\*\* Scoring Rubric \*\*\*

- 1: Workflow plan fails to reflect all of the text elements specified.
- 3: Workflow plan reflects approximately half of the text elements specified.
- 5: Workflow plan reflects all of the text elements specified.

Score should strictly be a number between 1 to 5. Do not include any further explanation other than the score.

**Score:**

Prompt D.2.3: Design Pass Rate Evaluation (Image)

**Task:** Evaluate if the workflow plan adequately applies the image elements (e.g., size, position) specified. Return a score between 1 to 5 according to the scoring rubric.

**Image elements:** {image}

**Workflow plan:** {workflow plan}

\*\*\*\* Scoring Rubric \*\*\*

- 1: Workflow plan fails to reflect all of the image elements specified.
- 3: Workflow plan reflects approximately half of the image elements specified.
- 5: Workflow plan reflects all of the image elements specified.

Score should strictly be a number between 1 to 5. Do not include any further explanation other than the score.

**Score:**

Prompt D.2.4: VQA Evaluation

**Instruction:** Look at the image and answer the question with 'Yes' or 'No'.

**Question:** {question}

**Answer:**

Prompt D.2.5: Creativity Evaluation (Originality)

**Instruction:** Evaluate the originality of the image generated based on the user query. Originality measures the uniqueness of the ideas generated. Original ideas are those that are rare or unconventional, differing from the norm. Return a score between 1 to 5 according to the scoring rubric.

**User query:** {user query}

\*\*\*\* Scoring Rubric \*\*\*\*

- 1: Image is highly conventional and predictable. No significant signs of creative thinking is shown.
- 2: Image shows minimal originality and mostly align with typical or common responses. Few novel elements are present.
- 3: Image is somewhat original, with a mix of conventional and unique elements.
- 4: Image is noticeable original and uncommon. It shows creative thinking and depart meaningfully from conventional norms.
- 5: Image is highly unique, rare, and stand out as unconventional. They demonstrate a strong departure from typical or expected approaches.

Score should strictly be a number between 1 to 5. Do not include any further explanation other than the score.

**Score:**

Prompt D.2.6: Creativity Evaluation (Elaboration)

**Instruction:** Evaluate the elaboration of the image generated based on the user query. Elaboration refers to the ability to expand upon, refine, and embellish an idea. It involves adding details, developing nuances, and building upon a basic concept to make it more intricate or complex. Return a score between 1 to 5 according to the scoring rubric.

**User query:** {user query}

\*\*\*\* Scoring Rubric \*\*\*\*

- 1: Image is presented in a simpler or vague manner with no meaningful development or supporting detail.
- 2: Image is minimally expanded, with few details or refinements added.
- 3: Image includes expansion of some details, but elaboration is somewhat surface-level.
- 4: Image well-expands the user query with several added details and refinements.
- 5: Image thoroughly expands the user query with rich, specific details or refinements added beyond the core concept.

Score should strictly be a number between 1 to 5. Do not include any further explanation other than the score.

**Score:**

# Scaling Vision–Language Models for Pharmaceutical Long-Form Video Reasoning on Industrial GenAI Platform

Suyash Mishra<sup>a</sup>, Qiang Li<sup>b</sup>, Satyanarayan Pati<sup>c</sup>, Srikanth Patil<sup>c</sup>, Narendra Baddu<sup>c</sup>

<sup>a</sup>Roche, <sup>b</sup>Accenture, <sup>c</sup>Involead,

Correspondence: [suyash.mishra@roche.com](mailto:suyash.mishra@roche.com), [qiang.i.li@accenture.com](mailto:qiang.i.li@accenture.com), [satyanarayan.pati@involead.com](mailto:satyanarayan.pati@involead.com),  
[srikanth.patil@involead.com](mailto:srikanth.patil@involead.com), [baddu.narendra@involead.com](mailto:baddu.narendra@involead.com)

## Abstract

Vision–Language Models (VLMs) have shown strong performance on multimodal reasoning tasks, yet most evaluations focus on short videos and assume unconstrained computational resources. In industrial settings such as pharmaceutical content understanding, practitioners must process long-form videos under strict GPU, latency, and cost constraints, where many existing approaches fail to scale. In this work, we present an industrial GenAI framework that processes over 200,000 PDFs, 25,326 videos across eight formats (e.g., MP4, M4V, etc.), and 888 multilingual audio files in more than 20 languages. Our study makes three contributions: (i) an industrial large-scale architecture for multimodal reasoning in pharmaceutical domains; (ii) empirical analysis of over 40 VLMs on two leading benchmarks (VideoMME and MMBench) and proprietary dataset of 25,326 videos across 14 disease areas; and (iii) four findings relevant to long-form video reasoning: the role of multimodality, attention mechanism trade-offs, temporal reasoning limits, and challenges of video splitting under GPU constraints. Results show 3–8× efficiency gains with SDPA attention on commodity GPUs, multimodality improving up to 8/12 task domains (especially length-dependent tasks), and clear bottlenecks in temporal alignment and keyframe detection across open- and closed-source VLMs. Rather than proposing a new "A+B" model, this paper characterizes practical limits, trade-offs, and failure patterns of current VLMs under realistic deployment constraints, and provide actionable guidance for both researchers and practitioners designing scalable multimodal systems for long-form video understanding in industrial domains.

\* Patent application to EPO: 25175653.2

## 1 Introduction

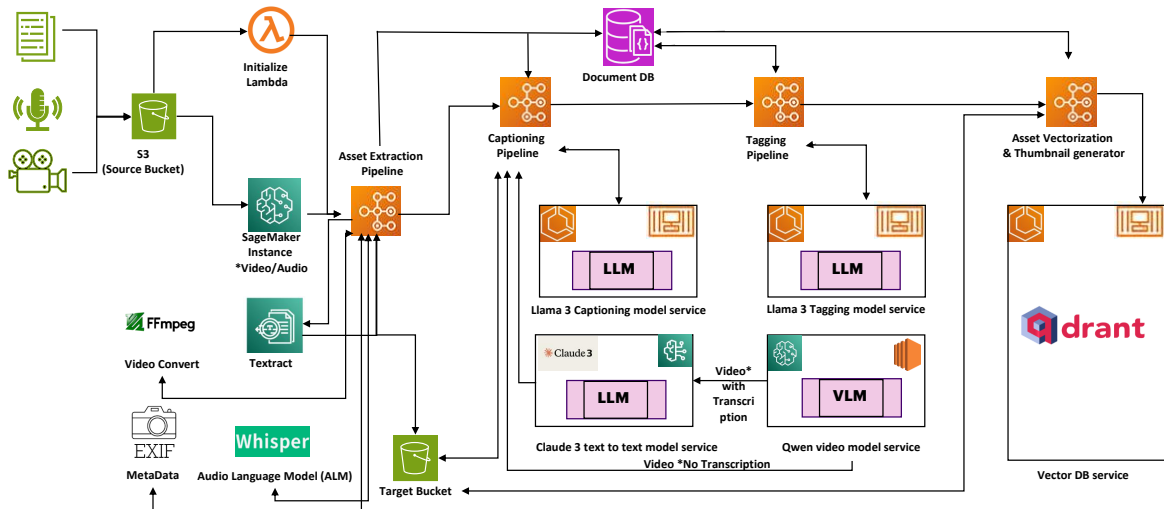
Large Language Models (LLMs) such as GPT-4 (Ouyang et al., 2022) and LaMDA (Thoppilan

et al., 2022) have significantly improved access to complex information in domains including health-care and public services (H&PS) (Li et al., 2025a; Zhang et al., 2026; Mishra et al., 2026; Ozmermer and Li, 2023). While these models excel at text-based reasoning, industrial use cases increasingly involve multimodal content spanning tables, graphs, charts, text, video and audio (Fu et al., 2024a). In pharmaceutical industry, such assets include clinical trial recordings, conference lectures, promotional materials, and multilingual patient educational videos (Zhang et al., 2023). Manual perception of these resources is inconsistent, costly, and infeasible at scale, particularly given compliance & computation constraints.

Practical use cases including chat agents (OpenAI, 2024), briefing agencies (OpenAI, 2023), document searching (Lewis et al., 2021), text/video summarization (Li et al., 2025b), and document quality checks (Yang et al., 2024) deliver transformative impacts on industry.

Current research has emphasized novel architectures or benchmark leaderboards (Singhal et al., 2022), but less attention has been given to how existing VLMs scale under practical GPU budgets, long-video scenarios, and compliance workflows. Challenges remain in producing reliable VLMs with growing amount of video data, reducing hallucinations and memory cost, improving quality of long video content reasoning, and addressing computational bottlenecks (Qu et al., 2025). Industrializing VLMs for large-scale data especially using closed-source models on proprietary data is also becoming increasingly urgent.

This work centers on a core **Research Question (RQ)**: how to scale VLM-based multimodal reasoning on long-form pharmaceutical videos under realistic industrial GPU constraints. To address this, we open-source a large-scale GenAI platform architecture designed for natural language search for (H&PS) users. The platform ingested over 200,000



**Figure 1:** System architecture of our GenAI platform for Natural Language (NL) search integrating LLMs, ALMs, and VLMs. The platform processed 25,326 videos, 888 audios covering > 20 languages.

PDFs, 25,326 videos across 8 formats (e.g., MP4, M4V, MSVideo, etc.), and 888 audio files spanning 20+ languages. We evaluate more than 40 VLMs using leading benchmarks (Video-MME, MMBench) and our proprietary dataset covering 14 disease areas. Our contributions including:

- A industrial multimodal architecture framework (in Figure 1) for scalable ingestion, captioning, and retrieval of large amount of data.
- Four key findings from industrial deployment: (1) multimodality boosts VLM performance across most tasks (8/12); (2) attention mechanisms show GPU-specific trade-offs; (3) both open- and closed-source models struggle with temporal alignment and keyframe reasoning; and (4) long-video splitting & compression is more error-prone rather than efficient.
- Extension of Video-MME with new subtasks (summarization, keyframe evaluation) along with newly designed evaluation schema using the Knowledge Graph, shown in Alg. 1, 2, 3.

## 2 Related Work

Vision-Language Models (VLMs) have advanced significantly, with several state-of-the-art models demonstrating strong performance across multimodal tasks (Fu et al., 2024a). Qwen-VL (Bai et al., 2023), improved vision-language alignment through extensive pretraining and data scaling. Recent multimodal Open-source models, such as

CLIP (Radford et al., 2021), BLIP (Li et al., 2022), FLAVA (Singh et al., 2022), and OFA (Wang et al., 2022), have demonstrated strong zero-shot and fine-tuned performance in various vision-language tasks across recognition, captioning, and retrieval. More recent VLMs—including Gemini Pro/Flash (Gemini Team, Google: Petko Georgiev and 1135 other authors, 2024), AdaReTaKe (Lourentzou et al., 2021), and Qwen2-VL (Bai et al., 2023), have pushed the boundaries of video-based multi-modal understanding by improving reasoning, temporal alignment, and multi-modal fusion strategies. As shown in recent leader-boards (Fu et al., 2024a), Qwen2-VL (Bai et al., 2023) achieves state-of-the-art results in long & short video caption tasks, reinforcing its role as a leading open-source VLMs.

Meanwhile, several benchmarks have been proposed to systematically evaluate the capabilities of MLLMs. Video-MME (Fu et al., 2024a) introduces an evaluation framework specifically designed for VLM, evaluating their understanding of dynamic and multimodal content. Additionally, MME (Fu et al., 2023) provides a comprehensive benchmark for the evaluation of more general multimodal LLMs, while MME-Survey (Fu et al., 2024b) offers a detailed review of existing evaluation methodologies. The MME-RealWorld benchmark (Zhang et al., 2024a) further extends this evaluation to real-world, high-resolution scenarios, testing the robustness and generalization of multimodality beyond synthetic datasets.

What’s more, efficient attention mechanisms

have been crucial for scaling large multimodal models (Face, 2024). FlashAttention (Dao, 2023) and Scaled Dot-Product Attention (SDPA) (Vaswani et al., 2017) have played significant roles in improving efficiency in transformer-based architectures. FlashAttention reduces memory overhead and computational costs by optimizing key-query-value matrix operations, making it well-suited for large-scale applications. Similarly, SDPA, widely implemented in frameworks like Hugging Face’s Transformers, optimizes inference performance on GPUs, particularly with hardware accelerations e.g. AMD ROCm and NVIDIA TensorRT (Face, 2024).

### 3 Dataset And Experimental Settings

**Table 1:** Statistics of Our Property Dataset.

Category	Details
VLM Models Covered	42, GPT series include GPT-4, Gemini 1.5 Pro, 2.0-Flash, Qwen-7B-VL, Qwen-72B, Owen VL Max, LLaVA-Video, Oryx-1.5, InternVL 2.5, Aria VideoLLaMA series, VideoChat Flash, NVLA, GPT-4o, Claude 3.5 Sonnet, TimeMarker, MiniCPM-V 3.2, MiniCPM-V 2.6, InternVL series, ST-LLaMA Video-XL, VITA-1.5, Kangaroo, Video-CCAM, ShareGemini, SIMM, Chat-Uni-VL 1.5, VideoChat2 Mistral, ShareGPT-4V Video,
ALM Models Covered	Whisper-turbo and Whisper-large V2
Number of Videos	Over <b>25,326</b> .
Number of Audios	Over <b>888</b> .
Covered Variants	Over <b>14</b> Diseases Ares. From Nephrology, Ophthalmology, Oncology, ... to Hematology, Immunology, Dermatology.
Covered video format Types	<b>8</b> . MP4, M4V, QuickTime, WMV, WebM, MSVideo, MPG, and 3GPP
Covered audio format Types	<b>4</b> . '.mp3', '.wav', '.m4a', '.flac'
Covered Video Lengths	< 2 mins to over 3 hours
Language Types	Over 20 languages, including German, Italian, English, Mandarin Hokkien, Hindi, Korean, French, Dutch, Spanish, and more.

Here, we primarily adopt two well-established MLLM benchmarks, Video-MME (Fu et al., 2024a) and MMBench (Liu et al., 2023), along with evaluations of more than 40+ VLMs, as well as our pharmacy property dataset shown in Table 1, and Figure 4. The benchmarks allow standardized comparisons, while our dataset provides a realistic testbed for long-form industrial content.

Video-MME (Fu et al., 2024a) is the first full-spectrum multi-modal evaluation benchmark designed specifically for video-based MLLMs. It stands out from existing benchmarks with several key features: (1) Diversity in video types, covering six primary visual domains with 30 subfields to ensure broad scenario generalizability; (2) Temporal coverage, including short-, medium-, and long-term videos ranging from 11 seconds to 1 hour. Video-MME includes 900 manually selected videos totaling 254 hours, annotated with 2,700 question-answer pairs.

MMBench (Liu et al., 2023) is designed to assess our findings across diverse visual understanding tasks, including video recognition, captioning,

visual question answering (VQA), and reasoning. Furthermore, our findings are tested from our industry data, includes 25,326 videos across eight formats (MP4, M4V, QuickTime, WMV, WebM, MSVideo, MPG, 3GPP) and 888 audio files across four formats (.mp3, .wav, .m4a, .flac). The content spans 14 disease areas, including oncology, hematology, immunology, ophthalmology, neuroscience, dermatology, nephrology, and respiratory disease, covering >20 languages. Video lengths range from under 2 minutes to more than 3 hours, reflecting the diversity of clinical trial recordings, medical lectures, and patient education materials.

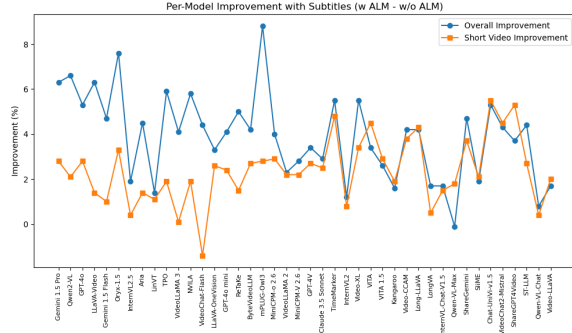
We benchmarked 42 VLMs / ALMs, including GPT-4 series (Ouyang et al., 2022), Gemini 1.5 Pro (Gemini Team, Google: Petko Georgiev and 1135 other authors, 2024), InternVL-Chat-V1.5 (Chen et al., 2024), and LLaVA-NeXT-Video (Zhang et al., 2024b), and Whisper etc. Each model was tested under default configurations and resource constrained GPU environments (NVIDIA A100 vs. A10G). Prompts are listed in Appendix Table 11.

### 4 Business And Technical Impact

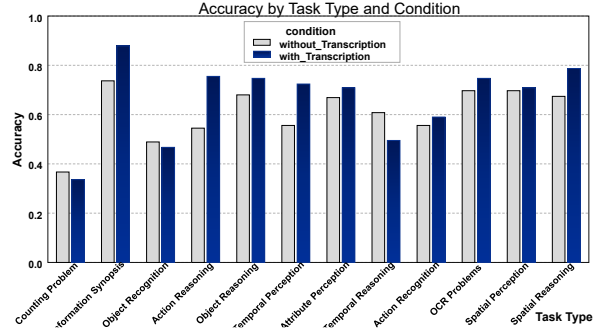
Finding reliable content remains a major challenge for healthcare professionals (HCPs) and patients. Traditional search methods are inefficient, leading to under-utilized assets and duplicated content creation. VLM-based NL search system streamlines discovery, reuse, and accessibility of video content. In production pilots, the system reduced the time required to create patient-facing materials by **66%**, accelerating workflows and improving consistency. On Video-MME, the end-to-end processing time, from voice abstraction by Ffmpeg, Whisper Turbo transcription to Bedrock LLM improvement on VLM video summary & captions storage in the database, averaged 2.2 minutes per longer video, 1.7 minutes per medium-length video, 1.6 minutes per short video. Compared to manual inspection (252.5 hours for the dataset), this represents a **94.4%** reduction in effort for long videos and a **88%** reduction for overall video categories. Scaling to the entire dataset implies a savings of approximately c.a. **224.3 hours**.

### 5 Main Results

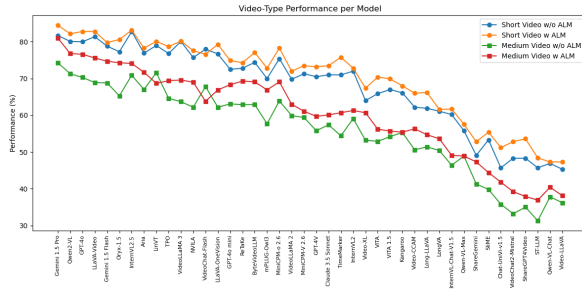
In this section, we first evaluate with the top two benchmarks, Video-MME (Fu et al., 2024a) and MMBench (Liu et al., 2023), as well as our proprietary dataset for multimodal vision language



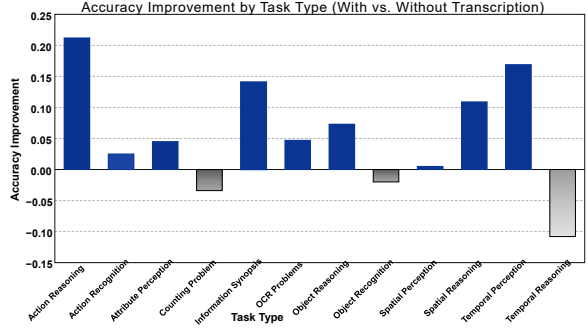
(a) Per-model improvement with ALM (overall vs. short).



(b) Accuracy by Subtask Type with/without Transcription.



(c) Video-type performance per model.



(d) Accuracy by Subtask Type with/without Transcription.

**Figure 2: Multimodality matters.** Combining metadata and voiceovers using LLM/ALMs could improve VLM summaries and understanding. Especially effective for longer videos, and for tasks like action recognition, object recognition, and OCR, but may negatively impact temporal reasoning tasks.

Task Type	With ALM	Without ALM	$\Delta$
Action Reasoning	0.759	0.545	+0.213
Action Recognition	0.589	0.564	+0.025
Attribute Perception	0.671	0.646	+0.025
Counting Problem	0.337	0.372	-0.035
Information Synopsis	0.879	0.737	+0.142
OCR Problems	0.744	0.698	+0.046
Object Recognition	0.469	0.490	-0.021
Spatial Perception	0.708	0.704	+0.005
Spatial Reasoning	0.789	0.680	+0.108
Temporal Perception	0.733	0.563	+0.171
Temporal Reasoning	0.500	0.615	-0.115
<b>Average</b>	<b>0.683</b>	<b>0.626</b>	<b>+0.057</b>

**Table 2:** Effect of multimodality on VLM performance (Video-MME) and task-level impact of audio transcription. Adding ALM-based voice transcriptions improves 8/12 task domains. Largest gains: Action Reasoning (+0.213), Information Synopsis (+0.142), Temporal Perception (+0.171). Negative effects: Counting (-0.035), Object Recognition (-0.021), Temporal Reasoning (-0.115).

Model	w/o	w/	$\Delta$
Gemini 2.5 Pro	84.7	85.2	+0.5
Gemini 1.5 Pro	75.0	81.3	+6.3
Qwen2-VL	71.2	77.8	+6.6
GPT-4o	69.0	77.2	+8.2
LLaVA-Video	76.0	76.9	+0.9
Gemini 1.5 Flash	72.6	75.0	+2.4
Oryx-1.5	67.3	74.9	+7.6
InternVL2.5	67.6	74.0	+6.4
Aria	70.3	72.1	+1.8
LinVT	65.6	71.7	+6.1
TPO	66.2	71.5	+5.3
<b>Average</b>	<b>68.4</b>	<b>72.3</b>	<b>+3.9</b>

**Table 3:** Model-level impact of incorporating audio transcriptions. Overall accuracy increases from 58.4% to 62.3% (+3.9). Gains are largest for GPT-4o (+8.2), Oryx-1.5 (+7.6), and Qwen2-VL (+6.6). Newer models such as Gemini 2.5 Pro show smaller gains (+0.5), reflecting diminishing returns as models advance.

models (MVLMs), focusing on widely recognized 40+ vision language models (VLMs). Results are structured around four interesting findings.

**Multimodality Matters. Combining metadata and voiceovers using LLMs/ALMs enhances VLM summaries and understanding,** although ALM-only pipelines can already achieve competitive content summarization. Across 42 evaluated VLMs, including GPT-4, we observe a consistent

positive trend in performance when audio transcriptions are incorporated. Notably, this improvement strongly correlates with **video length**. Overall accuracy increased from 58.4% to 62.3% (+3.9 points). For short videos, accuracy rose from 67.7% to 70.0% (+2.3 points), for medium videos from 69.3% to 74.3% (+5.0 points), and for long videos from 61.7% to 69.6% (+7.9 points). As video length increases, video Q&A Reasoning accuracy

declines significantly from over 80% on short clips to below 50% on long-form videos, highlighting the persistent challenge of long video understanding. Incorporating subtitles and audio information mitigates this issue, yielding improvements of up to 7.9% for long videos, as shown in Figure 2 (a,b,c).

When further examining multimodality with sub-task improvements, we can see that it improves 8/12 task domains, such as action reasoning, action recognition, information synopsis, OCR problems, and object reasoning, with improvements ranging from 2% to 20%. Here, we see the benefits of voice information in understanding procedural or sequential tasks. However, **temporal-related tasks remain challenging**, with several models showing reduced performance when processing both audio and visual inputs simultaneously (Figure 2d). This suggests that current VLMs still struggle with synchronizing multimodal context over extended time spans. Cross-modal attention mechanisms are typically optimized for semantic alignment rather than temporal synchronization. Our results suggest that while multimodal inputs improve overall understanding, they can introduce temporal noise, leading to higher rates of misaligned timestamps and incorrect segment boundaries in long videos (see Appendix Table 6). Additionally, we also noticed a decline in completeness of answered questions, dropping from 817 to 699, as CUDA out-of-memory (OOM) issues frequently occurred when reasoning with both audio & visual information.

**Attention Mechanisms Matter.** The VLM community has shown growing interest in FlashAttention-2 (Dao, 2023), as it provides an efficient implementation of the standard attention mechanism by parallelizing computation across sequence length and optimizing GPU memory usage. While FlashAttention significantly accelerates inference, most leading VLMs are trained and benchmarked on high-end GPUs such as the NVIDIA A100, which inherently benefits from optimized tensor operations and high memory bandwidth.

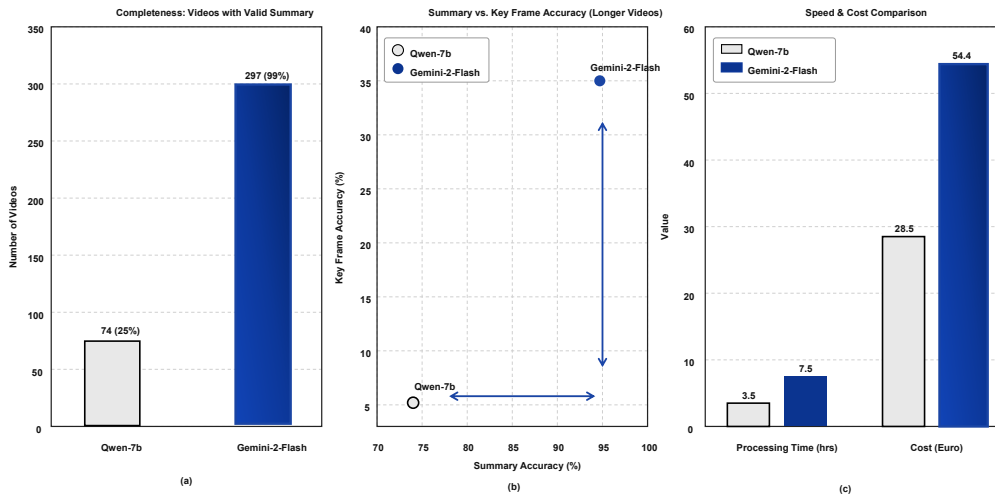
However, **FlashAttention-2 also exhibits strict GPU architecture dependencies**, requiring Torch > 3.6 and being optimized primarily for NVIDIA A100 or select AMD GPUs (e.g., Instinct MI210, MI250). In addition, dtype constraints limit its operation to fp16 or bf16, which restricts deployment in many industrial environments. **Deploying these mechanisms on standardized A10G Tensor Core GPUs—rather than on large A100 UltraCluster setups—introduces major**

**cost and compatibility challenges.** For instance, an A100 UltraCluster costs approximately \$40.96 per hour, whereas AWS G5 instances (A10G GPUs) range from \$5.67 to \$8.14 per hour, making A100-based inference roughly eight times more expensive. Empirically, FlashAttention-2 performs best on A100 GPUs and remains a popular choice for research benchmarks. In contrast, Scaled Dot-Product Attention (SDPA) (Vaswani et al., 2017) proves to be a more practical option for A10G-based deployments, offering up to a 4× speed improvement. Notably, SDPA achieved higher accuracy (58.73%) than FlashAttention (54.81%) when evaluated at FPS = 0.1 on the 2,700 Q&A Video-MME benchmark. SDPA completed inference in 4 hours and 37 minutes, compared to 7 hours and 40 minutes for FlashAttention-2, as summarized in Table 4, 8, 9. These findings suggest that **VLMs exhibit strong GPU architecture and FPS dependencies**, implying that the optimal attention mechanism varies with the hardware environment. SDPA (Vaswani et al., 2017) can also invoke FlashAttention and other memory-efficient attention kernels when needed, and native SDPA support is now expanding in the Transformers library. This underscores that selecting a well-matched—rather than the newest—attention mechanism can yield superior efficiency under realistic industrial GPU constraints.

**Time Localization Remains Challenging:** We evaluated VLM performance across three dimensions: speed, cost, and output completeness. Our findings indicate that Gemini 2.5 Flash is slower and more expensive, requiring 7–8 hours to process 300 videos at a cost of Euro 46.73 on Google Cloud. In contrast, Qwen-7B (Bai et al., 2023) demonstrated substantially faster processing, completing each video in 1 mins on an AWS EC2 g5.24xlarge instance, priced at \$8.14 per hour. However, when running at 0.01 FPS with SDPA attention, Qwen-7B required approximately 3–4 hours to process 300 videos, as shown in Figure 3. In terms of **output completeness**, Gemini 2.5 Flash outperformed Qwen-7B, generating valid summaries for 297/300 videos (with acc.94.6%), while Qwen-7B produced only 74 valid summaries (with acc.74.3%). Additionally, **key frame extraction remained challenging for both models**, with Gemini 2.5 Flash achieving a keyframe accuracy of 35.1% (26/74), while Qwen-7B reached only 5.4% (4/74). For long videos, Gemini 2.5 Flash required roughly 3 mins per video, while Qwen-7B remained below one minute

**Table 4:** Comparison of speed, accuracy, and completeness across leading attention mechanism on all length videos. SDPA yields higher accuracy and 4x faster runtime on commodity GPUs (A10G), while FlashAttention favors high-end A100 GPUs.

Experiments	Processing Time	Total Answered (%)	Correct Answered (%)
SDPA (0.1 FPS)	4h 37m 2s	37%	58.73%
SDPA (0.01 FPS)	2h 12m 37s	87%	48.40%
FlashAttention (0.1 FPS)	7h 40m 17s	100%	54.81%
FlashAttention (0.01 FPS)	1h 50m 11s	100%	48.55%



**Figure 3:** Time Localization Challenges for Open-Source and Closed-Source VLMs. Both top open-source and commercial models struggle with key frame detection, showing low accuracy (5-35%) and incorrect timestamps. Summaries are much more accurate, ranging from 75-95%.

per video. Despite these differences, **both leading open- and closed-source VLMs struggle with accurate temporal reasoning**, particularly in maintaining keyframe alignment and coherence across long video sequences (see Table 6). This highlights a persistent limitation in time-dependent video understanding for current VLM architectures.

**Trade-off of Long Video Splitting.** Processing long videos poses significant challenges, particularly on GPU-constrained instances. Out-of-memory (OOM) errors occur frequently, even when reducing frames per second (FPS) or lowering video resolution. A common mitigation strategy involves video splitting, compression, cutting. However, our analysis shows that these methods do not yield meaningful speed improvements and instead introduce additional challenges related to temporal alignment. **Splitting videos into multiple segments disrupts temporal consistency, making it increasingly difficult for VLMs to maintain coherent event sequences.** Moreover, compressing video files using standard tools such as FFmpeg requires substantial preprocessing time. Although shorter segments can be processed individually, the lack of contextual continuity causes VLMs to **focus on superficial cues**—such as logos, text color, or general stylistic attributes—rather than on semantic

content. This leads to redundant descriptions and weaker keyframe alignment, as shown in Table 7.

## 6 Conclusion

In this work, we introduce an industrial framework for large-scale VLM-based video processing and NL search. Unlike prior studies proposing new architectures, our contribution lies in benchmarking, scaling, and analyzing existing VLMs under realistic GPU and compliance constraints. Our evaluation yields four key findings: (1) **Multimodality matters**—incorporating ALMs, transcriptions, and metadata improves video reasoning beyond static or bi-modal setups; (2) **Attention mechanisms matter**—matching attention to GPU architecture improves efficiency; (3) **Open- and closed-source VLMs** perform well in video summarization but still struggle with keyframe detection and timestamping; and (4) **Splitting long videos** often increases runtime and misalignment errors rather than improving efficiency. Beyond these insights, we extend Video-MME with new subtasks (summarization, keyframe extraction) and propose a knowledge-graph evaluation schema. Future research should explore financial and manufacturing video domains under constrained hardware.

## 7 Limitations

We want to emphasize that this work is not intended to introduce a new VLM architecture, but rather to empirically characterize the capabilities, limitations, and failure modes of existing models under realistic long-video and hardware constraints. A main focus of this study is the justification of the pharma specific benefits, e.g. how to scale VLMs usage in industry GPU constrained hardware setting for business application. We provide a baseline comparison of more than 42 VLMs using both the Video-MME benchmark and our proprietary dataset. Future work should extend this line of research to other regulated domains, such as financial services and manufacturing videos e.g. Figure 7, to further validate the generalization of our solution blueprint.

## 8 Acknowledgments

We sincerely thank Samik Adhikary and Puneet Srivastava for their sponsorship support from Roche. We also appreciate the insightful discussions and technical assistance provided by Janina Kummerfeldt, Philipp Müller, and Kathrin Schwan from Accenture, Jennifer McGuire’s business support from Roche, as well as Aditya Kumar’s paper experiment support from Involead. This platform, RICi, would not have been possible without their contributions. We further extend our gratitude to the backend engineering teams supported RICi development, as well as to the healthcare professionals (HCPs), testers, and Roche Lab users whose consistent feedback brought our Content Finder use cases to life and enabled continuous improvement. This paper serves as our academic contribution, where we formalized experiments and evaluation methodologies using academic benchmarks. Through this work, we aim to share industry lessons learned and valuable large-scale GenAI experiments in the pharmaceutical domain.

## References

Ai central (american college of radiology) – fda-cleared imaging ai directory. <https://aicentral.acrdsi.org/>. Over 200 FDA-cleared imaging AI products cataloged; Accessed 2025-10-07.

Artificial intelligence (ai/ml)-enabled medical devices. <https://www.fda.gov/medical-devices/software-medical-device-samd/artificial-intelligence-enabled-medical-devices>. Accessed 2025-10-07.

Official FDA list of AI-enabled medical devices; Accessed 2025-10-07.

Isic: International skin imaging collaboration archive. <https://www.isic-archive.com/>. Dermoscopic/clinical *image* repository widely used for AI benchmarking.

2018. Idx-dr: De novo summary (den180001). [https://www.accessdata.fda.gov/cdrh\\_docs/reviews/DEN180001.pdf](https://www.accessdata.fda.gov/cdrh_docs/reviews/DEN180001.pdf). U.S. Food and Drug Administration De Novo classification for autonomous DR detection.

Sharib Ali, Debesh Jha, Noha Ghatwary, and et al. 2023. *A multi-centre polyp detection and segmentation dataset for generalisability assessment (polypgen)*. *Scientific Data*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

K. Behara and et al. 2024. *AI in dermatology: a comprehensive review into skin disease identification*.

Hanna Borgli and et al. 2020. *Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy*. *Scientific Data*.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. 2024. *How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites*. *arXiv preprint arXiv:2404.16821*.

Tri Dao. 2023. *Flashattention-2: Faster attention with better parallelism and work partitioning*. *arXiv preprint arXiv:2307.08691*.

L. Dick and et al. 2024. *Automated analysis of operative video in surgical training: a scoping review*.

Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. 2018. *The unmanned aerial vehicle benchmark: Object detection and tracking*. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 370–386.

Hugging Face. 2024. *Transformers performance inference on gpu*. Accessed: 2024-03-10.

Ghazala Fatima, Imran Khan, Saeed Farooq, et al. 2024. *A comprehensive review of advances in digital pathology*. *Diagnostics*.

- Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.
- Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. 2024a. Videomme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.
- Chaoyou Fu, Yi-Fan Zhang, Shukang Yin, Bo Li, Xinyu Fang, Sirui Zhao, Haodong Duan, Xing Sun, Ziwei Liu, Liang Wang, et al. 2024b. Mme-survey: A comprehensive survey on evaluation of multimodal llms. *arXiv preprint arXiv:2411.15296*.
- Gemini Team, Google: Petko Georgiev and 1135 other authors. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- N. Ghamsarian and et al. 2024. Cataract-1k: A large-scale cataract surgery video dataset for surgical workflow analysis. *Scientific Data*.
- Md. Haque et al. 2025. Demographic bias in public remote photoplethysmography datasets. *npj Digital Medicine*.
- Bin Huang et al. 2023. Challenges and prospects of visual contactless physiological monitoring. *npj Digital Medicine*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.
- Johann Li, Guangming Zhu, and Cong Hua. 2021. A systematic collection of medical image datasets for deep learning. *arXiv preprint arXiv:2106.12864*.
- Junnan Li, Ramprasaath R. Selvaraju, Akhilesh D. Gotmare, Shafiq Joty, Caiming Xiong, and Steven C.H. Hoi. 2022. Blip: Bootstrapped language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 12888–12900.
- Qiang Li, Mingkun Tan, Xun Zhao, Dan Zhang, Daoan Zhang, Shengzhao Lei, Anderson S. Chu, Lujun Li, and Porawit Kamnoedboon. 2025a. How LLMs react to industrial spatio-temporal data? assessing hallucination with a novel traffic incident benchmark dataset. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 36–53. Association for Computational Linguistics.
- Zongxia Li, Xiyang Wu, Hongyang Du, Huy Nghiem, and Guangyao Shi. 2025b. Benchmark evaluations, applications, and challenges of large vision language models: A survey. *Preprint*, arXiv:2501.02189.
- Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2023. Mmbench: Is your multi-modal model an all-around player? *arXiv:2307.06281*.
- Ismi Lourentzou, Chen Chen, and ChengXiang Zhai. 2021. Adarenet: Adaptive reweighted semi-supervised active learning. In *Proceedings of the 14th ACM International Conference on Pervasive Technologies Related to Assistive Environments (PE-TRA)*, pages 1–8.
- O. Mahmoud and et al. 2024. Catstep: Automated cataract surgical phase classification using deep learning. *Ophthalmology Science*.
- Suyash Mishra, Qiang Li, Srikanth Patil, and Anubhav Girdhar. 2026. From understanding to engagement: Personalized pharmacy video clips via vision language models (vlms). *Preprint*, arXiv:2601.05059.
- R. et al. Najjar. 2023. Redefining radiology: A review of artificial intelligence. *PMC*.
- OpenAI. 2023. Chatgpt for intelligence briefing: A case study. Accessed: March 2025.
- OpenAI. 2024. Chatgpt: Optimizing language models for dialogue. Accessed: March 2025.
- T. Oshika and et al. 2024. Artificial intelligence applications in ophthalmology.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.
- Evrin Ozmermer and Qiang Li. 2023. Self-supervised learning with temporary exact solutions: Linear projection. In *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*, pages 1–7.
- L. et al. Pinto-Coelho. 2023. How artificial intelligence is shaping medical imaging. *PMC*.
- Xiaoye Qu, Jiashuo Sun, Wei Wei, Daizong Liu, Jianfeng Dong, and Yu Cheng. 2025. Look, compare, decide: Alleviating hallucination in large vision-language models via multi-view multi-path reasoning. In *Proceedings of the 31st International Conference on Computational Linguistics*. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark,

- Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8748–8763.
- Manuel Sebastián Ríos and et al. 2023. [Cholec80-cvs: An open dataset with an evaluation of strasberg’s critical view of safety for ai](#). *Scientific Data*.
- H. Saeidi and et al. 2022. [Autonomous robotic laparoscopic surgery for intestinal anastomosis](#). *Science Robotics*.
- Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Dhruv Mahajan, Xinlei Wu, Christoph Feichtenhofer, Trevor Darrell, and Ross Girshick. 2022. [Flava: A foundational vision and language model for generalized multimodal learning](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15638–15650.
- Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2022. [Large language models encode clinical knowledge](#). *arXiv preprint arXiv:2212.13138*.
- K. et al. Strzpek. 2023. [Quantitative and qualitative analysis of agricultural fields with uavs](#). *PMC*.
- Amanda Tan-Garcia et al. 2025. [Computational pathology in the age of artificial intelligence](#). *Histopathology*.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. [Lamda: Language models for dialog applications](#). *arXiv preprint arXiv:2201.08239*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*.
- D. et al. Wang. 2025. [A survey of unmanned aerial vehicles and deep learning in precision agriculture](#). *ScienceDirect*.
- Peng Wang, An Yang, Rui Men, Ming Zhou, Yinan Zhang, Junyang Lin, Xu Sun, Shuai Li, Houqiang Wang, and Luo Si. 2022. [Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework](#). *arXiv preprint arXiv:2202.03052*.
- Jing Wei, Zhanqing Li, Alexei Lyapustin, Jun Wang, Oleg Dubovik, Joel Schwartz, Lin Sun, Chi Li, Song Liu, Tong Zhu, et al. 2023. [First close insight into global daily gapless 1 km pm<sub>2.5</sub> pollution, variability, and health impact](#). *Nature Communications*, 14:8349.
- X. Xu and et al. 2024. [The application of artificial intelligence in diabetic retinopathy: progress and challenges](#).
- Kaixun Yang, Mladen Raković, Zhiping Liang, Lixiang Yan, Zijie Zeng, Yizhou Fan, Dragan Gašević, and Guanliang Chen. 2024. [Modifying ai, enhancing essays: How active engagement with generative ai boosts writing quality](#). *Preprint*, arXiv:2412.07200.
- Moran Yanuka, Assaf Ben-Kish, Yonatan Bitton, Idan Szpektor, and Raja Giryes. 2025. [Bridging the visual gap: Fine-tuning multimodal models with knowledge-adapted captions](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10497–10518. Association for Computational Linguistics.
- A. Yilmaz and et al. 2024. [Derm12345: A large, multi-source dermatoscopic skin lesion dataset](#). *Scientific Data*.
- Dan Zhang, Fangfang Zhou, Yuwen Jiang, Felix Albu, Yuanzhou Wei, Xiao Yang, and Qiang Li. 2026. [Unleashing the power of self-supervised image denoising: A comprehensive review](#). *Computer Vision and Image Understanding*, 269:104786.
- Dan Zhang, Fangfang Zhou, Yuwen Jiang, and Zhengming Fu. 2023. [Mm-bsn: Self-supervised image denoising for real-world with multi-mask based on blind-spot network](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4189–4198.
- Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. 2024a. [Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans?](#) *arXiv preprint arXiv:2408.13257*.
- Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. 2022. [Bytetrack: Multi-object tracking by associating every detection box](#). In *Computer Vision – ECCV 2022*, volume 13682 of *Lecture Notes in Computer Science*, pages 1–21. Springer, Cham.
- Yuanhan Zhang, Bo Li, Haotian Liu, Yong Jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. 2024b. [Llava-next: A strong zero-shot video understanding model](#).

## A Appendix

In this section we provide the supplementary compiled together with the main paper includes:

- Evaluation Metrics and Knowledge Graph Evaluation Schema on Algorithm 1, 2, 3, and Figure 5;
- Ablation study on Frame Per Second (FPS) in Table 8, Table 9;
- Property Dataset distribution on Table 10, and VideoMME raw rata example on Table 5;
- Deliverable attributes of each VLM / ALM and Metadata on Figure 6;
- The training details and hyper-parameters of experiments including prompts lists in Table 11, output example on Table 6, 7;
- The business value case and area of impact of GenAI-driven Video Processing on Table 12, Table 13 and Figure 7.

### A.1 Evaluation Metrics

**Assigned accuracy scores strategies in Finding 3, Time Localization Challenges in Open-Source and Closed-Source VLMs.**

$$\text{Scores}_{a,g} = \frac{1}{n_a} \sum_{i=1}^{n_a} S(x) \quad (1)$$

where

$$S(x) = \begin{cases} 1 & \text{if } S_{a,i} = g_{a,i} \\ 0 & \text{if } S_{a,i} \neq g_{a,i} \end{cases}$$

where  $S$  is the Matching Node score,  $a \in A$  refers to an Key Frame or Summary scenarios,  $g$  refers to ground truth of timestamp, and  $n_a$  is the total number of valid video output (e.g., if 74 videos have valid JSON outputs, we match key frames to verify timestamp accuracy and compare summary accordingly).

### A.2 Summary & Key Frame evaluation using Knowledge Graph

To compare the quality of video summaries generated by VLMs, we then employ a knowledge graph-based method. This is particularly useful in scenarios where human-annotated ground truth is incomplete or unavailable, such as with large-scale video datasets in industry setting.

#### A.2.1 Knowledge Graph Construction

We use the NetworkX library with DiGraph to construct the knowledge graph, NetworkX library encapsulated so well where:

- Nodes represent extracted keyframes and conceptual entities (nouns or keywords) from the generated summaries.
- Edges represent semantic or temporal relationships between these concepts.
- The graph layout is generated using the `spring_layout` function, which implements the Fruchterman-Reingold force-directed algorithm, as shown in Algorithm 1.

#### A.2.2 Mathematical Basis Behind

The force-directed layout models the graph using physical analogies:

- **Repulsion:** All nodes repel each other according to **Coulomb’s law**.
- **Attraction:** Connected nodes attract each other like springs (**Hooke’s law**).

These forces iteratively adjust node positions until a stable configuration is reached, visually revealing clustering and coverage. Thereby:

- **Node Count:** Indicates the richness or breadth of extracted information.
- **Node to Node Distance:** Measures how widely concepts or key words are spread in the graph.
- **Distance to Central Node:** We compute shortest path lengths using Dijkstra’s algorithm to measure how far keyframe nodes are from the central summary node.as shown in Algorithm 2.

This is an emerging area with ongoing efforts to define metrics for summary evaluation without ground truth. Recent work from researchers at Google and Apple (Yanuka et al., 2025) (e.g., *Descriptiveness Recall, Contradiction Precision*, Cosine Similarity) highlights the need for new metrics when ground truth of video summary, Key frame captions are missing.

In summary, we introduce new task domains based on open-source Video-MME (Fu et al., 2024a) tasks, which previously lacked summary

and keyframe subtasks due to the manual effort required for key frame localization labeling. To address this, we propose a knowledge graph approach to compare the output quality of various VLMs, as shown in Algorithm 3. This approach provides visually interpretable and computationally supported method to assess summary & key frame quality using graph-based representations, forming the basis for future work on automated evaluation metrics in GenAI applications.

---

**Algorithm 1** Force-Directed Graph Layout (*Fruchterman-Reingold*) conceptually

---

1: **Let:**

$$d(u, v) \leftarrow \text{distance between nodes } u, v$$

$$k \leftarrow C \cdot \sqrt{\frac{A}{n}} \quad \triangleright$$

$C$  is constant,  $A$  is layout area,  $n$  is number of nodes,  $K$  is optimal distance between nodes.

2: **Forces:**

1. **Attractive force (between connected nodes):**

$$F_{\text{attr}}(d) = \frac{d^2}{k}$$

2. **Repulsive force (between all nodes):**

$$F_{\text{rep}}(d) = \frac{k^2}{d}$$

3: **Loop:** Apply forces iteratively until convergence or maximum iterations reached.

---



---

**Algorithm 2** *Dijkstra's Algorithm* for Shortest Paths

---

1: **Input:** Directed graph  $G = (V, E)$  with non-negative weights  $w(u, v) \geq 0$

2: **Input:** Source node  $s$

3: **Initialization:**

$$\text{distance}[v] \leftarrow \infty \text{ for all } v \in V$$

$$\text{distance}[s] \leftarrow 0$$

Initialize a priority queue  $Q$

4: **while**  $Q$  is not empty **do**

5:   Extract node  $u$  with minimum distance[ $u$ ]

6:   **for each** neighbor  $v$  of  $u$  **do**

7:     Update:

$$\text{dist}[v] \leftarrow \min(\text{dist}[v], \text{dist}[u] + w(u, v))$$

8:   **end for**

9: **end while**

10: **Output:** Shortest distances from  $s$  to all  $v \in V$

---



---

**Algorithm 3** Knowledge Graph Construction for *Summary and Key Frame Evaluation*

---

1: **Input:** JSON data with key frames from Gemini-2 Flash and Qwen-7B

2: **Output:** Visualized Knowledge Graph

3: **Step 1: Initialize Graph**

4: Create directed graph  $G \leftarrow \text{nx.DiGraph}()$ . Nodes ( $V$ )  $\rightarrow$  Individual entities in the graph (e.g., "Gemini-2 Flash", "Snow White in rags"). Edges ( $E$ )  $\rightarrow$  Directed connections between nodes (e.g., "Gemini-2 Flash"  $\rightarrow$  "Key Frames"). Attributes  $\rightarrow$  Additional properties of nodes/edges (e.g., color, size).

5: DiGraph  $G = (V, E)$  is defined as:  $V$ =nodes,  $E$ =(source, target), where each edge has a direction.

6: **Step 2: Add Core Nodes**

7: Add node *KeyFrames* with attributes (color: gray, size: 800)

8: Add node *VideoSummary* with attributes (color: gray, size: 600)

9: **Step 3: Connect Models to Core Nodes**

10: Add edge (e.g., *KeyFrames*)

11: Add edge (e.g., *VideoSummary*)

12: **Step 4: Add Key Frames for Each Model**

13: **for each** (*timestamp, description*) in **r.g. Gemini-2 Flash key frames do**

14:   Add node *description* with attributes (color: light blue, size: 400)

15:   Add edge (*KeyFrames, description*)

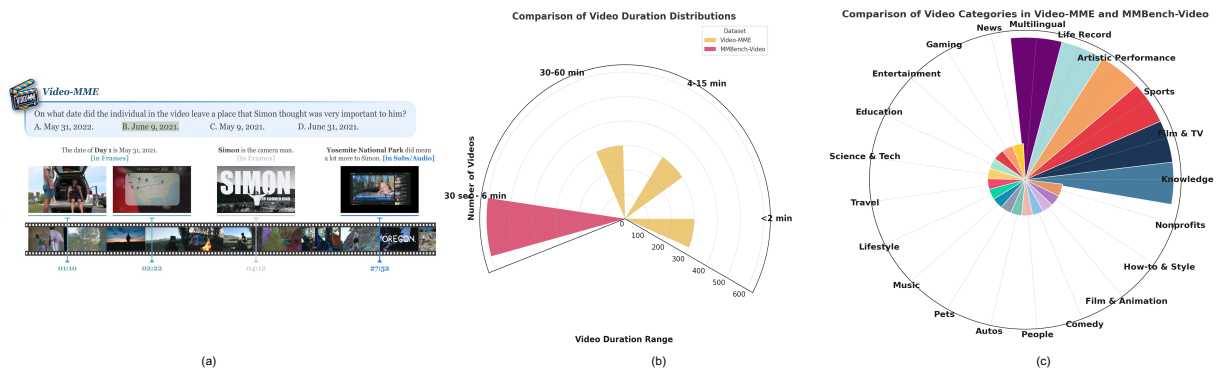
16: **end for**

17: **Step 5: Visualize the Graph**

18: Compute node layout  $pos \leftarrow \text{spring\_layout}(G, \text{seed}=42)$

19: Extract node colors and sizes for Knowledge Graph: Key Frames Evaluation.

---



**Figure 4:** Comparison of the Video-MME (Fu et al., 2024a) and MMBench-Video datasets (Liu et al., 2023) in terms of video categories and duration distributions. The Video-MME dataset consists of 900 videos spanning six primary visual domains with 30 subfields, categorized into 300 short-term (<2 min), 300 medium-term (4-15 min), and 300 long-term (30-60 min) videos. In contrast, the MMBench-Video dataset comprises approximately 609 videos across 16 major categories, with durations ranging all from 30 seconds to 6 minutes.

**Table 5:** Video-MME (Fu et al., 2024a) raw dataset structure, Q/A details in Json format.

Field	Details
Video ID	001
Duration	Short
Domain	Knowledge
Sub-Category	Humanity & History
URL	<a href="https://www.youtube.com/watch?v=fFjy93ACGo8">https://www.youtube.com/watch?v=fFjy93ACGo8</a>
VideoID	fFjy93ACGo8
Question ID	001-2
Task Type	Information Synopsis
Question	What is the genre of this video?
<b>Options</b>	
A	It is a news report that introduces the history behind Christmas decorations.
B	It is a documentary on the evolution of Christmas holiday recipes.
C	It is a travel vlog exploring Christmas markets around the world.
D	It is a tutorial on DIY Christmas ornament crafting.
<b>Answer</b>	<b>A</b>

**Table 6:** Comparison of Video Summarization: Gemini-2-Flash vs Qwen-7B. Here *the timestamps are all wrong*.

Category	Gemini-2-Flash	Qwen-7B
<b>Video ID</b>	P69idA8JO98	P69idA8JO98
<b>Duration</b>	Long	Long
<b>Domain</b>	Artistic Performance	Artistic Performance
<b>Summary</b>	A stage performance of <i>Snow White</i> . The Evil Queen consults the Magic Mirror, instructs Snow White to clean the castle, and the story unfolds as Snow White meets the Seven Dwarfs, receives the poisoned apple, collapses, and is revived by the Prince.	A fairy tale performance, likely <i>Snow White and the Seven Dwarfs</i> . The video introduces characters, a forest scene, a confrontation between a queen and a prince, interactions between Snow White and the dwarfs, and ends with a song.

#### Key Frames (Gemini-2-Flash)

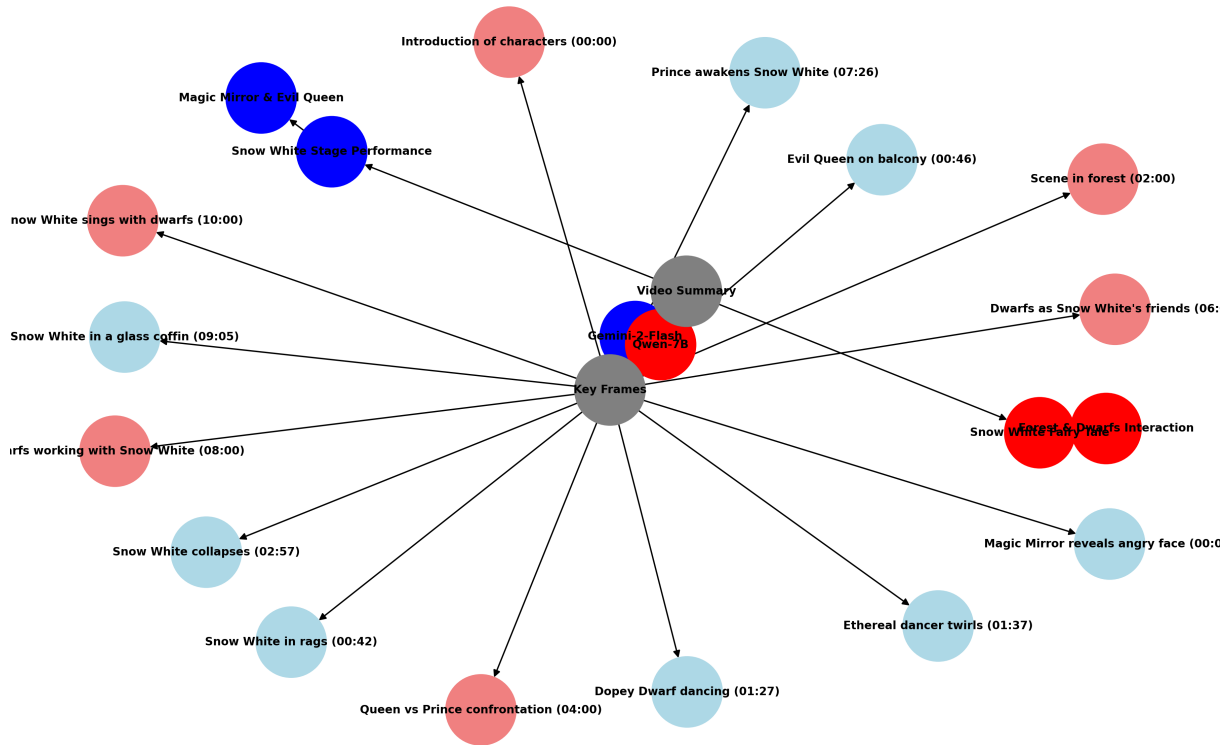
Time	Description
00:08	Magic Mirror reveals an angry face
00:42	Snow White in rags looking at her stepmother
00:46	The Evil Queen on a castle balcony
01:27	Dopey Dwarf dancing in silk costume
01:37	Ethereal dancer twirling with a deer
02:17	Snow White with basket approaching animals
02:57	Snow White collapses onto a stage of rocks
03:42	Snow White at a wishing well
05:09	The Evil Queen on a balcony speaking to a soldier
05:37	Snow White dancing in her new dress
06:09	Snow White and her prince hold hands
07:02	Snow White falls, animals mourn her
07:26	The Prince awakens Snow White with a kiss
08:00	Snow White is held up for celebration
08:07	Evil Queen standing on castle balcony
09:05	Snow White lies in a glass coffin as prince kneels

#### Key Frames (Qwen-7B)

Time	Description
00:00	Introduction of characters and setting
02:00	Scene with group of people in a forest
04:00	Confrontation between a queen and a prince
06:00	Introduction of the dwarfs as Snow White's friends
08:00	Scenes of the dwarfs working and interacting with Snow White
10:00	Snow White singing a song with the dwarfs

**Table 7:** With and Without ALM for Summary and Key Frame: Using ALM for voiceover transcription improves the relevance and precision of the summary content (*see the blue marked texts*). However, timing remains a significant issue.

Component	Output Text
<b>Prompt (Qwen Model)</b>	"Focusing on the content and workflow rather than specific logos or the color of text? After summarizing, list the key frames with brief captions in the format (00:00, caption). Ensure the analysis is accurate and avoid including any assumptions or extrapolations. Use a pharmaceutical expert domain perspective to enhance relevance and precision."
<b>Whisper Model (Voice-over Transcription)</b>	<p>" "id": 0, "start": 7.72, "end": 13.6, "text": " PCR of course refers to pathological complete response where once the patient has surgery" ...</p> <p><b>"Full Transcription":</b> " PCR of course refers to pathological complete response where once the patient has surgery the pathologist does not find any cancer at all and pleasingly over the last sort of 15-20 years we've seen improvements in systemic treatment to such an extent that certainly for HER2 positive breast cancers we are now able to expect 50-60% of patients who have a PCR following the neoadjuvant treatment and we know that the patients who have a PCR are the ones who are likely to do the best in the long term. "</p>
<b>Qwen Model Output (Before Transcription)</b>	<p><b>VLM Output:</b> "The video appears to be a medical demonstration, likely related to the Roche brand, which is known for its involvement in healthcare and diagnostics. The video showcases a 3D model of a human body with various organs highlighted, including the liver and brain. The model is displayed within a transparent box, possibly representing a medical device or a simulation environment.</p> <p><b>Key Frames with Captions:</b> 00:00 - Introduction: The video begins with a close-up of a transparent box containing a 3D model of a human body. 01:00 - Organ Highlighting: The liver and brain are highlighted in the model, indicating a focus on these organs. 02:00 - Interactive Display: The model rotates, providing a better view of the highlighted organs. 03:00 - Data Overlay: Digital text and graphics appear, likely representing medical data or information related to the highlighted organs. 04:00 - Conclusion: The video ends with a summary or conclusion, possibly emphasizing the importance of the highlighted organs or the role of the device in medical diagnostics. The video likely aims to educate viewers about the importance of the liver and brain in the human body and how the device or simulation can be used to study or diagnose related conditions." "</p>
<b>Qwen Model Output (After Transcription Integration)</b>	<p><b>VLM Output:</b> "Here is a summary focusing on the content and workflow from a pharmaceutical expert's perspective, along with key frames and brief captions"</p> <p>Summary: The video provides an explanation of the term "<i>pathological complete response</i>" (<i>pCR</i>) and its significance in cancer treatment, particularly for <i>HER2-positive breast cancer</i>. The speaker, a medical professional, <i>defines pCR as the absence of any detectable cancer cells in the surgical specimen after neoadjuvant (pre-operative) systemic treatment</i>. He highlights the remarkable progress made in systemic therapies over the past 15-20 years, <i>enabling 50-60% of HER2-positive breast cancer patients to achieve pCR following neoadjuvant treatment</i>. The speaker <i>emphasizes that patients who achieve pCR have the best long-term outcomes</i>, underscoring the importance of this endpoint in cancer management. <b>Key Frames:</b>(00:07) Introduction to the concept of pathological complete response (pCR).(00:19) Progress in systemic treatments improving pCR rates. (00:24) Specifically mentioning 50-60% pCR rates in HER2-positive breast cancer.(00:33) Linking pCR to better long-term outcomes for patients."</p>



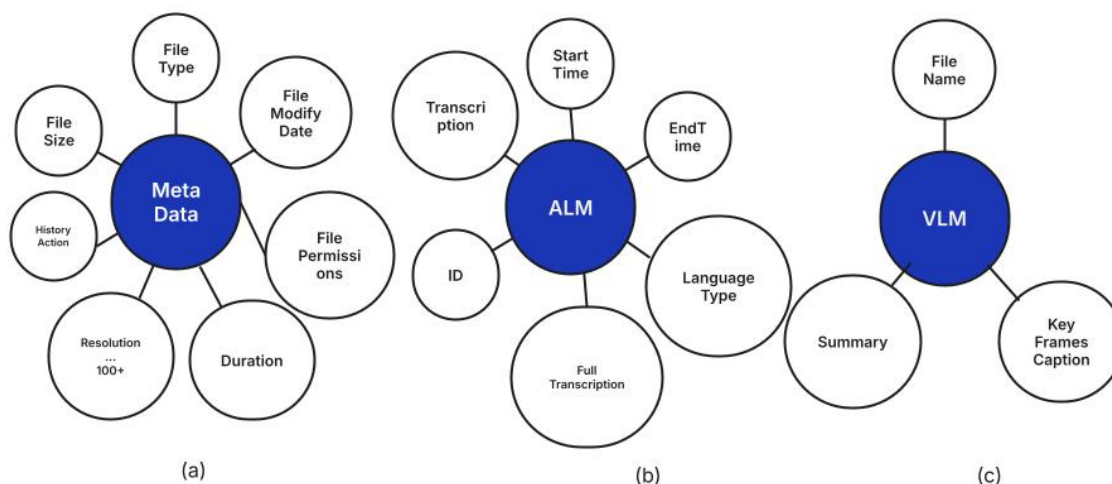
**Figure 5:** Knowledge graph for summary and key frames using Qwen and Gemini models. The knowledge graph visualizes the comparison between Gemini-2-Flash and Qwen-7B in summarizing a 'Snow White' stage performance. Each key frame from Gemini-2-Flash is marked in light blue and video summary in dark blue, while Qwen-7B's key frames are in light red, and video summary in red. The central node represents the key frames, with connections showing their relationships to each model's summary. Gemini-2-Flash emphasizes narrative elements such as the Magic Mirror, the Evil Queen, and the climax involving Snow White's revival, while Qwen-7B structures the story around broad thematic transitions like character introductions, forest scenes, and musical elements. This graph presents a structured comparison of the keyframes extracted by Gemini-2-Flash and Qwen-7B from a 'Snow White' performance. *The blue nodes represent Gemini-2-Flash's emphasis on theatrical storytelling, focusing on individual character moments, while the red nodes highlight Qwen-7B's broader narrative structure, including interactions between Snow White and supporting characters.* Additionally, the red nodes are more widely distributed, whereas the blue nodes are clustered more closely, indicating a difference in granularity and focus.

**Table 8:** Ablation Study on FPS (1): Attention Mechanism Dependence on FPS. Comparison of Speed, Accuracy, and Completeness Across Leading Attention Mechanisms on Short Videos (<120s). *This study highlights the strong dependence of FPS on each model's performance. For short videos, FlashAttention is recommended over SDPA.* \*Default settings as per the Video-MME benchmark (FPS = 1), with no additional audio ALM transcription fed into the Qwen model.

Experiments on <b>Short Videos*</b>	Processing Time	Total Answered (%)	Correct Answered (%)
SDPA (1 FPS)	44m 12s	6%	64.81%
FlashAttention (1 FPS)	1h 30m 12s	100%	<b>70.78%</b>
Experiments on <b>All Length Videos</b>	Processing Time	Total Answered (%)	Correct Answered (%)
SDPA (0.1 FPS)	4h 37m 2s	37%	<b>58.73%</b>
FlashAttention (0.1 FPS)	7h 40m 17s	100%	54.81%
SDPA (0.01 FPS)	2h 12m 37s	87%	48.40%
FlashAttention (0.01 FPS)	1h 50m 11s	100%	48.55%

**Table 9:** Ablation Study on FPS (2) *Reducing FPS does not necessarily help the Qwen model answer more questions correctly.* In fact, it can have a negative impact, as lower frames per second lead to missing information. Here, the completeness percentage increases from 37% to 87% significantly, but the accuracy drop from 58.73% to 48.40%. However, *with the support of audio ALM transcription, accuracy is maintained*, improving from 58.73% to 61.80% when FPS = 0.1, and from 48.40% to 49.36%. This further validates our first finding from a different perspective.

Experiments on All Length Videos	Total Answered (%)	Correct Answered (%)
SDPA (0.1 FPS) without Audio Transcription	37%	58.73%
SDPA (0.1 FPS) with Audio Transcription	26.78%	<b>61.80%</b>
SDPA (0.01 FPS) without Audio Transcription	87%	48.40%
SDPA (0.01 FPS) with Audio Transcription	68.92%	49.36%



**Figure 6:** Deliverable Attributes. Define each GenAI output file attributes. \*ALM stands for Audio Large Language model generated output, and \*VLM represents Video Large Language model generated output.

**Table 10:** Distribution of Property Audio and Video Data Across Medical Diseases Specialties.

Specialty	Audio	Video	Specialty	Audio	Video
Oncology	208	8934	Ophthalmology	159	2862
Cardiovascular	1	14	Respiratory Disease	16	467
Dermatology	0	30	Nephrology	1	380
Hematology	67	3606	Not Applicable	59	2853
Immunology	144	510	Movement Disorder	9	289
Infectious Disease	1	239	Inflammatory Disease	20	222
Metabolism	0	6	Neuroscience	202	4914

**Table 11:** The backbones, hyper-parameters, and prompt settings of **selective** SOTA VLMs.

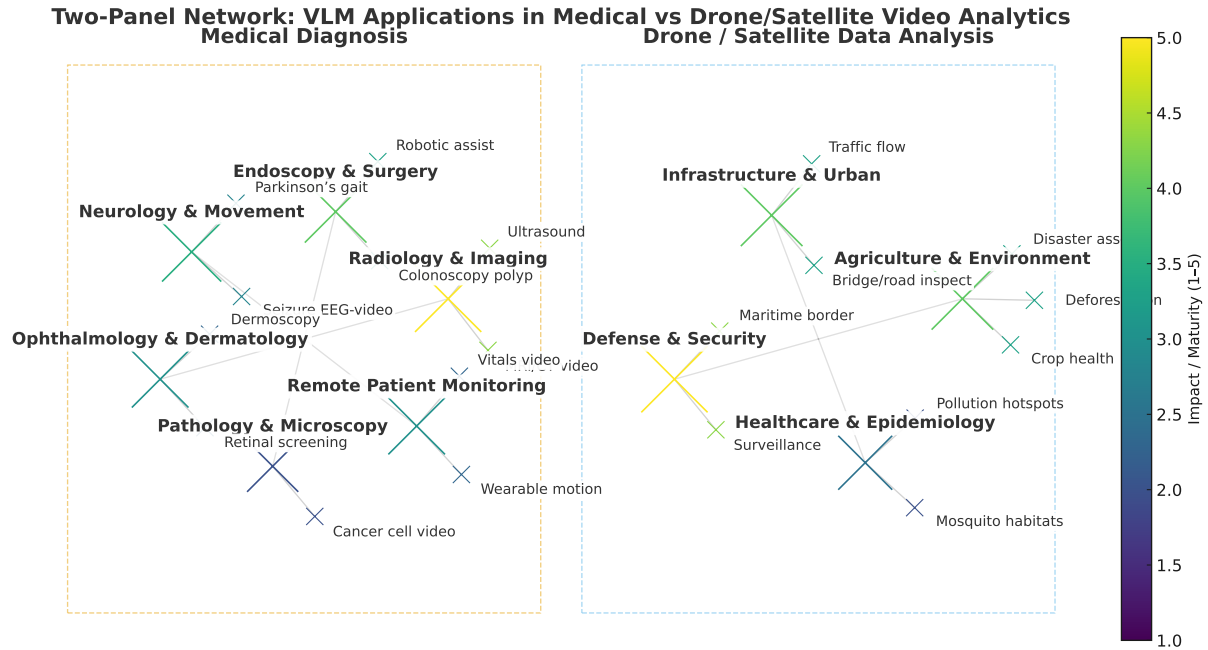
Model Description	Type	Token Limit	API Price in Dollars	Modality	Prompt Example
GPT-4 Turbo, The latest GPT-4 model with improved instruction, reproducible outputs, parallel function calling. Returns max of 4,096 output tokens. Training data up to Apr 2023	gpt-4-1106-preview	Input token limit:128K	Input 0.06/K Tokens. Output 0.12/K Tokens	Text Generation, Chat Completion, Image to Text	Could you please provide a summary of this video based on sample frames focusing on the content and workflow rather than specific logos or the color of text? After summarizing, list the key frames with brief captions in the format (00:00, caption). Ensure the analysis is accurate and avoid including any assumptions or extrapolations. Use an expert domain perspective to enhance relevance and precision. Do not repeat sentences or focus on QR codes or logos.
Qwen2-VL. updated on Huggingface Jan. 2025	Qwen2-VL-7B	Input token limit:32K	Opensource	Text Generation, Video to Text	Could you please provide a summary of this video, focusing on the content and workflow rather than specific logos or the color of text? After summarizing, list the key frames with brief captions in the format (00:00, caption). Ensure the analysis is accurate and avoid including any assumptions or extrapolations. Use an expert domain perspective to enhance relevance and precision. Do not repeat sentences or focus on QR codes or logos.
Gemini 2.0 Flash Model, released data 05th Feb 2025	Gemini 2.0 Flash 001	Input token limit:1048K Output token limit: 8K	Per 1M tokens in USD: API Cost 0.10 (text / image / video) 0.70 (audio). Output 0.40. e.g., 300 Long Videos from Video-MME costed 56.62	Audio, images, video, text, and PDF to text	Could you please provide a summary of this video, focusing on the content and workflow rather than specific logos or the color of text? After summarizing, list the key frames with brief captions in the format (00:00, caption). Ensure the analysis is accurate and avoid including any assumptions or extrapolations. Use an expert domain perspective to enhance relevance and precision. Do not repeat sentences or focus on QR codes or logos.

**Table 12:** Area of Impact 1: AI/VLM-driven Video Processing for Medical Diagnosis.

Category	Applications
<b>Radiology and Imaging</b>	<ul style="list-style-type: none"> <li>• <b>MRI/CT Scan Video Processing:</b> Advanced AI can analyze full-length MRI or CT scans in motion (e.g., cardiac MRI or functional MRI), detecting anomalies faster than manual review.</li> <li>• <b>Ultrasound Interpretation:</b> AI-powered real-time video analysis can help with fetal health assessments, echocardiography, and liver disease detection.</li> </ul>
<b>Endoscopy and Surgery</b>	<ul style="list-style-type: none"> <li>• <b>Colonoscopy Polyp Detection:</b> AI can process hours of colonoscopy footage to detect polyps in real-time, improving colorectal cancer screening.</li> <li>• <b>Robotic Surgery Assistance:</b> AI-driven video processing can provide real-time insights to surgeons, flagging anomalies or suggesting procedural adjustments.</li> </ul>
<b>Neurology and Movement Disorders</b>	<ul style="list-style-type: none"> <li>• <b>Seizure and Tremor Analysis:</b> AI can analyze EEG-video recordings to classify epilepsy types.</li> <li>• <b>Parkinson’s and ALS Monitoring:</b> AI can assess gait, facial expressions, and movement from patient videos for early diagnosis and tracking progression.</li> </ul>
<b>Ophthalmology and Dermatology</b>	<ul style="list-style-type: none"> <li>• <b>Retinal Scan Analysis:</b> AI models can process retinal scan videos to detect early diabetic retinopathy or macular degeneration.</li> <li>• <b>Skin Cancer Detection:</b> Dermatologists can use AI-enhanced dermoscopy video processing to detect melanoma more accurately.</li> </ul>
<b>Pathology and Microscopy</b>	<ul style="list-style-type: none"> <li>• AI can analyze continuous microscopy footage to identify cancerous cells, bacterial infections, or rare hematological conditions in blood samples faster than human pathologists.</li> </ul>
<b>Remote Patient Monitoring</b>	<ul style="list-style-type: none"> <li>• Wearable devices that record and process patient videos (e.g., heart rate monitors, movement trackers) can enable early diagnosis of conditions like heart arrhythmias or sleep apnea at lower costs.</li> </ul>

**Table 13:** Area of Impact 2: AI/VLM-driven Video Processing for Drone/Satellite Data Analysis.

Category	Applications
<b>Agriculture and Environment</b>	<ul style="list-style-type: none"> <li>• <b>Crop Health Monitoring:</b> AI-driven video analysis can quickly identify stressed crops, pest infestations, or nutrient deficiencies.</li> <li>• <b>Deforestation and Land Use:</b> Detecting illegal logging or monitoring ecosystem changes becomes faster and cheaper.</li> <li>• <b>Disaster Assessment:</b> Rapid damage assessment after hurricanes, earthquakes, or floods helps authorities respond effectively.</li> </ul>
<b>Infrastructure and Urban Planning</b>	<ul style="list-style-type: none"> <li>• <b>Road and Bridge Inspections:</b> AI can process high-resolution drone footage to detect cracks, erosion, or weak points.</li> <li>• <b>Traffic and Urban Planning:</b> Satellite video can track congestion patterns and optimize urban development.</li> </ul>
<b>Defense and Security</b>	<ul style="list-style-type: none"> <li>• <b>Surveillance and Threat Detection:</b> Automated analysis of drone/satellite feeds can detect anomalies, unauthorized activities, or suspicious movements.</li> <li>• <b>Border and Maritime Security:</b> Continuous video monitoring can identify smuggling, illegal crossings, or unauthorized vessel movements.</li> </ul>
<b>Healthcare and Epidemiology</b>	<ul style="list-style-type: none"> <li>• <b>Mosquito-Borne Disease Prevention:</b> Satellite video can help detect standing water bodies where mosquitoes breed, aiding in malaria/dengue prevention.</li> <li>• <b>Air Pollution and Public Health:</b> Fast video analysis can track pollution hotspots, correlating air quality data with disease outbreaks.</li> </ul>



**Figure 7: Two-Panel Network of Vision–Language Model (VLM) / AI Applications Across Medical and Drone/Satellite Video Analytics.** Node color and size encode an *impact/maturity score* (1–5). Impact is estimated as a composite index:  $\text{Impact} = 0.4R + 0.2D + 0.2A + 0.2C$ , where  $R$  denotes normalized research intensity (publications indexed in PubMed, IEEE Xplore, Scopus from 2020–2025 containing “video + AI or VLM + domain”),  $D$  dataset availability (standardized/public video datasets),  $A$  application readiness (evidence of clinical or industrial deployment and/or regulatory signals), and  $C$  cross-domain generalizability (extent of transferability to other domains or modalities). Scores are normalized to [1, 5] (5 = high maturity, well-validated, standardized datasets, active commercial use) and were assigned approximately as: Radiology & Imaging 5.0 with extensive literature (>200 papers, 2020–2025), benchmark datasets (e.g., MIMIC-CXR, cardiac MRI) (Li et al., 2021; Najjar, 2023; Pinto-Coelho, 2023), and multiple FDA-approved AI tools (FDA; ACR); Endoscopy & Surgery 4.0 with validated prototypes for polyp detection and robotic surgery support (Saeidi and et al., 2022), but moderate dataset availability (Borgli and et al., 2020; Ali et al., 2023; Ríos and et al., 2023); Neurology & Movement 3.5; Ophthalmology & Dermatology 3.0 with strong still-image AI base (fundus/OCT in ophthalmology; dermatology in dermatology), including FDA-cleared autonomous DR screening and large image datasets/challenges (FDA, 2018; Oshika and et al., 2024; Xu and et al., 2024; ISI; Yilmaz and et al., 2024; Behara and et al., 2024) but few real-time video pipelines available (Mahmoud and et al., 2024; Ghamsarian and et al., 2024; Dick and et al., 2024); Pathology & Microscopy 2.0 with only limited continuous video/time-lapse microscopy datasets and research mostly whole-slide image (WSI)–based (Fatima et al., 2024; Tan-Garcia et al., 2025); Remote Patient Monitoring 3.0 with active wearable and webcam-based monitoring studies and few open datasets (Haque et al., 2025; Huang et al., 2023); Agriculture & Environment 4.0 with large drone video datasets and real-world agricultural monitoring systems and high industrial uptake (Strzpek, 2023; Wang, 2025); Infrastructure & Urban 4.0 with drone/traffic inspection systems widely deployed and mature technical readiness (Du et al., 2018); Defense & Security 5.0 with advanced object tracking and anomaly detection systems in full industrial operation (Zhang et al., 2022); Healthcare & Epidemiology 2.5 with exploratory studies linking environmental video data with disease risk or pollution metrics (Wei et al., 2023). Smaller satellite nodes indicate representative sub-tasks within each category.

# PGGA: A Plan-Grounded GUI Agent for Automated Device Support

Lei Hsiung<sup>\*1,2</sup>, Zhiyu Chen<sup>2</sup>, Seonhoon Kim<sup>2</sup>, Qun Liu<sup>2</sup>

<sup>1</sup>Dartmouth College, <sup>2</sup>Amazon.com, Inc.

lei.hsiung.gr@dartmouth.edu, {zhiyu, seonhoon, qunliu}@amazon.com

## Abstract

Current GUI agents struggle with multi-step digital device support. We investigate whether this failure is partly caused by a procedural knowledge deficit: agents often rely on zero-shot visual exploration instead of executing verified instructions. To address this, we introduce the Plan-Grounded GUI Agent (PGGA), framing interface navigation as a knowledge-execution problem by conditioning low-level actions on step-by-step text plans. Evaluated on our focused Device-Support Interaction Benchmark (DSIB), results reveal a sharp gap between knowing which operation to perform and grounding that operation on the screen: GTA1-7B reaches 99.59% Operation Accuracy with expert plans, but only 82.99% Element Accuracy and 45.61% Task Success Rate; without plans, its Task Success Rate is 0.00%. Our fine-tuned 2B-parameter PGGA achieves 54.39% Task Success Rate and 91.28% Element Accuracy when guided by expert plans, suggesting that explicit procedural grounding can substantially improve GUI execution when high-quality plans are available. Project Page: <https://hsiung.cc/PGGA/>

## 1 Introduction

Autonomous Graphical User Interface (GUI) agents powered by Large Multimodal Models (LMMs) have demonstrated significant potential for automating digital workflows (Zhang et al., 2025; Nguyen et al., 2025; Hong et al., 2024). These agents aim to execute multi-step tasks by using the model’s reasoning and decision-making abilities to translate visual interface states into actionable commands. However, deploying these agents in real-world environments, particularly for complex device support, remains a persistent challenge. In this context, device support refers to the multi-step, highly specific procedures required to configure, maintain, or adapt digital devices, such

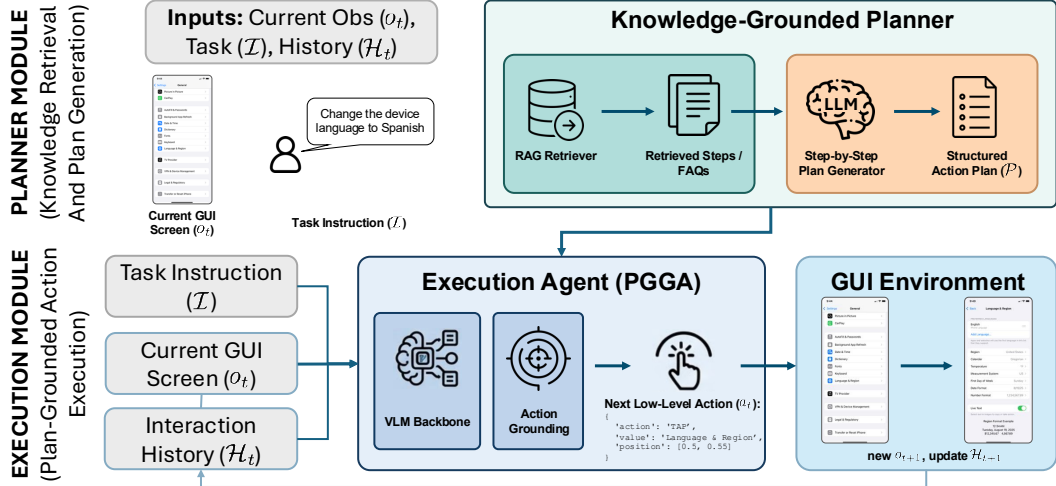
as navigating deep-nested system menus to alter localization inputs or managing granular security and privacy configurations.

We use *grounding* to mean the mapping from a symbolic instruction or plan step, such as “tap General”, to the concrete GUI element, operation, and screen coordinate required in the current visual state. This notion follows the broader symbol-and language-grounding literature (Harnad, 1990; Shridhar et al., 2020; Song et al., 2022) and recent GUI visual grounding work (Zheng et al., 2024; Gou et al., 2025), where successful action depends on connecting language to perceptual evidence rather than only generating plausible text.

A fundamental challenge in this domain is planning. While LMMs exhibit strong zero-shot reasoning capabilities, their efficacy in situated GUI navigation is often bottlenecked by the semantic gap between high-level user instructions and the low-level visual control required for execution. Existing approaches predominantly rely on zero-shot visual planning, implicitly expecting the model to infer complex, multi-step procedural logic purely from visual observations (Wu et al., 2025; Gou et al., 2025; Yan et al., 2023a). We argue that the frequent failure of GUI agents in these scenarios stems not merely from a deficit in visual grounding, but from an acute lack of domain-specific procedural knowledge. When confronted with unfamiliar interfaces, agents lacking structural priors default to inefficient exploration, leading to compounding errors and ultimate task failure.

This paper targets the domain of device support: tasks that are highly deterministic given a standard operating procedure but remain opaque to an unguided agent. To address this, we introduce the Plan-Grounded GUI Agent (PGGA), as illustrated in Figure 1. Instead of relying only on the Vision-Language Model’s (VLM) internal, implicit knowledge to guess the next action, PGGA grounds its execution in step-by-step action plans retrieved

<sup>\*</sup>Work done during an internship at Amazon.



**Figure 1: Plan-Grounded GUI Agent (PGGA) Framework.** Given a natural language task and the current visual state, an intermediate Planner module queries an external Knowledge Base to retrieve relevant documentation and synthesizes it into a step-by-step Action Plan. The model then predicts the next action by heavily conditioning on this explicit procedural plan, alongside the original task, current screenshot, and interaction history.

from an external knowledge base, following the retrieval-augmented generation paradigm (Lewis et al., 2020; Hayashi et al., 2025). Our central hypothesis is intentionally narrow: for deterministic device-support tasks, external procedural knowledge can reduce the search space enough to make visual grounding and action execution substantially more reliable.

To evaluate this device support capability, we curate the Device-Support Interaction Benchmark (DSIB). DSIB is a focused benchmark with 57 high-intent device configuration tasks and 241 individual steps collected directly on a mobile device, averaging 4.23 steps per task. The benchmark focuses on deep-nested system configurations and complex visual disambiguation, requiring navigation depths of up to five hierarchical menu levels across diverse functional domains. We therefore describe DSIB as a multi-step, deep-menu benchmark rather than claiming broad coverage of all long-horizon GUI automation.

We evaluate PGGA across varying degrees of plan quality: 1) Task Only, 2) Task with a Retrieved Action Plan, and 3) Task with an Expert-Annotated Action Plan. Relying solely on task instructions yields a 7.02% Task Success Rate (SR) for PGGA-2B. When grounded in expert-annotated plans, PGGA-2B achieves a 54.39% Task SR. This gap suggests that high-quality procedural priors can improve next-action prediction in complex digital environments, while the much lower score under retrieved plans highlights retrieval quality as an unresolved bottleneck. We summarize our main contributions as follows:

- We introduce the Device-Support Interaction Benchmark (DSIB), a focused dataset of complex, multi-step device support tasks designed to evaluate procedural execution and visual disambiguation in deep mobile settings menus.
- We propose the Plan-Grounded GUI Agent (PGGA), a framework that reformulates interface navigation from a zero-shot exploration problem into a knowledge-execution problem by conditioning low-level actions on step-by-step text plans.
- We empirically show that explicit procedural grounding substantially improves DSIB performance when plan quality is high, while also identifying retrieval noise, benchmark scale, and plan-specific fine-tuning as key limitations requiring further ablation.

## 2 Methodology

We model the environment as a Partially Observable Markov Decision Process (POMDP). At each timestep  $t$ , the agent receives an observation  $o_t \in \mathcal{O}$  (the current GUI screenshot) and must predict an action  $a_t \in \mathcal{A}$  based on a natural language task description  $\mathcal{I}$  and the interaction history  $h_t = \{(o_0, a_0), \dots, (o_{t-1}, a_{t-1})\}$ . Standard GUI agents learn a policy  $\pi(a_t | o_t, h_t, \mathcal{I})$ .

We hypothesize that learning this direct mapping is sub-optimal for complex device support, as it forces the agent to rely on ungrounded visual guessing when encountering unfamiliar interfaces. Inspired by the natural human cognitive workflow of consulting external manuals to

resolve device issues (detailed in Appendix B), the PGGA framework introduces an intermediate planning module. Given the task  $\mathcal{I}$  and current state  $o_t$ , the Planner queries an external Knowledge Base to retrieve a relevant document  $D$ . The Planner synthesizes  $D$  into a structured Action Plan  $P = \{p_1, p_2, \dots, p_n\}$ , where each  $p_i$  represents a discrete operational step. The PGGA policy is thus reformulated to condition heavily on this explicit plan:  $\pi_\theta(a_t|o_t, h_t, \mathcal{I}, P)$ .

## 2.1 Plan Construction and Retrieval

PGGA uses the same action space as the base GUI agent, but augments the input with a textual plan. The plan is not treated as an executable script: at every timestep, the model must still align the current screenshot and interaction history with the next relevant plan step, choose the correct operation, and ground it to the target UI element.

**Retrieved plans.** The default DSIB retrieval script first prompts the planner model to inspect the task and initial screenshot, producing a web-search query and a short description of the current GUI state. It then calls gpt-5.4 through the OpenAI Responses API with the web-search tool enabled, using deterministic decoding (temperature=0 and top\_p=1.0), and asks the model to synthesize a numbered, atomic action plan grounded in the current state.

**Expert-annotated plans.** For the expert-plan condition, we construct the plan from the ground-truth trajectory. The implementation first concatenates the previous low-level step representations with the current ground-truth step, then uses gpt-4.1-nano to rewrite these low-level traces into a concise numbered action plan. This condition is an oracle upper bound for plan quality rather than a deployable retrieval setting, and is used to measure whether the executor can ground and follow a correct procedure.

## 2.2 Model Architecture and Training

As illustrated in our system architecture, PGGA integrates the generated Action Plan alongside the visual state and action history. We use ShowUI-2B (Lin et al., 2025) as the base model because it is an open-weight, 2B-parameter vision-language-action model explicitly trained for GUI action prediction, making it a suitable backbone for testing whether plan conditioning can improve a compact executor.

The fine-tuning process utilizes Mind2Web (Deng et al., 2023). We convert each ground-truth interaction trace into a readable Action Plan using gpt-4.1-nano, the same lightweight trace-rewriting model used in the expert-plan construction script. This model is used only to rewrite already observed low-level actions into human-readable atomic steps; it is not used to infer missing procedures. The exact trace-to-plan prompt is provided in Appendix D. This training phase optimizes PGGA-2B as a plan-conditioned executor; it does not train the retrieval module.

## 2.3 Device-Support Interaction Benchmark (DSIB)

We curate the DSIB, building up 57 high-intent device support and configuration tasks, comprising a total of 241 individual steps, designed to mimic real-world mobile setup scenarios (Liu et al., 2025). All task trajectories within the dataset were collected directly on an Apple iPhone XR. The benchmark averages 4.23 steps per task and requires navigation depths of 3 to 5 hierarchical menu levels (see DSIB task example in Appendix C).

The dataset evaluates breadth and stability across six functional domains: Localization & Input (13.5%), System Customization (22.5%), Connectivity & Networking (17.5%), Security & Privacy (12.5%), Accessibility & Vision (19%), and Battery & Power Management (15%). Tasks frequently feature near-neighbor distractors (e.g., distinguishing “DD/MM/YYYY” from “DD-MM-YYYY”), demanding high-fidelity visual grounding. Instructions vary in semantic diversity, ranging from technical directives to colloquial user queries.

# 3 Experiments

## 3.1 Experimental Setup

**Models.** To evaluate the impact of plan grounding, we benchmark PGGA-2B against two representative baselines:

- **ShowUI-2B (Lin et al., 2025):** The model serving as our base architecture, evaluated zero-shot to demonstrate the delta achieved by our plan-grounding fine-tuning.
- **GTA1-7B (Yang et al., 2026):** A larger 7B-parameter VLM specialized for GUI tasks. We select GTA1-7B because it is an open GUI-focused model with strong reported test-time scaling performance, providing a high-capacity

Instruction Format	Models	Task SR.	Elem. Acc.	Op. Acc.	Step SR.
Task Only	ShowUI-2B	1.75%	46.47%	65.97%	35.27%
	GTA1-7B	0.00%	40.66%	<b>98.76%</b>	40.66%
	<b>PGGA-2B</b>	<b>7.02%</b>	<b>57.26%</b>	90.87%	<b>51.45%</b>
Task Only + <u>Action Plan</u> (Retrieval)	ShowUI-2B	3.51%	46.47%	70.95%	40.25%
	GTA1-7B	<b>29.82%</b>	65.15%	<b>97.10%</b>	65.15%
	<b>PGGA-2B</b>	19.30%	<b>76.76%</b>	87.55%	<b>68.05%</b>
Task + <u>Action Plan</u> (Expert-Annotated)	ShowUI-2B	3.51%	50.21%	88.38%	48.96%
	GTA1-7B	45.61%	82.99%	<b>99.59%</b>	82.57%
	<b>PGGA-2B</b>	<b>54.39%</b>	<b>91.28%</b>	95.02%	<b>88.38%</b>

**Table 1:** Performance comparison of GUI navigation agents on the DSIB dataset.

comparison point against the compact 2B executor.

We evaluate these models across three distinct instruction formats (see Appendix D for examples):

1. **Task Only** (zero-shot exploration): Only provide the task (e.g., Change the device language to Spanish),
2. **Task + Action Plan via Retrieval:** The action plan is constructed using the retrieval pipeline described in Appendix D, and
3. **Task + Expert-Annotated Action Plan** (the theoretical upper bound for plan quality): The action plan is constructed from all the ground-truth steps.

**Evaluation Metrics.** Agents are evaluated on four key axes to decouple reasoning from visual grounding:

- **Task Success Rate (Task SR):** A strict, binary metric indicating if the final target state was successfully reached and the correct ultimate action was executed.
- **Element Accuracy (Elem. Acc.):** The accuracy of correctly predicting the spatial coordinates or bounding box of the target UI element.
- **Operation Accuracy (Op. Acc.):** The accuracy of selecting the correct interaction type (e.g., TAP, TYPE, SCROLL) independent of element localization.
- **Step Success Rate (Step SR):** The percentage of individual steps within a task completed correctly, providing partial credit for trajectory progress.

### 3.2 Results

Table 1 presents the comparative performance of our fine-tuned PGGA-2B against the baselines across the three instruction formats.

**Zero-Shot Performance (Task Only).** Without procedural guidance, all models struggle to complete multi-step GUI navigation. PGGA-2B achieves the highest Task Success Rate (Task SR) at 7.02% and Step Success Rate (Step SR) at 51.45%. Notably, the GUI-specific GTA1-7B fails to complete any full trajectories in this setting (0.00% Task SR), highlighting the limitation of purely visual zero-shot exploration on DSIB.

**Retrieval-Augmented Performance.** Augmenting the instruction with a retrieved action plan yields noticeable improvements, but also exposes the retrieval bottleneck. GTA1-7B exhibits the strongest task-level performance in this setting, increasing to a 29.82% Task SR and outperforming PGGA-2B (19.30% Task SR). However, PGGA-2B maintains a higher Element Accuracy (76.76%) and Step SR (68.05%), suggesting it follows individual retrieved steps more accurately, even when the overall retrieved plan fails to reach the final target state.

**Expert-Annotated Performance.** Grounding the models in expert-annotated plans substantially improves execution. PGGA-2B achieves the best DSIB results in this setting, with a 54.39% Task SR, 91.28% Element Accuracy, and 88.38% Step SR. GTA1-7B also shows large gains (45.61% Task SR), indicating that high-quality action plans help both compact fine-tuned executors and larger GUI-specialized models bridge the procedural knowledge gap.

## 4 Discussion

**The Knowledge Bottleneck.** When models are restricted to the Task-Only format, performance collapses across all architectures. Notably, the GUI-specific GTA1-7B fails to complete any DSIB trajectory, achieving a 0.00% Task SR. However, when provided with an Expert-Annotated Action

Plan, PGGA-2B’s Task SR rises to 54.39%. This improvement suggests that, in deterministic device-support tasks, procedural knowledge is a major bottleneck: under expert guidance, PGGA-2B reaches 91.28% Element Accuracy, indicating that much of the remaining challenge is selecting and sequencing the right targets rather than merely recognizing visible widgets.

**Compact Plan-Conditioned Execution.** Our fine-tuned PGGA-2B consistently outperforms the much larger GTA1-7B model in Element Accuracy and Step SR when provided with structured plans. The behavior of the 7B-parameter baseline is revealing: while it maintains near-perfect Operation Accuracy (peaking at 99.59%, indicating strong semantic understanding of action types), its lower Element Accuracy (82.99%) compared to PGGA-2B suggests that scaling parameters alone does not guarantee precise pixel-level grounding in dense GUIs without task-format adaptation. We therefore frame PGGA as evidence for compact plan-conditioned execution, not as a pure parameter-efficiency result: PGGA-2B is fine-tuned for plan following, while GTA1-7B is evaluated without additional fine-tuning in our setup.

**The Retrieval Gap.** An important nuance in our findings is the performance degradation between Expert-Annotated plans and Retrieved plans. While PGGA-2B achieves a 54.39% Task SR with expert plans, it drops to 19.30% with retrieved plans. Interestingly, the larger GTA1-7B model demonstrates superior robustness to noisy or imperfect retrieved plans, achieving a 29.82% Task SR in this setting. This does not contradict the executor result; rather, it suggests that PGGA-2B is more literal and less recovery-oriented. It often grounds individual retrieved steps well, but a missing prerequisite, wrong step order, or irrelevant retrieved instruction can derail the full trajectory. GTA1-7B appears better able to ignore or repair some noisy plan fragments, while PGGA-2B more faithfully follows them.

## 5 Conclusion

This paper introduces the Plan-Grounded GUI Agent (PGGA), a framework that reframes automated device support from a problem of zero-shot visual exploration to one of grounded procedural execution. By formalizing the integration of retrieved action plans into the VLM observation space, we demonstrate that a parameter-efficient

2B model can achieve profound improvements in visual grounding and task success. Our findings isolate long-horizon planning and domain knowledge as the current fundamental bottlenecks in GUI agent capabilities. While PGGA establishes a new standard for plan execution, our analysis of the *retrieval gap* underscores the necessity for future research focused on improving the robustness of the knowledge retrieval mechanisms and error-recovery policies.

## Limitations

PGGA demonstrates that plan grounding can substantially improve DSIB execution when high-quality procedures are available, but several limitations remain. First, DSIB is intentionally focused: it contains 57 tasks, 241 steps, and one device configuration environment (Apple iPhone XR). This makes it useful as a controlled stress test for deep settings navigation, but insufficient for broad claims about all device-support tasks or GUI automation. We therefore treat the current results as pilot-benchmark evidence. Future work should also evaluate multiple devices, iOS versions, Android settings flows, and interactive re-planning methods that can detect when a retrieved plan is inconsistent with the observed screen. Second, the gap between expert-annotated and retrieved plans is large. Our 2B-parameter model is precise when instructions are correct, but less robust to noisy or suboptimal retrieved plans than GTA1-7B. This indicates that retrieval quality is not an implementation detail but a central part of the system. Third, our comparison is not a pure parameter-efficiency comparison because PGGA-2B is fine-tuned for plan following while GTA1-7B is evaluated without additional fine-tuning.

## References

- Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. 2024. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *arXiv preprint arXiv:2406.11896*.
- Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. *Mind2web: Towards a generalist agent for the web*. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su.

2025. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *The Thirteenth International Conference on Learning Representations*.
- Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.
- Hiroaki Hayashi, Bo Pang, Wenting Zhao, Ye Liu, Akash Gokul, Srijan Bansal, Caiming Xiong, Semih Yavuz, and Yingbo Zhou. 2025. Self-abstraction from grounded experience for plan-guided policy refinement. *arXiv preprint arXiv:2511.05931*.
- Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and 1 others. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14281–14290.
- Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipalli, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. 2024. Llm2llm: Boosting llms with novel iterative data enhancement. *arXiv preprint arXiv:2403.15042*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Zhangheng Li, Keen You, Haotian Zhang, Di Feng, Harsh Agrawal, Xiujun Li, Mohana Prasad Sathya Moorthy, Jeff Nichols, Yinfei Yang, and Zhe Gan. 2024. Ferret-ui 2: Mastering universal user interface understanding across platforms. *arXiv preprint arXiv:2410.18967*.
- Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. 2025. Showui: One vision-language-action model for gui visual agent. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19498–19508.
- Guangyi Liu, Pengxiang Zhao, Yaozhen Liang, Liang Liu, Yaxuan Guo, Han Xiao, Weifeng Lin, Yuxiang Chai, Yue Han, Shuai Ren, Hao Wang, Xiaoyu Liang, WenHao Wang, Tianze Wu, Zhengxi Lu, Siheng Chen, LiLinghao, Hao Wang, Guanqing Xiong, and 2 others. 2025. LLM-powered GUI agents in phone automation: Surveying progress and prospects. *Transactions on Machine Learning Research*.
- Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, and 1 others. 2025. Gui agents: A survey. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22522–22538.
- Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. 2023. Adapt: As-needed decomposition and planning with language models. *arXiv preprint arXiv:2311.05772*.
- Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Xinyue Yang, Jiadai Sun, Yu Yang, Shuntian Yao, Tianjie Zhang, and 1 others. 2024. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. *arXiv preprint arXiv:2411.02337*.
- Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2024. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *CoRR*, abs/2302.04761.
- Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10737–10746. Computer Vision Foundation / IEEE.
- Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M. Sadler, Wei-Lun Chao, and Yu Su. 2022. Llm-planner: Few-shot grounded planning for embodied agents with large language models. *CoRR*, abs/2212.04088.
- Hang Wu, Hongkai Chen, Yujun Cai, Chang Liu, Qingwen Ye, Ming-Hsuan Yang, and Yiwei Wang. 2025. Dimo-gui: Advancing test-time scaling in gui grounding via modality-aware visual reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26257–26267.
- An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, and 1 others. 2023a. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. *arXiv preprint arXiv:2311.07562*.
- An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, and 1 others. 2023b. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. *arXiv preprint arXiv:2311.07562*.
- Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *Preprint*, arXiv:2310.11441.

- Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. 2024. Agentoccam: A simple yet strong baseline for llm-based web agents. *arXiv preprint arXiv:2410.13825*.
- Yan Yang, Dongxu Li, Yutong Dai, Yuhao Yang, Ziyang Luo, Zirui Zhao, Zhiyuan Hu, JUNZHE HUANG, Amrita Saha, Zeyuan Chen, Ran Xu, Liyuan Pan, Caiming Xiong, and Junnan Li. 2026. [GTA1: GUI test-time scaling agent](#). In *The Fourteenth International Conference on Learning Representations*.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.
- Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2025. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20.
- Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. 2024. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. *arXiv preprint arXiv:2408.15978*.
- Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. [Gpt-4v\(ision\) is a generalist web agent, if grounded](#). In *Forty-first International Conference on Machine Learning*.

## Appendix

### A Related Work

**Graphical User Interface (GUI) Agents.** Recent advancements in Large Multimodal Models (LMMs) have catalyzed the development of autonomous agents capable of navigating digital environments (Zhang et al., 2025; Zheng et al., 2024; Hong et al., 2024). These systems generally fall into two paradigms: training-free and training-based models. Training-free approaches rely on proprietary models like GPT-4 to process external UI representations, such as accessibility trees or OCR-derived bounding boxes, via zero-shot prompting (Yang et al., 2023; Yan et al., 2023b). Conversely, training-based models fine-tune open-weight architectures on massive datasets of web and mobile interactions to directly predict actions from raw pixels (Hong et al., 2024; Li et al., 2024). While these models have significantly closed the visual perception gap, they predominantly focus on generalized navigation. In contrast, PGGa explicitly targets the domain of automated device support. We recognize that visual acuity alone is insufficient; without the domain-specific procedural knowledge required for complex system configurations, models default to inefficient and error-prone guessing.

**Planning and Reasoning in Autonomous Agents.** To tackle long-horizon tasks, prior work has extensively explored hierarchical planning and tool use (Yao et al., 2023; Schick et al., 2023). In the digital agent domain, systems like WebPilot (Zhang et al., 2024), AgentOccam (Yang et al., 2024), and ADaPT (Prasad et al., 2023) employ complex, multi-agent prompting frameworks using massive closed-source models to iteratively decompose tasks and replan upon failure. While effective in unconstrained web browsing, these zero-shot exploratory methods are highly inefficient—and potentially unsafe—for deterministic device support. Operating deep within system settings demands a high degree of operational robustness to avoid unintended configurations. PGGa addresses this by replacing open-ended exploration with authoritative plan execution. By grounding a parameter-efficient 2B model in retrieved procedural steps, we ensure a more reliable, trustworthy, and computationally efficient trajectory than complex multi-agent prompting cascades.

**Grounded Language Understanding and Synthetic Data.** Mapping high-level natural lan-

guage instructions to executable environment actions is a core challenge in grounded language understanding (Shridhar et al., 2020; Song et al., 2022). For digital agents, datasets like Mind2Web (Deng et al., 2023) and AndroidInTheWild (Rawles et al., 2024) have provided critical interaction traces. Recently, synthetic data generation has been widely adopted to augment these datasets, utilizing powerful LLMs to annotate trajectories, filter failures, or synthesize new task queries (Qi et al., 2024; Bai et al., 2024; Lee et al., 2024). Our methodology builds upon this paradigm but shifts the fundamental focus from purely visual-action mapping to procedural conditioning. We utilize LLMs to programmatically convert ground-truth interaction traces into structured step-by-step action plans, enabling our model to learn explicit plan-following behaviors. This data synthesis approach facilitates our parameter-efficient fine-tuning and directly motivates the creation of our Device-Support Interaction Benchmark (DSIB) to rigorously evaluate these specific capabilities.

### B Motivation from Human Problem-Solving

To design a more robust autonomous agent, we draw direct inspiration from how human users naturally resolve complex device support tasks (Figure 2). When a user encounters an unfamiliar interface or does not intuitively know how to configure a specific setting, they rarely resort to blind, trial-and-error clicking. Instead, they implicitly acknowledge their knowledge deficit and consult external authoritative sources—such as searching online tutorials or reading user manuals. Once this procedural knowledge is acquired, the user systematically follows the suggested step-by-step instructions, directly translating text-based guidance into visual actions until the problem is solved. This human baseline is highly efficient and minimizes the risk of altering unintended system configurations.

In stark contrast, current state-of-the-art GUI agents handle knowledge deficits fundamentally differently. As illustrated in the lower trajectory of Figure 2, when an agent lacks the requisite domain knowledge to navigate a deep-nested menu, it relies heavily on zero-shot visual guessing. Rather than retrieving a verified plan, the agent hallucinates the next logical step based solely on the immediately visible UI elements. While this ungrounded exploration might occasionally result in the cor-

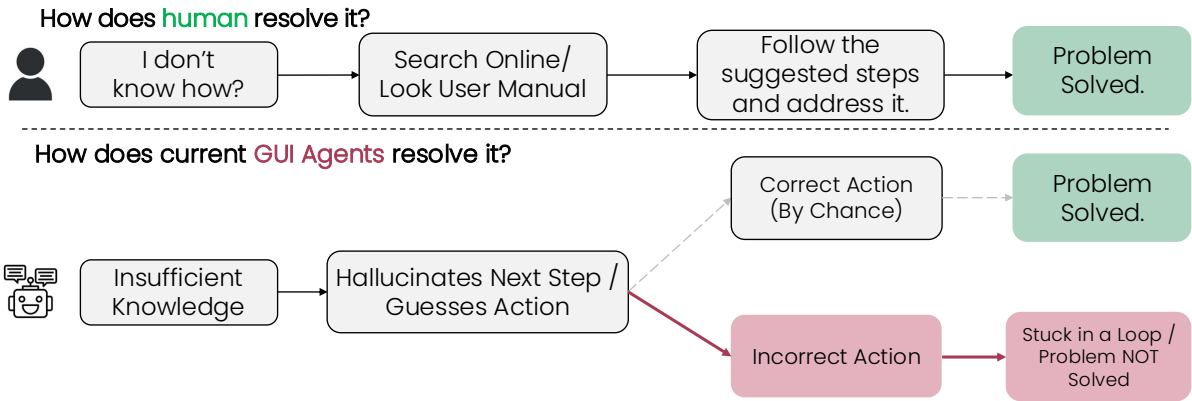


Figure 2: Task Resolving Trajectory

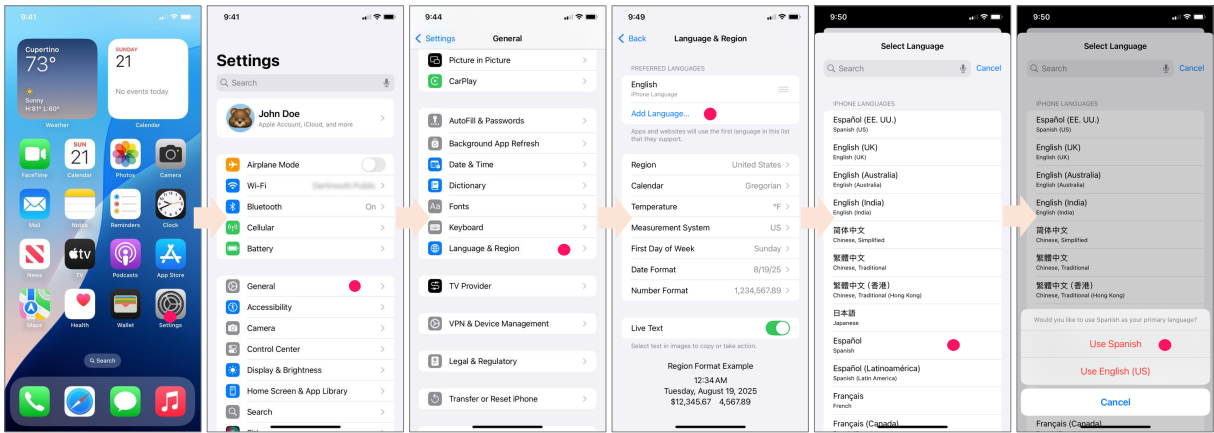


Figure 3: Example DSIB trajectory for a Localization & Input task. The visual sequence illustrates the necessary steps to change the device language to Spanish.

rect action by sheer chance, it predominantly leads to incorrect operations. In complex device support scenarios, a single incorrect action can trap the agent in an unrecoverable state, causing it to become stuck in a loop and ultimately fail the task.

This dichotomy reveals that the primary failure mode of current agents is not a lack of visual perception, but an architectural absence of procedural grounding. PGGA is directly motivated by this human-centric problem-solving trajectory. By equipping the agent with an intermediate planner to retrieve and strictly follow verified action plans, we shift the paradigm from brittle, hallucination-prone guessing to robust, knowledge-grounded execution.

### C DSIB Task Example

To illustrate the complexity and depth required by the DSIB dataset, Figure 3 provides a concrete example of a “Localization & Input” task: changing the primary device language to Spanish. This sequence demonstrates the precise hierarchical navigation and visual disambiguation challenges agents face.

### D Task-format Examples, Plan Generation and Retrieval Prompts

#### Task-Only

You are an assistant trained to navigate the iPhone. Given a task instruction, a screen observation, and an action history sequence, output the next action and wait for the next observation.

Here is the action space:

1. ‘TAP’: Tap on an element, value is the element to tap and the position [x,y] is required.
2. ‘TYPE’: Type a string into an element, value is the string to type and the position [x,y] is required.
3. ‘SCROLL’: Scroll the screen, value is the direction to scroll and the position [x,y] is not applicable.

Format the action as a dictionary with the following keys:

```
{‘action’: ‘ACTION_TYPE’, ‘value’: ‘element’, ‘position’: [x,y]}
```

Position represents the relative coordinates on the screenshot and should be scaled to a range of 0-1. Only respond in dictionary format.

Task: Change the device language to Spanish  
<IMAGE - Screenshot Observation>

What is the next action? (Response should only contain the json dictionary.)

### Task + Action Plan (Retrieval)

You are an assistant trained to navigate the iPhone. Given a task instruction, action plan, a screen observation, and an action history sequence, output the next action and wait for the next observation.

Here is the action space:

1. 'TAP': Tap on an element, value is the element to tap and the position [x,y] is required.
2. 'TYPE': Type a string into an element, value is the string to type and the position [x,y] is required.
3. 'SCROLL': Scroll the screen, value is the direction to scroll and the position [x,y] is not applicable.

Format the action as a dictionary with the following keys:

```
{'action': 'ACTION_TYPE', 'value': 'element', 'position': [x,y]}
```

Position represents the relative coordinates on the screenshot and should be scaled to a range of 0-1. Only respond in dictionary format.

Task: Change the device language to Spanish  
Action Plan: 1. Tap Settings on the iPhone Home Screen.  
2. Scroll if needed and tap General.  
3. Tap Language & Region.  
4. Tap Add Language... if Spanish is not already listed, or tap iPhone Language if that option is visible.  
5. In the language list or search field, select Spanish.  
6. If multiple Spanish variants appear, tap the desired option, such as Español (España) or Español (Latinoamérica).  
7. Tap Use Spanish or confirm the change when prompted.  
8. Wait for the iPhone interface to refresh and display in Spanish.

<IMAGE - Screenshot Observation>

What is the next action? (Response should only contain the json dictionary.)

### Mind2Web Trace-to-Plan Prompt

System: Convert a demonstrated GUI trajectory into a user-readable action plan.

Input:

Task: {Mind2Web task instruction}

Trajectory: {ordered ground-truth actions and target element descriptions}

Instructions:

1. Convert the trajectory into atomic, imperative steps.
2. Keep the order exactly aligned with the demonstrated trajectory.
3. Use visible element names when available.
4. Do not add steps that are not present in the trajectory.
5. Do not include coordinates or implementation details.

Output: A numbered action plan only.

### Task + Action Plan (Expert-Annotated)

You are an assistant trained to navigate the iPhone. Given a task instruction, action plan, a screen observation, and an action history sequence, output the next action and wait for the next observation.

Here is the action space:

1. 'TAP': Tap on an element, value is the element to tap and the position [x,y] is required.
2. 'TYPE': Type a string into an element, value is the string to type and the position [x,y] is required.
3. 'SCROLL': Scroll the screen, value is the direction to scroll and the position [x,y] is not applicable.

Format the action as a dictionary with the following keys:

```
{'action': 'ACTION_TYPE', 'value': 'element', 'position': [x,y]}
```

Position represents the relative coordinates on the screenshot and should be scaled to a range of 0-1. Only respond in dictionary format.

Task: Change the device language to Spanish  
Action Plan: 1. Tap Settings  
2. Tap General  
3. Tap Language & Region  
4. Tap Add Language ...  
5. Tap Español  
6. Tap Use Spanish

<IMAGE - Screenshot Observation>

What is the next action? (Response should only contain the json dictionary.)

### Retrieved-Plan Synthesis Prompt

System: You are a device-support planning assistant. Generate a concise, ordered action plan for an iPhone settings task using only the retrieved support passages.

Input:

Task: {natural language user task}

Retrieved passages: {top-k retrieved passages with source identifiers}

Instructions:

1. Output numbered steps that can be executed on the device.
2. Each step should name one visible UI target or one operation.
3. Preserve prerequisite order.
4. If the passages do not support a required step, write "uncertain" for that step rather than inventing details.
5. Avoid redundant back-and-forth navigation unless it is required by the task.

Output: A numbered action plan only.

# CAFES: A Collaborative Multi-Agent Framework for Multi-Granular Multimodal Essay Scoring

Jiamin Su<sup>1</sup>, Yibo Yan<sup>1,2</sup>, Zhuoran Gao<sup>1</sup>,  
Han Zhang<sup>1</sup>, Xiang Liu<sup>1,2</sup>, Huiyu Zhou<sup>3</sup>, Xuming Hu<sup>1,2\*</sup>

<sup>1</sup>The Hong Kong University of Science and Technology (Guangzhou)

<sup>2</sup>The Hong Kong University of Science and Technology

<sup>3</sup>Guangxi Zhuang Autonomous Region Big Data Research Institute

j-su360@connect.hkust-gz.edu.cn, xuminghu@hkust-gz.edu.cn

## Abstract

Automated Essay Scoring (AES) is crucial for modern education, particularly with the increasing prevalence of multimodal assessments. However, traditional AES methods struggle with *evaluation generalizability and multimodal perception*, while even recent Multimodal Large Language Model (MLLM)-based approaches can produce *hallucinated justifications and scores misaligned with human judgment*. To address the limitations, we introduce **CAFES**, the first collaborative multi-agent framework specifically designed for AES. It orchestrates three specialized agents: an *Initial Scorer* for rapid, trait-specific evaluations; a *Feedback Pool Manager* to aggregate detailed and evidence-grounded feedback; and a *Reflective Scorer* that iteratively refines scores based on this feedback to enhance human alignment. Extensive experiments, using widely adopted MLLMs, achieve an average relative improvement of 21% in Quadratic Weighted Kappa (QWK) against ground truth, with particularly strong gains in grammatical and lexical diversity. Our proposed CAFES paves the way for an intelligent multimodal AES system. The code and dataset are available at <https://anonymous.4open.science/r/CAFES-C87F/>.

## 1 Introduction

Automated Essay Scoring (AES) plays a crucial role in educational assessment today, offering efficient, fair, and scalable evaluation of student writing tasks (Ramesh and Sanampudi, 2022; Li and Liu, 2024; Wu et al., 2024; Xia et al., 2024). AES systems benefit both students by highlighting key areas for improvement and providing actionable feedback, and educators by reducing manual grading workloads. As contemporary assessments increasingly emphasize students’ abilities to integrate information from both text and images, multimodal

\*Corresponding author.

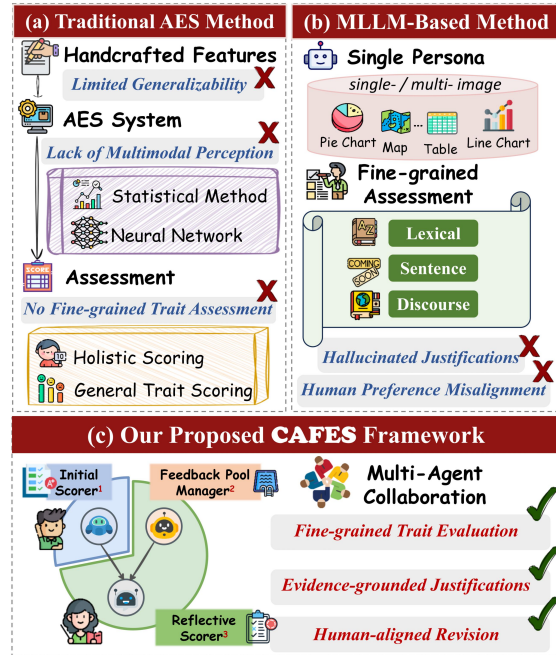


Figure 1: Comparisons among the traditional AES method (a), MLLM-based method (b), and our proposed multi-agent CAFES framework (c) on AES task.

writing tasks have become a key focus. Therefore, there is a rapidly growing need for AES systems capable of precise, detailed, context-aware evaluations that effectively handle multimodal inputs (Ye et al., 2025; Su et al., 2025; Li et al., 2024).

**Traditional AES methods** suffer from several limitations, as shown in Figure 1 (a). ❶ They rely on handcrafted features like word frequency and essay length, limiting their generalizability across diverse topics (Yang et al., 2024; Uto et al., 2020). ❷ They lack multimodal perception, making them unsuitable for multimodal inputs. ❸ They struggle to assess fine-grained traits, such as coherence and organizational structure (Wang et al., 2022). Recently, Multimodal Large Language Models (MLLMs) have been applied to AES, yet MLLM-based methods still introduce challenges like ❹ hallucinated justifications and ❺ scoring misaligned

with human preference (Su et al., 2025; Do et al., 2024), as shown in Figure 1 (b).

However, the emergence of multimodal multi-agent systems offers a compelling and promising solution to these challenges (Chu et al., 2025). Specifically, multi-agent frameworks demonstrate the following key advantages, as shown in Figure 1 (c): ✓ They enable fine-grained trait evaluation, providing comprehensive feedback across various writing traits. ✓ They can generate evidence-grounded justifications and engage in cross-agent collaboration, effectively mitigating hallucinations introduced by a single MLLM. ✓ The reflective mechanism of multi-agent systems ensures human-aligned revisions, adjusting scores to better align with human preferences.

Therefore, we propose CAFES, the first-ever collaborative multi-agent framework designed specifically for AES. In particular, CAFES decomposes the scoring process into three specialized modules: an *initial scoring agent* that provides fast trait-specific scores; a *feedback pool agent* that aggregates detailed strengths across writing traits; and a *reflective scoring agent* that iteratively updates scores based on the feedback pool. In summary, our contributions lie in three aspects:

- We introduce CAFES, the first multi-agent framework for AES, integrating three specialized agents including Initial scorer, Feedback Pool Manager, and Reflective Scorer to enable collaborative multi-granular essay scoring.
- We demonstrate the essential impact of the Feedback Pool Manager, Reflective Scorer, and teacher-student MLLM collaboration mechanism through ablation studies.
- We evaluate the CAFES framework with eight widely adopted MLLMs as student models, GPT-4o as the default teacher model, achieving an average improvement of 21% in Quadratic Weighted Kappa (QWK), especially for grammatical and lexical diversity.

By addressing the gaps in the existing AES approaches, CAFES paves the way for reliable, nuanced, and context-sensitive multi-agent AES systems driven by MLLMs in the AGI era.

## 2 Related Work

### 2.1 AES Datasets

Existing AES datasets have been widely used to support research on writing assessment (more details are shown in the Appendix A.1). In terms of modality, these datasets can be categorized into text-only and multimodal datasets.

Among **text-only datasets**, ASAP<sub>AES</sub> (Cozma et al., 2018) is widely used due to its large scale and high quality. Its extended version, ASAP++, adds trait-level annotations, but merges key content traits into a single “CONTENT” label (Mathias and Bhattacharyya, 2018). Both of them only have few topics in total. The CLC-FCE dataset provides detailed annotations of grammatical errors (Yannakoudakis et al., 2011). The TOEFL11 dataset uses only coarse-grained proficiency labels (low / medium / high) (Lee et al., 2024a). The ICLE (Granger et al., 2009) and ICLE++ (Li and Ng, 2024c) datasets offer more detailed and multi-granular annotations. Nevertheless, their topic coverage is highly limited. Similarly, the AAE corpus focuses solely on argumentative structure (Stab and Gurevych, 2014). The CREE corpus is designed to evaluate sentence understanding and error types (Bailey and Meurers, 2008). In summary, existing text-only AES datasets generally suffer from two key limitations: (1) limited topic diversity, and (2) a lack of fine-grained trait-level annotations (Ke and Ng, 2019; Li and Ng, 2024b,a).

EssayJudge is **the only publicly available multimodal AES dataset** (Su et al., 2025), providing ten fine-grained trait annotations for comprehensive assessment, organized into three levels. Lexical-level traits include *lexical accuracy* (LA) and *lexical diversity* (LD). Sentence-level traits include *coherence* (CH), *grammatical accuracy* (GA), *grammatical diversity* (GD), and *punctuation accuracy* (PA). Discourse-level traits include *argument clarity* (AC), *justifying persuasiveness* (JP), *organizational structure* (OS), and *essay length* (EL).

### 2.2 AES Systems

AES systems are typically classified into three types: heuristic, machine learning, and deep learning approaches (Li and Ng, 2024a; Kamalov et al., 2025; Atkinson and Palma, 2025; Xu et al., 2025; Song et al., 2025). **Heuristic methods** assign overall scores by combining rule-based trait scores such as organization, coherence, and grammar. For instance, the organization can be assessed using

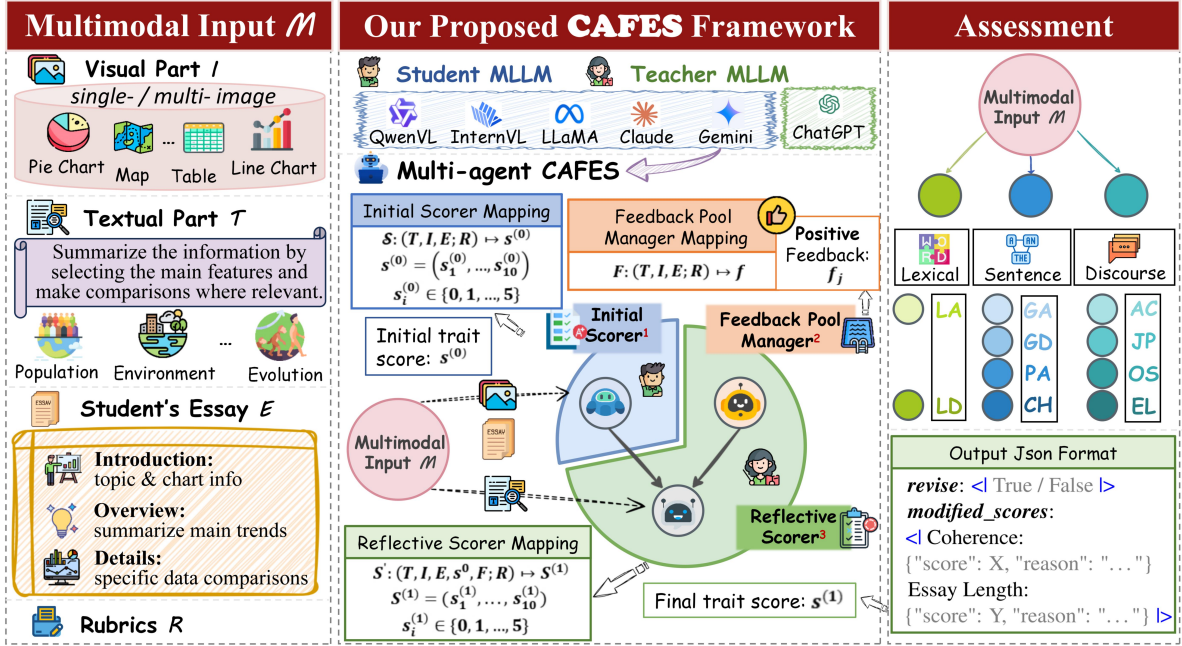


Figure 2: The framework of our proposed CAFES. The system follows a three-stage process: ❶ Initial scoring via the student MLLM; ❷ Feedback generation for each trait via the teacher MLLM; and ❸ Final reflective scoring with justification-based revision via the teacher MLLM.

templates like the three-paragraph essay format (Attali and Burstein, 2006). **Machine learning methods** (e.g., Logistic Regression, SVMs) rely on handcrafted features (Chen and He, 2013; Yannakoudakis and Briscoe, 2012), including length-based attributes (Vajjala, 2016; Yannakoudakis and Briscoe, 2012). Thus, their cross-topic generalization is limited. **Deep learning methods**, especially Transformer models like BERT (Wang et al., 2022), improve AES by learning semantic features directly from text (Jiang et al., 2023; Cao et al., 2020; Wang and Liu, 2025; Li and Pan, 2025), enabling multi-trait and cross-topic scoring. **LLM-based methods** have recently advanced AES (Mizumoto and Eguchi, 2023; Choi et al., 2025; Cai et al., 2025; Do et al., 2024). They support zero-shot scoring using rubrics alone (Lee et al., 2024a), or few-shot settings with minimal examples (Mansour et al., 2024; Xiao et al., 2024), offering better performance and adaptability in low-resource scenarios.

### 2.3 Multi-Agent Collaboration

Recent studies suggest that multi-agent collaboration, by organizing and coordinating multiple LLMs, enables more effective handling of complex tasks (Tran et al., 2025; Yan et al., 2025b). Systems like CAMEL (Li et al., 2023) and AutoGen (Wu et al., 2023) assign roles such as planner, coder, and critic, allowing agents to interact through multi-

turn dialogue and perform better in reasoning, generation, and self-revision (Liang et al., 2024). This approach offers key benefits: improved task decomposition and control through role division, reduced bias via mutual verification, and enhanced adaptability and modularity. It is increasingly adopted in areas such as decision-making (Liu et al., 2024b), code generation (Yuan et al., 2024), and automated evaluation (Lifshitz et al., 2025). More related work on MLLMs can be found in Appendix A.2.

### 3 Methodology

Figure 2 illustrates the overall architecture of our multi-agent AES framework, which consists of **three core agents**: ❶ Initial Scorer, ❷ Feedback Pool Manager, and ❸ Reflective Scorer. To execute these agents, we introduce **two types of models**: ❶ a student MLLM and ❷ a teacher MLLM. The student model, which has relatively weaker capabilities, handles the Initial Scorer by giving an initial score for each of the ten fine-grained traits. The teacher model, with stronger **reasoning abilities**, executes the Feedback Pool Manager to generate feedback comments and applies the Reflective Scorer to revise the student model’s initial scores based on the feedback pool. This collaborative setup mirrors the human-in-the-loop process of "scoring → feedback → revision" in real-world scoring assessment. In the following sections, we

provide a detailed description of each module.

### 3.1 Initial Scorer

The Initial Scorer module is responsible for producing preliminary scores across the ten fine-grained traits. Given the text of the essay topic  $T$ , the corresponding image  $I$ , the student’s essay  $E$ , and the detailed scoring rubrics  $\mathbf{R} \in \mathbb{R}^{10}$ , the student MLLM assigns an initial score  $s_i^{(0)}$  for each trait  $d_i$ . Formally, the Initial Scorer defines a mapping:

$$\mathcal{S} : (T, I, E; \mathbf{R}) \mapsto \mathbf{s}^{(0)} \in \mathbb{R}^{10}$$

where  $\mathbf{s}^{(0)} = (s_1^{(0)}, \dots, s_{10}^{(0)})$  denotes the preliminary scores with  $s_i^{(0)} \in \{0, 1, \dots, 5\}$ .

This step can be viewed as the student model independently answering an exam based on its own understanding. The subsequent modules, executed by the teacher MLLM, are responsible for reviewing the student MLLMs’ answers, providing feedback, and refining the initial judgments.

### 3.2 Feedback Pool Manager

The Feedback Pool Manager module is responsible for generating feedback for the student’s essay based on the ten traits. Prior studies have indicated that MLLMs tend to adhere to the rubrics more strictly than human raters, often assigning lower scores during essay scoring (Su et al., 2025; Kundu and Barbosa, 2024). To address this tendency, we design the Feedback Pool Manager to focus exclusively on extracting feedback, emphasizing the strengths demonstrated in the essay. Formally, the Feedback Pool Manager defines a mapping:

$$\mathcal{F} : (T, I, E; \mathbf{R}) \mapsto \mathbf{f} \in \mathbb{R}^{10}$$

where  $\mathbf{f} = (f_1, \dots, f_{10})$  denotes a set of feedback entries, each associated with a specific trait  $d_i$ . For each trait, MLLMs return the extracted comments highlighting well-performed aspects of the essay.

The feedback generated by the teacher MLLM provides crucial and structured guidance, enabling the Reflective Scorer to determine whether the initially assigned scores require further revision.

### 3.3 Reflective Scorer

The Reflective Scorer module is responsible for revising the student’s initial scores by integrating the feedback information. Formally, the Reflective Scorer defines a mapping:

$$\mathcal{S}' : (T, I, E, \mathbf{s}^{(0)}, \mathbf{f}; \mathbf{R}) \mapsto \mathbf{s}^{(1)} \in \mathbb{R}^{10}$$

where  $\mathbf{s}^{(1)} = (s_1^{(1)}, \dots, s_{10}^{(1)})$  denotes the revised scores. The teacher MLLM outputs a revised JSON object, and an example is shown in Figure 3.

This reflective revision mechanism ensures that the final assessment fairly incorporates the strengths recognized in the essay, while avoiding unnecessary or overly aggressive adjustments.

```

revise: <| True / False |>
modified_scores:
<| Coherence: {"score": X, "reason": "..."}
Essay Length: {"score": Y, "reason": "..."} |>

```

Figure 3: Reflective scorer’s JSON output format.

## 4 Experiments and Analysis

### 4.1 Experimental Setup

**Dataset.** We evaluate our agent-based AES system CAFES on the ESSAYJUDGE dataset, the only publicly available multimodal AES dataset. It consists of 1,054 multimodal essays written at the university level. Each essay requires students to analyze and construct arguments based on visual inputs such as line charts and flow charts, posing significant challenges for MLLMs in terms of visual-textual understanding and reasoning. What’s more, it covers 125 distinct essay topics across diverse domains including education, environment, and evolution. More details about the dataset are shown in Table 1. The diversity in both topics and visual formats increases the complexity of the scoring task and provides a strong foundation for evaluating the robustness and generalizability of AES systems under varied multimodal scenarios.

Statistic	Number
<b>Total Multimodal Essays</b>	1,054
<b>Image Type</b>	
- Single-Image	703 (66.7%)
- Multi-Image	351 (33.3%)
<b>Multimodal Essay Type</b>	
- Flow Chart	305 (28.9%)
- Bar Chart	211 (20.0%)
- Table	153 (14.5%)
- Line Chart	145 (13.8%)
- Pie Chart	71 (6.7%)
- Map	62 (5.9%)
- Composite Chart	107 (10.2%)

Table 1: Key statistics of the ESSAYJUDGE dataset.

**Basic Settings.** In the CAFES AES framework, **GPT-4o is assigned as the default teacher model**

throughout all experiments to guide and refine student MLLM’s outputs, given its strong performance in AES (Hurst et al., 2024; Su et al., 2025). To evaluate CAFES’ generalization ability, we systematically assign a wide range of leading MLLMs to the student model, grouped as follows: (i) **Open-source MLLMs**: InternVL2.5 (2B, 4B, 8B, 26B) (Chen et al., 2025b), Qwen2.5-VL (3B, 32B) (Chen et al., 2025c), and LLaMA-3.2-Vision (11B, 90B) (Dubey et al., 2024); (ii) **Closed-source MLLMs**: Claude-3.5-Sonnet (Anthropic, 2024), Gemini-2.5-Flash (DeepMind, 2025), and GPT-4o-mini (OpenAI, 2024). Since no existing AES model is designed for multimodal settings, we use the initial scores produced by each student model — before any teacher MLLM’s feedback or reflection — to serve as the baseline for comparison. This setup ensures that any observed improvements can be fully attributed to the multi-agent process introduced in the CAFES. Detailed rubrics, prompts and model sources are listed in Appendix B.1, B.2, and B.3.

**Evaluation Metric.** After reviewing previous AES studies (Song et al., 2024; Lee et al., 2024b; Mathias and Bhattacharyya, 2018), we select QWK, Pearson’s Correlation Coefficient (PCC) and Spearman’s Rank Correlation Coefficient (SCC) as our evaluation metrics, which are commonly used for assessing model alignment with ground truth scores. The QWK formula is expressed as:

$$k = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}},$$

where  $w_{i,j} = \frac{(i-j)^2}{(N-1)^2}$  is the element of the weight matrix penalizing larger differences between  $i$  and  $j$ ,  $O_{i,j}$  is the observed agreement, and  $E_{i,j}$  is the expected agreement under random chance. QWK values range from -1 to 1. We report results using PCC and SCC as supplementary metrics in Appendix B.4. In general, higher values are expected.

## 4.2 Main Results

**Our proposed CAFES framework yields consistent and significant improvements of QWK across each student MLLM on most traits.** All results of QWK are shown in Table 2. Compared to the initial scores of single MLLMs, it achieves a 21% relative improvement in average QWK. In addition, CAFES brings a 12% average improvement in PCC and a 9% average improvement in

<sup>1</sup>The initial scores are generated in a single inference over ten traits, whereas EssayJudge predicts one trait per inference.

SCC. These results clearly demonstrate the robustness and effectiveness of CAFES.

**The CAFES framework yields the most significant improvements in grammatical and lexical diversity.** As shown in Figure 4, these two traits show the greatest improvements under the CAFES framework. This is because single student MLLMs tend to focus on surface-level errors of grammar and vocabulary in initial scoring while overlooking the positive aspects of diversity (Su et al., 2025; Kundu and Barbosa, 2024), leading to underestimation compared to human raters (as shown in Appendix B.5). With the help of the Feedback Pool Manager, the agent highlights key strengths and passes them to the Reflective Scorer, enabling more accurate recognition of diverse expressions and better-aligned score revisions.

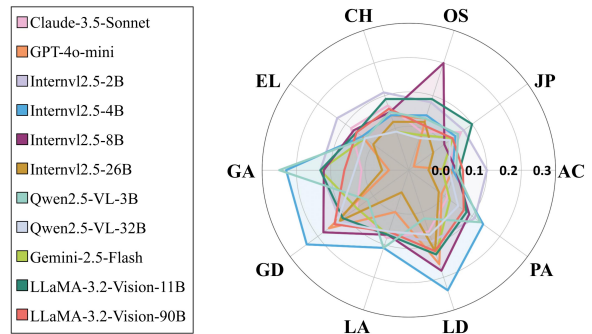


Figure 4: Trait-level score improvements after reflection via CAFES across different student MLLMs.

**In general, the lower the QWK of initial score generated by Initial Scorer, the greater the QWK improvement brought by CAFES.** This trend appears both within and across student MLLMs. Within a student MLLM, traits with a lower initial QWK tend to improve more with CAFES (as shown in Figure 5). More examples are shown in Appendix B.6. Across different student MLLMs, those with weaker initial performance benefit more from CAFES framework. For instance, Claude-3.5-Sonnet, one of the best-performing models initially, achieved only a 0.04 absolute gain (8% relative improvement), while LLaMA-3.2-11B-Vision achieved a 0.09 gain (47.8% relative improvement). A full comparison of QWK improvements across all student MLLMs is provided in the Appendix B.6. This is likely because lower-performing traits or MLLMs have more room for improvement, while stronger ones are already close to the teacher MLLM’s level.

MLLMs	Lexical Level		Sentence Level				Discourse Level			
	LA	LD	CH	GA	GD	PA	AC	JP	OS	EL
<i>Open-Source MLLMs</i>										
InternVL2.5-8B (Chen et al., 2025b)	0.32	0.20	0.27	0.27	0.11	0.21	0.26	0.39	0.09	0.10
+ CAFES (Ours)	<b>0.38</b>	<b>0.37</b>	<b>0.32</b>	<b>0.38</b>	<b>0.29</b>	<b>0.29</b>	<b>0.27</b>	0.39	<b>0.29</b>	<b>0.17</b>
Improvements	<u>+0.07</u>	<u>+0.18</u>	<u>+0.05</u>	<u>+0.12</u>	<u>+0.18</u>	<u>+0.09</u>	<u>+0.01</u>	-	<u>+0.20</u>	<u>+0.07</u>
Qwen2.5-VL-3B (Chen et al., 2025c)	0.19	0.28	0.34	0.19	0.29	0.20	0.26	0.29	0.34	0.32
+ CAFES (Ours)	<b>0.30</b>	<b>0.30</b>	<b>0.39</b>	<b>0.44</b>	<b>0.32</b>	<b>0.34</b>	<b>0.27</b>	<b>0.35</b>	<b>0.37</b>	<b>0.35</b>
Improvements	<u>+0.11</u>	<u>+0.02</u>	<u>+0.05</u>	<u>+0.25</u>	<u>+0.02</u>	<u>+0.13</u>	<u>+0.01</u>	<u>+0.05</u>	<u>+0.03</u>	<u>+0.03</u>
Qwen2.5-VL-32B (Chen et al., 2025c)	0.43	0.40	<b>0.50</b>	0.48	0.39	0.38	0.26	0.35	<b>0.46</b>	0.22
+ CAFES (Ours)	<b>0.49</b>	<b>0.47</b>	0.49	<b>0.51</b>	<b>0.51</b>	<b>0.43</b>	0.26	<b>0.38</b>	0.43	<b>0.25</b>
Improvements	<u>+0.06</u>	<u>+0.07</u>	-0.01	<u>+0.03</u>	<u>+0.13</u>	<u>+0.05</u>	-	<u>+0.02</u>	-0.03	<u>+0.03</u>
LLaMA-3.2-Vision-11B (Dubey et al., 2024)	0.25	0.16	0.22	0.22	0.17	0.21	0.11	0.16	0.20	0.14
+ CAFES (Ours)	<b>0.32</b>	<b>0.29</b>	<b>0.31</b>	<b>0.35</b>	<b>0.28</b>	<b>0.29</b>	<b>0.14</b>	<b>0.27</b>	<b>0.29</b>	<b>0.20</b>
Improvements	<u>+0.07</u>	<u>+0.13</u>	<u>+0.10</u>	<u>+0.13</u>	<u>+0.11</u>	<u>+0.08</u>	<u>+0.02</u>	<u>+0.10</u>	<u>+0.09</u>	<u>+0.06</u>
LLaMA-3.2-Vision-90B (Dubey et al., 2024)	0.40	0.29	0.38	0.40	0.30	0.32	0.21	0.30	0.35	0.16
+ CAFES (Ours)	<b>0.45</b>	<b>0.42</b>	<b>0.43</b>	<b>0.46</b>	<b>0.44</b>	<b>0.39</b>	<b>0.25</b>	<b>0.33</b>	<b>0.37</b>	<b>0.22</b>
Improvements	<u>+0.06</u>	<u>+0.12</u>	<u>+0.06</u>	<u>+0.06</u>	<u>+0.14</u>	<u>+0.07</u>	<u>+0.03</u>	<u>+0.03</u>	<u>+0.01</u>	<u>+0.06</u>
<i>Closed-Source MLLMs</i>										
Claude-3.5-Sonnet (Anthropic, 2024)	0.50	0.53	0.49	0.54	0.52	0.55	<b>0.41</b>	0.39	0.57	0.27
+ CAFES (Ours)	<b>0.58</b>	<b>0.59</b>	<b>0.55</b>	<b>0.55</b>	<b>0.58</b>	<b>0.55</b>	0.39	<b>0.45</b>	0.57	<b>0.35</b>
Improvements	<u>+0.08</u>	<u>+0.06</u>	<u>+0.07</u>	<u>+0.01</u>	<u>+0.06</u>	<u>+0.01</u>	-0.02	<u>+0.06</u>	-	<u>+0.08</u>
Gemini-2.5-Flash (DeepMind, 2025)	0.33	0.26	0.47	0.28	0.30	0.36	<b>0.31</b>	0.32	<b>0.51</b>	0.23
+ CAFES (Ours)	<b>0.40</b>	<b>0.39</b>	0.47	<b>0.41</b>	<b>0.36</b>	<b>0.38</b>	0.28	<b>0.34</b>	0.50	<b>0.26</b>
Improvements	<u>+0.07</u>	<u>+0.13</u>	-	<u>+0.13</u>	<u>+0.06</u>	<u>+0.02</u>	-0.03	<u>+0.03</u>	-0.01	<u>+0.03</u>
GPT-4o-mini (OpenAI, 2024)	0.51	0.34	0.48	<b>0.64</b>	0.38	<b>0.50</b>	0.37	<b>0.55</b>	0.45	0.24
+ CAFES (Ours)	0.51	<b>0.50</b>	<b>0.52</b>	0.57	<b>0.54</b>	0.49	0.37	0.44	<b>0.48</b>	<b>0.28</b>
Improvements	-	<u>+0.16</u>	<u>+0.04</u>	-0.07	<u>+0.15</u>	-0.01	-	-0.11	<u>+0.03</u>	<u>+0.04</u>
<i>Human Performance</i>										
Human performance	0.91	0.91	0.89	0.93	0.56	0.86	0.72	0.86	0.88	0.77

Table 2: QWK scores of different student MLLMs on ten multi-granular essay traits. For each MLLM, the first row shows the baseline performance<sup>1</sup>, the second shows the final result with the CAFES framework, and the third shows the improvement. Only positive improvements are underlined. The best results are highlighted in **bold**. All values are reported after rounding to two decimal places, strictly following a consistent rounding rule.

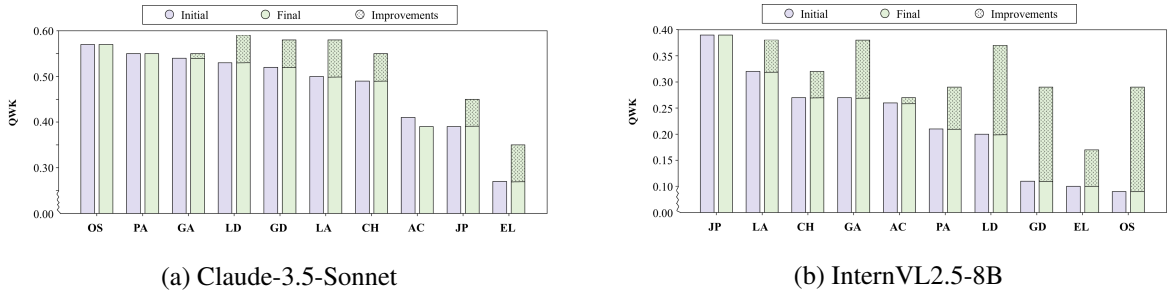


Figure 5: Improvements of QWK score across all traits based on different student MLLMs.

### 4.3 Image Setting Analysis

**CAFES framework yields greater improvements in the multi-image setting across most traits.** The average QWK improvements across all traits except OS and CH are 0.058 in single-image settings and 0.065 in multi-image settings. This is

because MLLMs face greater difficulty in interpreting complex visual inputs in multi-image settings, which leads to more conservative scores. This conservative scoring provides room for adjustment, allowing the CAFES to achieve more noticeable improvements. Notably, these more improvements

of QWK in multi-image settings also support the necessity of incorporating multimodal inputs.

**For traits like Organizational Structure and Coherence, single-image topics yield greater improvements than multi-image topics.** Unlike most traits where multi-image topics lead to greater improvements, OS and CH show higher QWK values in the single-image setting. Specifically, OS reaches 0.04 in single-image compared to 0.03 in multi-image, and CH reaches 0.05 compared to 0.02. This may be because single-image topics contain less visual information, which lowers the demand for essay structure and coherence in students’ essays and makes it easier for MLLMs to assess these traits with confidence after reflection.

#### 4.4 Teacher MLLM Analysis

**CAFES is a flexible and modular framework that does not rely on any specific teacher MLLM.** To demonstrate this, we conducted additional experiments using Qwen2.5-VL-32B as the teacher and Qwen2.5-VL-3B as the student. CAFES achieved an average QWK improvement of 0.10 across ten traits, with several traits even outperforming the teacher model itself (as shown in Table 3). More results of other student MLLMs are available in the Appendix B.7. These findings highlight CAFES’ ability to integrate different teacher MLLMs and adapt to various scenarios and resource constraints without affecting the reproducibility.

Setting	AC	JP	OS	CH	EL
<b>Before</b>	0.26	0.29	0.34	0.34	0.32
<b>After</b>	<b>0.27</b>	<b>0.39</b>	<b>0.38</b>	<b>0.42</b>	<b>0.37</b>

Setting	GA	GD	LA	LD	PA
<b>Before</b>	0.19	0.29	0.19	0.28	0.20
<b>After</b>	<b>0.44</b>	<b>0.33</b>	<b>0.36</b>	<b>0.36</b>	<b>0.37</b>

Table 3: QWK scores before and after applying CAFES framework, using Qwen2.5-VL-32B as the teacher MLLM and Qwen2.5-VL-3B as the student MLLM.

#### 4.5 Prompt Robustness Analysis

**CAFES is robust to variations in prompt structure.** To further validate the robustness of the prompt structure, we conducted an additional experiment where the positions of the Task Definition and Reference Content sections in the prompts for the Reflective Scorer and Feedback Pool Manager modules were swapped. This experiment was performed using InternVL2.5-2B as the student model. Notably, the average QWK remains unchanged at

0.146 before and after the prompt modification, indicating that CAFES’ scoring performance is stable even under variations in prompt structure. Detailed results of trait-level QWK are provided in Table 4.

Setting	AC	JP	OS	CH	EL
<b>Before</b>	0.10	0.06	0.18	0.23	0.17
<b>After</b>	0.12	0.08	0.13	0.18	0.17

Setting	GA	GD	LA	LD	PA
<b>Before</b>	0.11	0.20	0.10	0.17	0.14
<b>After</b>	0.15	0.20	0.11	0.16	0.16

Table 4: Comparison of trait-level QWK scores before and after swapping the Task Definition and Reference Content sections in the prompts.

#### 4.6 Scaling Analysis

**The performance of the student MLLM consistently improves with the scale of MLLM parameters.** We observe a trend similar to the scaling law (Kaplan et al., 2020) in our setting. As shown in Figure 6, when the size of InternVL2.5 increases from 2B to 26B, the average QWK score rises from 0.045 to 0.33 in the initial scoring stage. After incorporating CAFES framework, the performance further improves, with the QWK increasing from 0.146 to 0.335. This result suggests that larger MLLMs exhibit stronger alignment with human judgment and greater reasoning ability.

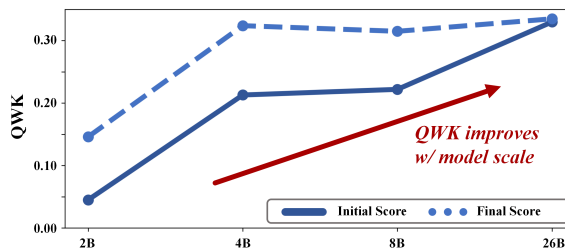


Figure 6: The average QWK scores of InternVL2.5 at different parameter scales (2B, 4B, 8B, and 26B), comparing initial baseline performance with final scores after applying the CAFES framework.

#### 4.7 Ablation Study

We conduct two ablation studies to test key components of CAFES. The first study removes the Feedback Pool Manager. The second study uses the same MLLM for student and teacher models.

**Removing the Feedback Pool Manager results in reflected scores worse than the initial ones.** In this variant, we allowed the initial scores to be directly revised without incorporating Feedback Pool. To evaluate the impact, we applied

this setup to two MLLMs as the student model: Claude-3.5-Sonnet (closed-source) and Qwen2.5-VL-32B (open-source). In both cases, as shown in Figure 7, QWK scores significantly dropped after reflection when the feedback pool was omitted. This degradation suggests that without structured, trait-level feedback, CAFES may overemphasize errors while neglecting strengths in the essays, leading to biased or overly critical revisions. These findings reinforce the necessity of maintaining a balanced feedback mechanism to support more human-aligned and nuanced score adjustments.

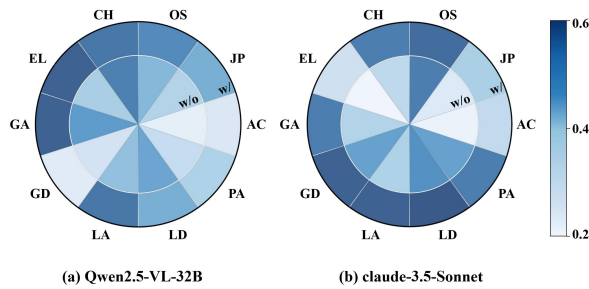


Figure 7: Trait-level QWK comparisons w/ and w/o Feedback Pool Manager for Qwen2.5-VL-32B & Claude-3.5-Sonnet. The inner circle represents results without the Feedback Pool Manager (w/o), and the outer circle represents results with the Feedback Pool (w/).

**Removing the teacher-student collaboration mechanism leads to a drop in QWK.** To investigate the importance of this mechanism, we test two variants of our framework: one where LLaMA-3.2-Vision-90B serves as both the student and teacher, and another where GPT-4o fulfills both roles. As presented in Table 5, QWK decreases in both cases compared to the original teacher-student configuration of CAFES framework. This decline occurs even when both roles are assigned to GPT-4o, highlighting that merely using a powerful model for both positions does not guarantee optimal performance. These findings suggest that the deliberate role differentiation and independent reasoning between the student and teacher MLLMs are critical for achieving effective score revisions in the cross-agent collaborative framework.

#### 4.8 Case Study

To demonstrate how our CAFES framework revises scores through feedback and reflection, we show an example using Claude-3.5-Sonnet as the student MLLM (as shown in Figure 8). More examples are shown in Appendix C. The essay explains how ethanol is produced, based on a flow chart.

MLLM	Baseline	w/	w/o
GPT-4o	<b>0.54</b>	-	0.49
LLaMA-3.2-Vision-90B	0.31	<b>0.38</b>	0.32

Table 5: Average QWK scores across traits for two MLLMs (GPT-4o and LLaMA-3.2-Vision-90B), comparing the baseline performance (*i.e.*, initial score), with results obtained with (w/) and without (w/o) the proposed teacher-student collaboration mechanism.

For example, we can find that the student MLLM initially gives a low argument clarity score of 1, while the ground truth score is 3. As mentioned in *Main Results*, this is because the MLLM focuses too much on surface-level errors and overlooks key strengths, such as the relevant introduction and logical structure. After receiving trait-specific feedback from the teacher MLLM, the Reflective Scorer revises the score upward. The final score better matches the human judgment, showing that targeted feedback helps correct overly harsh assessments and highlight overlooked merits. This case illustrates how the CAFES framework leverages structured feedback and a subsequent reflection step to refine the initial model output, leading to better alignment with human preferences.

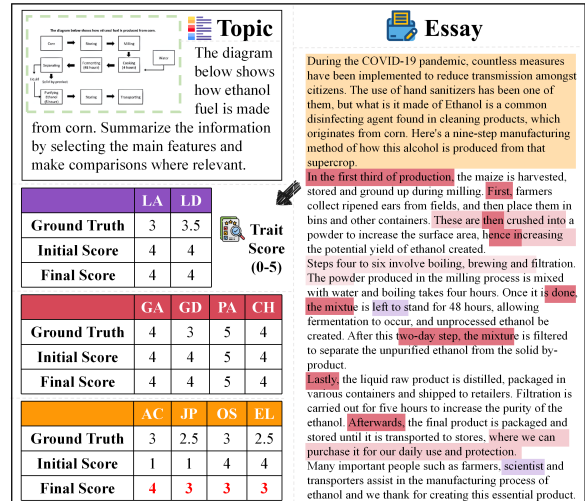


Figure 8: A representative case study illustrating CAFES' score revision process. The student MLLM is Claude-3.5-Sonnet, and the teacher MLLM is GPT-4o.

## 5 Conclusion

In this work, we present CAFES, the first collaborative multi-agent framework for AES task. It divides the essay scoring process into three core stages (*i.e.*, initial scoring, feedback generation, and reflective revision), enabling structured collab-

oration between three agents. Experiments across different student MLLMs show significant QWK improvements with CAFES framework, especially in grammatical and lexical diversity. Ablation studies further confirm the necessity of the Feedback Pool Manager and teacher-student collaboration mechanism. We hope CAFES can offer a new paradigm for building reliable and human-aligned AES systems and encourage the community to advance more effective and accurate scoring methods.

## Limitations

Despite the improvements we demonstrate in our CAFES framework, there are still minor limitations:

1. Our framework is evaluated on the Essay-Judge dataset and achieves notable improvements over baseline models. However, Essay-Judge — the only available multimodal essay dataset — focuses mainly on chart-based topics and does not cover more complex visual inputs such as film frames. We plan to include a broader range of multimodal essay types in future evaluations.
2. The reflection mechanism helps suppress hallucinations from a single MLLM, such as fabricated justifications or misinterpretation of charts, but hallucination-induced scoring errors still occur. In future work, we aim to further strengthen evidence grounding.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No.62506318); Guangdong Provincial Department of Education Project (Grant No.2024KQNCX028); CAAI-Ant Group Research Fund; Scientific Research Projects for the Higher-educational Institutions (Grant No.2024312096), Education Bureau of Guangzhou Municipality; Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2025A03J3957), Education Bureau of Guangzhou Municipality

## References

Anthropic. 2024. [Claude 3.5 sonnet](#).

John Atkinson and Diego Palma. 2025. An llm-based hybrid approach for enhanced automated essay scoring. *Scientific Reports*, 15(1):14551.

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v.2. *Journal of Technology, Learning, and Assessment*, 4(3). *Journal of Technology, Learning, and Assessment*, 4.

Stacey Bailey and Detmar Meurers. 2008. Diagnosing meaning errors in short answers to reading comprehension questions. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 107–115.

Yida Cai, Kun Liang, Sanwoo Lee, Qinghan Wang, and Yunfang Wu. 2025. Rank-then-score: Enhancing large language models for automated essay scoring. *arXiv preprint arXiv:2504.05736*.

Yue Cao, Hanqi Jin, Xiaojun Wan, and Zhiwei Yu. 2020. Domain-adaptive neural automated essay scoring. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1011–1020. Association for Computing Machinery.

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752.

Junkai Chen, Zhijie Deng, Kening Zheng, Yibo Yan, Shuliang Liu, PeiJun Wu, Peijie Jiang, Jia Liu, and Xuming Hu. 2025a. Safeeraser: Enhancing safety in multimodal large language models through multimodal machine unlearning. *arXiv preprint arXiv:2502.12520*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, and 23 others. 2025b. [Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling](#). *Preprint*, arXiv:2412.05271.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, and 23 others. 2025c. [Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling](#). *Preprint*, arXiv:2412.05271.

Jaeyoon Choi, Tamara Tate, Daniel Ritchie, Nia Nixon, and Mark Warschauer. 2025. Anchor is the key: Toward accessible automated essay scoring with large language models through prompting.

Zhendong Chu, Shen Wang, Jian Xie, Tinghui Zhu, Yibo Yan, Jinheng Ye, Aoxiao Zhong, Xuming Hu, Jing Liang, Philip S Yu, and 1 others. 2025. Llm agents for education: Advances and applications. *arXiv preprint arXiv:2503.11733*.

- Mădălina Cozma, Andrei Butnaru, and Radu Tudor Ionescu. 2018. Automated essay scoring with string kernels and word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 503–509.
- Yunkai Dang, Kaichen Huang, Jiahao Huo, Yibo Yan, Sirui Huang, Dongrui Liu, Mengxi Gao, Jie Zhang, Chen Qian, Kun Wang, and 1 others. 2024. Explainable and interpretable multimodal large language models: A comprehensive survey. *arXiv preprint arXiv:2412.02104*.
- Google DeepMind. 2025. [Gemini 2.5 flash](#).
- Heejin Do, Sangwon Ryu, and Gary Lee. 2024. [Autoregressive multi-trait essay scoring via reinforcement learning with scoring-aware multiple rewards](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida, USA. Association for Computational Linguistics.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Sylviane Granger, Estelle Dagneaux, Fanny Meunier, and Magali Paquot. 2009. *International Corpus of Learner English. Version 2. Handbook and CD-ROM*.
- Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, and 6 others. 2024. [Minicpm: Unveiling the potential of small language models with scalable training strategies](#). *Preprint*, arXiv:2404.06395.
- Kaichen Huang, Jiahao Huo, Yibo Yan, Kun Wang, Yutao Yue, and Xuming Hu. 2024. Miner: Mining the underlying pattern of modality-specific neurons in multimodal large language models. *arXiv preprint arXiv:2410.04819*.
- Jiahao Huo, Yibo Yan, Boren Hu, Yutao Yue, and Xuming Hu. 2024. Mmneuron: Discovering neuron-level domain-specific interpretation in multimodal large language model. *arXiv preprint arXiv:2406.11193*.
- Jiahao Huo, Yibo Yan, Xu Zheng, Yuanhuiyi Lyu, Xin Zou, Zhihua Wei, and Xuming Hu. 2025. Mmunlearner: Reformulating multimodal machine unlearning in the era of multimodal large language models. *arXiv preprint arXiv:2502.11051*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Zhiwei Jiang, Tianyi Gao, Yafeng Yin, Meng Liu, Hua Yu, Zifeng Cheng, and Qing Gu. 2023. [Improving domain generalization for prompt-aware essay scoring via disentangled representation learning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12456–12470. Association for Computational Linguistics.
- Firuz Kamalov, David Santandreu Calonge, Linda Smail, Dilshod Azizov, Dimple R Thadani, Theresa Kwong, and Amara Atif. 2025. Evolution of ai in education: Agentic workflows. *arXiv preprint arXiv:2504.20082*.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
- Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6300–6308. International Joint Conferences on Artificial Intelligence Organization.
- Anindita Kundu and Denilson Barbosa. 2024. [Are large language models good essay graders?](#)
- Sanwoo Lee, Yida Cai, Desong Meng, Ziyang Wang, and Yunfang Wu. 2024a. Unleashing large language models’ proficiency in zero-shot essay scoring. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 181–198.
- Sanwoo Lee, Yida Cai, Desong Meng, Ziyang Wang, and Yunfang Wu. 2024b. [Unleashing large language models’ proficiency in zero-shot essay scoring](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 181–198, Miami, Florida, USA. Association for Computational Linguistics.
- Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. [Camel: Communicative agents for "mind" exploration of large language model society](#). *Preprint*, arXiv:2303.17760.
- Hang Li, Tianlong Xu, Chaoli Zhang, Eason Chen, Jing Liang, Xing Fan, Haoyang Li, Jiliang Tang, and Qingsong Wen. 2024. Bringing generative ai to adaptive learning in education. *arXiv preprint arXiv:2402.14601*.
- Shengjie Li and Vincent Ng. 2024a. Automated essay scoring: A reflection on the state of the art. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17876–17888.
- Shengjie Li and Vincent Ng. 2024b. Automated essay scoring: Recent successes and future directions. In *Proceedings of the Thirty-Third International*

- Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8114–8122.
- Shengjie Li and Vincent Ng. 2024c. Icle++: Modeling fine-grained traits for holistic essay scoring. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8458–8478.
- Wenchao Li and Haitao Liu. 2024. Applying large language models for automated essay scoring for non-native japanese. *Humanities and Social Sciences Communications*, 11(1):1–15.
- Xia Li and Wenjing Pan. 2025. [CEAES: Bidirectional reinforcement learning optimization for consistent and explainable essay assessment](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26267–26279, Vienna, Austria. Association for Computational Linguistics.
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.
- Shalev Lifshitz, Sheila A McIlraith, and Yilun Du. 2025. Multi-agent verification: Scaling test-time compute with multiple verifiers. *arXiv preprint arXiv:2502.20379*.
- Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. [Llava-next: Improved reasoning, ocr, and world knowledge](#).
- Zeyang Liu, Xinrui Yang, Shiguang Sun, Long Qian, Lipeng Wan, Xingyu Chen, and Xuguang Lan. 2024b. Grounded answers for multi-agent decision-making problem through generative world model. *Advances in Neural Information Processing Systems*, 37:46622–46652.
- Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, and 1 others. 2024. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*.
- Watheq Mansour, Salam Albatarni, Sohaila Eltanbouly, and Tamer Elsayed. 2024. [Can large language models automatically score proficiency of written essays? Preprint](#), arXiv:2403.06149.
- Sandeep Mathias and Pushpak Bhattacharyya. 2018. ASAP++: Enriching the ASAP automated essay grading dataset with essay attribute scores. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Atsushi Mizumoto and Masaki Eguchi. 2023. Exploring the potential of using an ai language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2):100050.
- OpenAI. 2024. [Gpt-4o mini: advancing cost-efficient intelligence](#).
- Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527.
- SeungWoo Song, Junghun Yuk, ChangSu Choi, HanGyeol Yoo, Hyeonseok Lim, KyungTae Lim, and Jungyeul Park. 2025. Unified automated essay scoring and grammatical error correction. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4412–4426.
- Yishen Song, Qianta Zhu, Huaibo Wang, and Qinhuang Zheng. 2024. Automated essay scoring and revising based on open-source large language models. *IEEE Transactions on Learning Technologies*.
- Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510.
- Jiamin Su, Yibo Yan, Fangteng Fu, Han Zhang, Jingheng Ye, Xiang Liu, Jiahao Huo, Huiyu Zhou, and Xuming Hu. 2025. Essayjudge: A multi-granular benchmark for assessing automated essay scoring capabilities of multimodal large language models.
- Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O’Sullivan, and Hoang D. Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms.
- Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. Neural automated essay scoring incorporating hand-crafted features. In *Proceedings of the 28th international conference on computational linguistics*, pages 6077–6088.
- Sowmya Vajjala. 2016. Automated assessment of non-native learner essays: Investigating the role of linguistic features. *CoRR*.
- Jiong Wang and Jie Liu. 2025. T-MES: Trait-aware mix-of-experts representation learning for multi-trait essay scoring. In *Proceedings of the 31st International Conference on Computational Linguistics*. Association for Computational Linguistics.
- Yongjie Wang, Chuang Wang, Ruobing Li, and Hui Lin. 2022. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3416–3425.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryan W White, Doug Burger, and Chi

- Wang. 2023. [Autogen: Enabling next-gen llm applications via multi-agent conversation](#). *Preprint*, arXiv:2308.08155.
- Xuansheng Wu, Padmaja Pravin Saraf, Gyeong-Geon Lee, Ehsan Latif, Ninghao Liu, and Xiaoming Zhai. 2024. Unveiling scoring processes: Dissecting the differences between llms and human graders in automatic scoring. *arXiv preprint arXiv:2407.18328*.
- Wei Xia, Shaoguang Mao, and Chanjing Zheng. 2024. Empirical study of large language models as automated essay scoring tools in english composition\_taking toefl independent writing task for example. *arXiv preprint arXiv:2401.03401*.
- Changrong Xiao, Wenxing Ma, Qingping Song, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Qi Fu. 2024. [Human-ai collaborative essay scoring: A dual-process framework with llms](#). *Preprint*, arXiv:2401.06431.
- Wenbo Xu, Muhammad Shahreeza, Wai Lam Hoo, and Wudao Yang. 2025. Explainable ai for education: Enhancing essay scoring via rubric-aligned chain-of-thought prompting.
- Yibo Yan and Joey Lee. 2024. Georeasoner: Reasoning on geospatially grounded context for natural language understanding. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4163–4167.
- Yibo Yan, Jiamin Su, Jianxiang He, Fangteng Fu, Xu Zheng, Yuanhuiyi Lyu, Kun Wang, Shen Wang, Qingsong Wen, and Xuming Hu. 2024a. A survey of mathematical reasoning in the era of multimodal large language model: Benchmark, method & challenges. *arXiv preprint arXiv:2412.11936*.
- Yibo Yan, Shen Wang, Jiahao Huo, Hang Li, Boyan Li, Jiamin Su, Xiong Gao, Yi-Fan Zhang, Tianlong Xu, Zhendong Chu, and 1 others. 2024b. Errorradar: Benchmarking complex mathematical reasoning of multimodal large language models via error detection. *arXiv preprint arXiv:2410.04509*.
- Yibo Yan, Shen Wang, Jiahao Huo, Jingheng Ye, Zhendong Chu, Xuming Hu, Philip S Yu, Carla Gomes, Bart Selman, and Qingsong Wen. 2025a. Position: Multimodal large language models can significantly advance scientific reasoning. *arXiv preprint arXiv:2502.02871*.
- Yibo Yan, Shen Wang, Jiahao Huo, Philip S Yu, Xuming Hu, and Qingsong Wen. 2025b. Mathagent: Leveraging a mixture-of-math-agent framework for real-world multimodal mathematical error detection. *arXiv preprint arXiv:2503.18132*.
- Yibo Yan, Haomin Wen, Sirui Zhong, Wei Chen, Haodong Chen, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. 2024c. Urbanclip: Learning text-enhanced urban region profiling with contrastive language-image pretraining from the web. In *Proceedings of the ACM on Web Conference 2024*, pages 4006–4017.
- Kaixun Yang, Mladen Raković, Yuyang Li, Quanlong Guan, Dragan Gašević, and Guangliang Chen. 2024. Unveiling the tapestry of automated essay scoring: A comprehensive investigation of accuracy, fairness, and generalizability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22466–22474.
- Helen Yannakoudakis and Ted Briscoe. 2012. Modeling coherence in ESOL learner texts. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 33–43.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189.
- Jingheng Ye, Shen Wang, Deqing Zou, Yibo Yan, Kun Wang, Hai-Tao Zheng, Zenglin Xu, Irwin King, Philip S Yu, and Qingsong Wen. 2025. Position: Llms can be good tutors in foreign language education. *arXiv preprint arXiv:2502.05467*.
- Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, and 1 others. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.
- Zhiqiang Yuan, Weitong Chen, Hanlin Wang, Kai Yu, Xin Peng, and Yiling Lou. 2024. Transagent: An llm-based multi-agent system for code translation. *arXiv preprint arXiv:2409.19894*.
- Kening Zheng, Junkai Chen, Yibo Yan, Xin Zou, and Xuming Hu. 2024. Reefknot: A comprehensive benchmark for relation hallucination evaluation, analysis and mitigation in multimodal large language models. *arXiv preprint arXiv:2408.09429*.
- Guanyu Zhou, Yibo Yan, Xin Zou, Kun Wang, Aiwei Liu, and Xuming Hu. 2024. Mitigating modality prior-induced hallucinations in multimodal large language models via deciphering attention causality. *arXiv preprint arXiv:2410.04780*.
- Xin Zou, Yizhou Wang, Yibo Yan, Sirui Huang, Kening Zheng, Junkai Chen, Chang Tang, and Xuming Hu. 2024. Look twice before you answer: Memory-space visual retracing for hallucination mitigation in multimodal large language models. *arXiv preprint arXiv:2410.03577*.

## A More Related Work

### A.1 AES Datasets

Table 6 summarizes widely used AES datasets in terms of dataset size, number of essay topics, modality, and trait-level annotations. Most existing datasets (e.g., ASAP, CLC-FCE, TOEFL11) are unimodal and offer either holistic scores or a limited number of traits, primarily focusing on text-based prompts. Recently, EssayJudge (Su et al., 2025) has been introduced as a multimodal benchmark that includes both textual and visual inputs, covering 125 topics and annotated across ten fine-grained scoring traits. This enables more comprehensive evaluation of AES systems, especially those leveraging MLLMs.

Benchmarks	Venue	Size	#Topics	Modality	#Traits
ASAP <sub>AES</sub> (Cozma et al., 2018)	ACL	17,450	8	T	0
ASAP++ (Mathias and Bhattacharyya, 2018)	ACL	10,696	6	T	8
CLC-FCE (Yannakoudakis et al., 2011)	ACL	1,244	10	T	0
TOEFL11 (Lee et al., 2024a)	EMNLP	1,100	8	T	0
ICLE (Granger et al., 2009)	COLING	3,663	48	T	4
AAE (Stab and Gurevych, 2014)	COLING	102	101	T	1
ICLE++ (Li and Ng, 2024c)	NAACL	1,008	10	T	10
CREE (Bailey and Meurers, 2008)	BEA	566	75	T	1
EssayJudge (Su et al., 2025)	-	1054	125	T, I	10

Table 6: Comparison between previous AES datasets.

### A.2 Multimodal Large Language Models

MLLMs have experienced rapid development in recent years and have been widely adopted across various domains (Yan et al., 2024a). Their core advantage lies in the ability to jointly process visual and textual inputs to handle a range of complex tasks (Huo et al., 2024; Yan et al., 2024c; Yan and Lee, 2024; Dang et al., 2024; Huo et al., 2025; Chen et al., 2025a). On the proprietary side, MLLMs such as GPT-4o (Hurst et al., 2024) and Gemini-1.5 (DeepMind, 2025) have demonstrated state-of-the-art performance in multimodal reasoning, instruction following, and question answering tasks (Yan et al., 2024b,a; Zheng et al., 2024; Yan et al., 2025a). Meanwhile, open-source MLLMs have made notable advances in terms of accessibility and modularity. LLaVA-NEXT (Liu et al., 2024a) employs pretrained encoders and adapters to align vision and language representations efficiently. Other representative MLLMs—such as Qwen2.5-VL (Chen et al., 2025c), DeepSeek-VL (Lu et al., 2024), InternVL (Chen et al., 2025b), Yi-VL (Young et al., 2024), LLaMA3-VL (Dubey et al., 2024), and MiniCPM-V (Hu et al., 2024)—have introduced a variety of fusion mechanisms, including visual projection heads, mixture-of-experts ar-

chitectures, and image-grounded token masking. These MLLMs have been applied to a wide range of domains, including education and medical diagnostics (Zou et al., 2024; Zhou et al., 2024; Huang et al., 2024), showcasing the expanding scope and depth of MLLM capabilities.

Building upon these diverse MLLMs, our proposed CAFES multi-agent framework flexibly incorporates different MLLMs as the backbone for each agent module, enabling collaborative interaction between student and teacher MLLMs to enhance the accuracy and robustness of AES.

## B Additional Experimental Details

### B.1 Trait-Specific Rubrics

In this section, we introduce the rubrics of the 10 traits which is similar to EssayJudge. The rubrics are detailed in Table 7 to Table 16. Each trait is assessed using a numerical score ranging from 0 to 5. A score of 5 represents high-quality performance with respect to the trait being evaluated, while a score of 0 represents low-quality performance in the same regard.

### B.2 Prompt for CAFES framework

Our agent-based CAFES framework consists of three modules—Initial Scorer, Feedback Pool Generator, and Reflective Scorer—each employing customized prompt designs for their functional roles. While all agents operate under a unified trait-based rubric schema, the input structure and expected output vary to support multi-stage evaluation. The details are shown in Figure 9 to 11.

### B.3 Model Sources

Table 17 details specific sources for the various student MLLMs. The hyperparameters for the experiments are set to their default values unless specified otherwise.

### B.4 PCC and SCC Results

In addition to the Quadratic Weighted Kappa (QWK) metric reported in the main text, we further conducted experiments using Pearson’s Correlation Coefficient (PCC) and Spearman’s Correlation Coefficient (SCC) as evaluation metrics to assess the correlation between the predicted scores and ground truth.

As shown in Table 18 and 19, the final scores generated by our multi-agent framework exhibit improved correlation with human ratings across

the majority of traits, demonstrating better consistency compared to the initial scores. This indicates that the fine-grained trait optimization within our system effectively enhances alignment with human scoring standards.

Overall, the results show that our method achieves a 12% average improvement in PCC and a 9% average improvement in SCC compared to the initial scores of single MLLMs. These findings provide strong evidence of the robustness and generalization capability of the proposed framework.

### B.5 Average Trait-Specific Score Comparison

Closed-source MLLMs tend to adopt a more rigorous scoring strategy compared to open-source MLLMs. This trend is supported by both quantitative and distributional evidence. First, as shown in the Figure 12, closed-source models consistently assign lower average scores than open-source models across most traits, regardless of whether the scores are initial or final (Su et al., 2025; Kundu and Barbosa, 2024). Second, Figure 13 reveals that closed-source models exhibit slightly higher score variance (0.81 vs. 0.79), indicating a broader and possibly more cautious distribution of judgments. Together, these findings suggest that closed-source MLLMs are more aligned with rigorous rubric interpretation, both when directly scoring and when acting as student models within the CAFES framework.

### B.6 Additional Examples of Within- and Across-MLLM Improvements

Beyond the examples of Claude-3.5-Sonnet and InternVL2.5-8B presented in the main paper, additional cases support this observation, as shown in Figure 14. These within-MLLM examples demonstrate that traits with lower initial QWK scores tend to benefit more from the CAFES framework.

In addition, we provide a comprehensive comparison across all student MLLMs in Table 15. This table summarizes the initial QWK scores, absolute improvements, and relative improvements for each model. The results confirm that student MLLMs with weaker initial performance achieve larger relative gains after applying CAFES, highlighting the framework’s generalizability and effectiveness across different models.

### B.7 Teacher MLLM Analysis

To further validate the flexibility and generalizability of CAFES, as described in the main

text, we provide detailed experimental results. Qwen2.5-VL-32B, one of the leading open-source MLLMs, was selected to evaluate CAFES’ modularity. Table 20 shows that, with InternVL2.5-2B as the student, CAFES achieves an average QWK improvement of +0.17 across ten scoring traits. These results reaffirm CAFES’ capability to seamlessly integrate alternative teacher MLLMs and maintain strong performance under different configurations.

## C More Essay Scoring examples

To further illustrate the effectiveness of our multi-agent AES framework, we include several additional essay cases (as shown in Figure 16 to 18). Each example consists of the essay topic, student’s essays, initial scores, feedback pool, and final revised scores after reflection. These examples highlight different error types, model reasoning behaviors, and improvement patterns across traits.

<b>Score</b>	<b>Scoring Criteria</b>
5	The central argument is clear, and the first paragraph clearly outlines the topic of the image and question, providing guidance with no ambiguity.
4	The central argument is clear, and the first paragraph mentions the topic of the image and question, but the guidance is slightly lacking or the expression is somewhat vague.
3	The argument is generally clear, but the expression is vague, and it doesn't adequately guide the rest of the essay.
2	The argument is unclear, the description is vague or incomplete, and it doesn't guide the essay.
1	The argument is vague, and the first paragraph fails to effectively summarize the topic of the image or question.
0	No central argument is presented, or the essay completely deviates from the topic and image.

Table 7: Rubrics for evaluating the argument clarity of the essays.

<b>Score</b>	<b>Scoring Criteria</b>
5	Transitions between sentences are natural, and logical connections flow smoothly; appropriate use of linking words and transitional phrases.
4	Sentences are generally coherent, with some transitions slightly awkward; linking words are used sparingly but are generally appropriate.
3	The logical connection between sentences is not smooth, with some sentences jumping or lacking flow; linking words are used insufficiently or inappropriately.
2	Logical connections are weak, sentence connections are awkward, and linking words are either used too little or excessively.
1	There is almost no logical connection between sentences, transitions are unnatural, and linking words are very limited or incorrect.
0	No coherence at all, with logical confusion between sentences.

Table 8: Rubrics for evaluating the coherence of the essays.

<b>Score</b>	<b>Scoring Criteria</b>
5	Word count is 150 words or more, with the content being substantial and without obvious excess or brevity.
4	Word count is around 150 words, but slightly off (within 10 words), and the content is complete.
3	Word count is noticeably too short or too long, and the content is not sufficiently substantial or is somewhat lengthy.
2	Word count deviates significantly, failing to fully cover the requirements of the prompt.
1	Word count is far below the requirement, and the content is incomplete.
0	Word count is severely insufficient or excessive, making it impossible to meet the requirements of the prompt.

Table 9: Rubrics for evaluating the essay length of the essays.

<b>Score</b>	<b>Scoring Criteria</b>
5	Sentence structure is accurate with no grammatical errors; both simple and complex sentences are error-free.
4	Sentence structure is generally accurate, with occasional minor errors that do not affect understanding; some errors in complex sentence structures.
3	Few grammatical errors, but more noticeable errors that affect understanding; simple sentences are accurate, but complex sentences frequently contain errors.
2	Numerous grammatical errors, with sentence structure affecting understanding; simple sentences are occasionally correct, but complex sentences have frequent errors.
1	A large number of grammatical errors, with sentence structure severely affecting understanding; sentence structure is unstable, and even simple sentences contain mistakes.
0	Sentence structure is completely incorrect, nonsensical, and difficult to understand.

Table 10: Rubrics for evaluating the grammatical accuracy of the essays.

<b>Score</b>	<b>Scoring Criteria</b>
5	Uses a variety of sentence structures, including both simple and complex sentences, with flexible use of clauses and compound sentences, demonstrating rich sentence variation.
4	Generally uses a variety of sentence structures, with appropriate use of common clauses and compound sentences. Sentence structures vary, though some sentence types lack flexibility.
3	Uses a variety of sentence structures, but with limited use of complex sentences, which often contain errors. Sentence variation is somewhat restricted.
2	Sentence structures are simple, primarily relying on simple sentences, with occasional attempts at complex sentences, though errors occur frequently.
1	Sentence structures are very basic, with almost no complex sentences, and even simple sentences contain errors.
0	Only uses simple, repetitive sentences with no complex sentences, resulting in rigid sentence structures.

Table 11: Rubrics for evaluating the grammatical diversity of the essays.

<b>Score</b>	<b>Scoring Criteria</b>
5	Fully addresses and accurately analyzes all important information in the image and prompt (e.g., data turning points, trends); argumentation is in-depth and logically sound.
4	Addresses most of the important information in the image and prompt, with reasonable analysis but slight shortcomings; argumentation is generally logical.
3	Addresses some important information in the image and prompt, but analysis is insufficient; argumentation is somewhat weak.
2	Mentions a small amount of important information in the image and prompt, with simple or incorrect analysis; there are significant logical issues in the argumentation.
1	Only briefly mentions important information in the image and prompt or makes clear analytical errors, lacking reasonable reasoning.
0	Fails to mention key information from the image and prompt, lacks any argumentation, and is logically incoherent.

Table 12: Rubrics for evaluating the justifying persuasiveness of the essays.

<b>Score</b>	<b>Scoring Criteria</b>
5	Vocabulary is accurately chosen, with correct meanings and spelling, and minimal errors; words are used precisely to convey the intended meaning.
4	Vocabulary is generally accurate, with occasional slight meaning errors or minor spelling mistakes, but they do not affect overall understanding; words are fairly precise.
3	Vocabulary is mostly correct, but frequent minor errors or spelling mistakes affect some expressions; word choice is not fully precise.
2	Vocabulary is inaccurate, with significant meaning errors and frequent spelling mistakes, affecting understanding.
1	Vocabulary is severely incorrect, with unclear meanings and noticeable spelling errors, making comprehension difficult.
0	Vocabulary choice and spelling are completely incorrect, and the intended meaning is unclear or impossible to understand.

Table 13: Rubrics for evaluating the lexical accuracy of the essays.

<b>Score</b>	<b>Scoring Criteria</b>
5	Vocabulary is rich and diverse, with a wide range of words used flexibly, avoiding repetition.
4	Vocabulary diversity is good, with a broad range of word choices, occasional repetition, but overall flexible expression.
3	Vocabulary diversity is average, with some variety in word choice but limited, with frequent repetition.
2	Vocabulary is fairly limited, with a lot of repetition and restricted word choice.
1	Vocabulary is very limited, with frequent repetition and an extremely narrow range of words.
0	Vocabulary is monotonous, with almost no variation, failing to demonstrate vocabulary diversity.

Table 14: Rubrics for evaluating the lexical diversity of the essays.

Score	Scoring Criteria
5	The essay has a well-organized structure, with clear paragraph divisions, each focused on a single theme. There are clear topic sentences and concluding sentences, and transitions between paragraphs are natural.
4	The structure is generally reasonable, with fairly clear paragraph divisions, though transitions may be somewhat awkward and some paragraphs may lack clear topic sentences.
3	The structure is somewhat disorganized, with unclear paragraph divisions, a lack of topic sentences, or weak logical flow.
2	The structure is unclear, with improper paragraph divisions and poor logical coherence.
1	The paragraph structure is chaotic, with most paragraphs lacking clear topic sentences and disorganized content.
0	No paragraph structure, content is jumbled, and there is a complete lack of logical connections.

Table 15: Rubrics for evaluating the organizational structure of the essays.

Score	Scoring Criteria
5	Punctuation is used correctly throughout, adhering to standard rules with no errors.
4	Punctuation is mostly correct, with occasional minor errors that do not affect understanding.
3	Punctuation is generally correct, but there are some noticeable errors that slightly affect understanding.
2	There are frequent punctuation errors, some of which affect understanding.
1	Punctuation errors are severe, significantly affecting comprehension.
0	Punctuation is completely incorrect or barely used, severely hindering understanding.

Table 16: Rubrics for evaluating the punctuation accuracy of the essays.

**Task Definition:** You are an experienced English writing examiner. Please evaluate the student's essay by assigning a score (0-5) for each of the ten traits and a confidence level (1–10) that reflects how certain you are about each score, where 1 is least certain and 10 is completely certain.

**Rubrics:** {Trait-specific corresponding rubrics}

**Below is the reference content:**  
Image: "{image}"  
Essay Topic: "{question}"  
Student's Essay: "{essay}"

**Instruction:** Please provide your answer in the same style and format as the example. Use the exact trait names as shown (with proper capitalization) Return your response strictly in JSON format without any additional text, explanations, or code block delimiters (no triple backticks).

Figure 9: Prompt for Initial Scorer.

**Task Definition:** You are an experienced English writing examiner. Your task is to provide detailed positive feedback on a student essay across ten traits: Argument Clarity, Justifying Persuasiveness, Organizational Structure, Coherence, Essay Length, Grammatical Accuracy, Grammatical Diversity, Lexical Accuracy, Lexical Diversity, Punctuation Accuracy.

**Below is the reference content:**

Image: "{image}"

Essay Topic: "{question}"

Student's Essay: "{essay}"

**Instruction:** please generate your feedback dimension by dimension. Your output must be in natural language paragraphs. Do not use JSON, code blocks, or bullet points. Start each dimension with the tag in square brackets, for example: [Argument Clarity]

Sample Format: [Argument Clarity] The opening paragraph clearly introduces the topic of the image and outlines the overall trend, effectively setting up the structure for later analysis.

Figure 10: Prompt for Feedback Pool Manager.

**Task Definition:** You are evaluating a set of essay scores originally provided by another assistant reviewer. A detailed feedback report—including both positive and negative comments across 10 traits—is available for reference, but should not be treated as an absolute judgment. Your task is to serve as a more careful and critical second-round reviewer. Do not assume the original scores are correct — examine each trait carefully and revise any score that appears inaccurate or unsupported by the essay.

**Rubrics:** {Trait-specific corresponding rubrics}

**Below is the reference content:**

Image: "{image}"

Essay Topic: "{question}"

Student's Essay: "{essay}"

Original Scores : "{score}"

Feedback Report : "{feedback}"

**Instruction:** If revision is needed, return only the affected dimensions with new scores and brief reasoning. Otherwise, confirm the original scores. Return your response strictly in JSON format without any additional text, explanations, or code block delimiters (no triple backticks). Only raw JSON is accepted.

Figure 11: Prompt for Reflective Scorer.

<b>MLLMs</b>	<b>Source</b>	<b>URL</b>
InternVL2.5-2B	local checkpoint	<a href="https://huggingface.co/OpenGVLab/InternVL2.5-2B">https://huggingface.co/OpenGVLab/InternVL2.5-2B</a>
InternVL2.5-4B	local checkpoint	<a href="https://huggingface.co/OpenGVLab/InternVL2-4B">https://huggingface.co/OpenGVLab/InternVL2-4B</a>
InternVL2.5-8B	local checkpoint	<a href="https://huggingface.co/OpenGVLab/InternVL2-8B">https://huggingface.co/OpenGVLab/InternVL2-8B</a>
InternVL2.5-26B	local checkpoint	<a href="https://huggingface.co/OpenGVLab/InternVL2-26B">https://huggingface.co/OpenGVLab/InternVL2-26B</a>
Qwen2.5-VL-3B	local checkpoint	<a href="https://huggingface.co/Qwen/Qwen2.5-VL-3B-Instruct">https://huggingface.co/Qwen/Qwen2.5-VL-3B-Instruct</a>
Qwen2.5-VL-32B	local checkpoint	<a href="https://huggingface.co/Qwen/Qwen2.5-VL-32B-Instruct">https://huggingface.co/Qwen/Qwen2.5-VL-32B-Instruct</a>
LLaMA-3.2-Vision-11B	local checkpoint	<a href="https://huggingface.co/meta-llama/Llama-3.2-11B-Vision-Instruct">https://huggingface.co/meta-llama/Llama-3.2-11B-Vision-Instruct</a>
LLaMA-3.2-Vision-90B	local checkpoint	<a href="https://huggingface.co/meta-llama/Llama-3.2-90B-Vision-Instruct">https://huggingface.co/meta-llama/Llama-3.2-90B-Vision-Instruct</a>
Claude-3.5-Sonnet	claude-3.5-sonnet-20241022	<a href="https://www.anthropic.com/claude/sonnet">https://www.anthropic.com/claude/sonnet</a>
Gemini-2.5-Flash	gemini-2.5-flash-preview-04-17	<a href="https://deepmind.google/technologies/gemini/flash">https://deepmind.google/technologies/gemini/flash</a>
GPT-4o-mini	gpt-4o-mini-2024-07-18	<a href="https://platform.openai.com/docs/models/gpt-4o-mini">https://platform.openai.com/docs/models/gpt-4o-mini</a>

Table 17: Sources of our evaluated MLLMs.

<b>MLLMs</b>	<b>Lexical Level</b>		<b>Sentence Level</b>				<b>Discourse Level</b>			
	<b>LA</b>	<b>LD</b>	<b>CH</b>	<b>GA</b>	<b>GD</b>	<b>PA</b>	<b>AC</b>	<b>JP</b>	<b>OS</b>	<b>EL</b>
InternVL2.5-8B	0.36	0.25	0.34	0.32	0.23	0.28	0.27	0.41	0.18	0.23
<b>+ CAFES (Ours)</b>	0.44	0.38	0.40	0.50	0.36	0.40	0.29	0.41	0.34	0.31
Qwen2.5-VL-3B	0.30	0.43	0.41	0.33	0.36	0.26	0.27	0.41	0.40	0.33
<b>+ CAFES (Ours)</b>	0.37	0.44	0.44	0.48	0.37	0.36	0.29	0.41	0.42	0.36
Qwen2.5-VL-32B	0.48	0.41	0.50	0.48	0.48	0.39	0.27	0.38	0.46	0.22
<b>+ CAFES (Ours)</b>	0.51	0.49	0.49	0.53	0.53	0.43	0.26	0.40	0.44	0.25
LLaMA-3.2-Vision-11B	0.28	0.24	0.27	0.25	0.27	0.24	0.17	0.26	0.26	0.21
<b>+ CAFES (Ours)</b>	0.35	0.34	0.36	0.39	0.34	0.32	0.16	0.35	0.32	0.26
LLaMA-3.2-Vision-90B	0.41	0.35	0.45	0.44	0.41	0.36	0.25	0.36	0.37	0.20
<b>+ CAFES (Ours)</b>	0.47	0.44	0.46	0.49	0.48	0.41	0.27	0.37	0.37	0.25
Claude-3.5-Sonnet	0.67	0.57	0.67	0.67	0.57	0.57	0.42	0.54	0.59	0.33
<b>+ CAFES (Ours)</b>	0.63	0.60	0.60	0.58	0.60	0.58	0.41	0.52	0.57	0.39
Gemini-2.5-Flash	0.57	0.57	0.62	0.59	0.58	0.51	0.35	0.44	0.54	0.36
<b>+ CAFES (Ours)</b>	0.58	0.56	0.59	0.55	0.55	0.50	0.33	0.46	0.52	0.37
GPT-4o-mini	0.57	0.40	0.54	0.64	0.50	0.50	0.38	0.55	0.50	0.29
<b>+ CAFES (Ours)</b>	0.53	0.52	0.53	0.59	0.55	0.49	0.37	0.46	0.49	0.32

Table 18: PCC scores of different student MLLMs on ten multi-granular essay traits. All values are reported after rounding to two decimal places.

MLLMs	Lexical Level		Sentence Level				Discourse Level			
	LA	LD	CH	GA	GD	PA	AC	JP	OS	EL
InternVL2.5-8B	0.35	0.23	0.33	0.31	0.18	0.30	0.24	0.40	0.19	0.23
+ CAFES (Ours)	0.42	0.33	0.36	0.49	0.3	0.41	0.25	0.39	0.34	0.27
Qwen2.5-VL-3B	0.29	0.42	0.38	0.31	0.34	0.24	0.23	0.38	0.39	0.31
+ CAFES (Ours)	0.35	0.42	0.38	0.46	0.34	0.35	0.25	0.34	0.41	0.31
Qwen2.5-VL-32B	0.47	0.40	0.49	0.48	0.48	0.40	0.24	0.37	0.46	0.17
+ CAFES (Ours)	0.50	0.47	0.46	0.52	0.54	0.44	0.23	0.37	0.44	0.19
LLaMA-3.2-Vision-11B	0.32	0.27	0.30	0.29	0.30	0.28	0.15	0.28	0.30	0.22
+ CAFES (Ours)	0.36	0.36	0.37	0.42	0.35	0.34	0.16	0.35	0.36	0.27
LLaMA-3.2-Vision-90B	0.41	0.36	0.43	0.45	0.41	0.37	0.22	0.33	0.34	0.16
+ CAFES (Ours)	0.45	0.44	0.42	0.48	0.48	0.41	0.21	0.32	0.35	0.22
Claude-3.5-Sonnet	0.65	0.55	0.66	0.65	0.54	0.58	0.37	0.53	0.59	0.25
+ CAFES (Ours)	0.64	0.58	0.59	0.58	0.58	0.58	0.34	0.49	0.58	0.32
Gemini-2.5-Flash	0.55	0.56	0.60	0.59	0.56	0.53	0.31	0.43	0.53	0.33
+ CAFES (Ours)	0.55	0.55	0.57	0.54	0.53	0.54	0.30	0.44	0.51	0.35
GPT-4o-mini	0.57	0.38	0.51	0.64	0.47	0.51	0.32	0.52	0.51	0.23
+ CAFES (Ours)	0.56	0.51	0.51	0.60	0.57	0.50	0.30	0.42	0.48	0.25

Table 19: SCC scores of different student MLLMs on ten multi-granular essay traits. All values are reported after rounding to two decimal places.

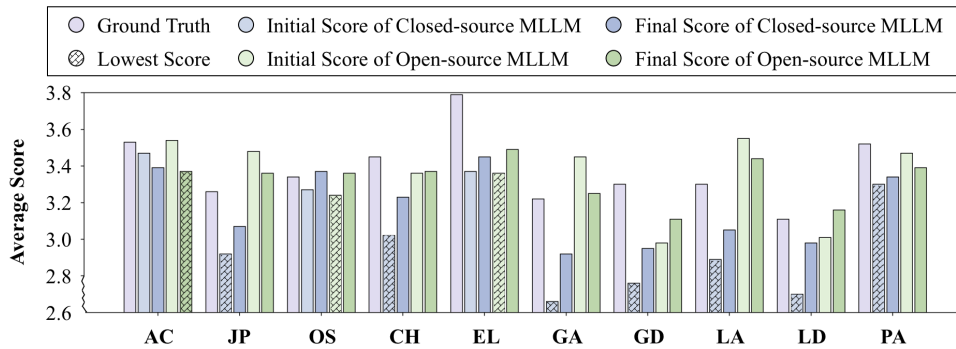


Figure 12: Average trait-specific scores assigned by closed-source and open-source MLLMs at both the initial stage and after revision through the CAFES framework.

MLLMs	Lexical Level		Sentence Level				Discourse Level			
	LA	LD	CH	GA	GD	PA	AC	JP	OS	EL
InternVL2.5-2B (Chen et al., 2025b)	0.04	0.06	0.07	0.02	0.08	0.05	0.03	0.01	0.05	0.04
+ CAFES (Ours)	<b>0.25</b>	<b>0.23</b>	<b>0.28</b>	<b>0.26</b>	<b>0.19</b>	<b>0.27</b>	<b>0.16</b>	<b>0.18</b>	<b>0.22</b>	<b>0.08</b>
Improvements	↑0.21	↑0.16	↑0.21	↑0.24	↑0.10	↑0.22	↑0.14	↑0.17	↑0.17	↑0.03

Table 20: QWK improvements across ten scoring traits when using Qwen2.5-VL-32B as the teacher and InternVL2.5-2B as the student.

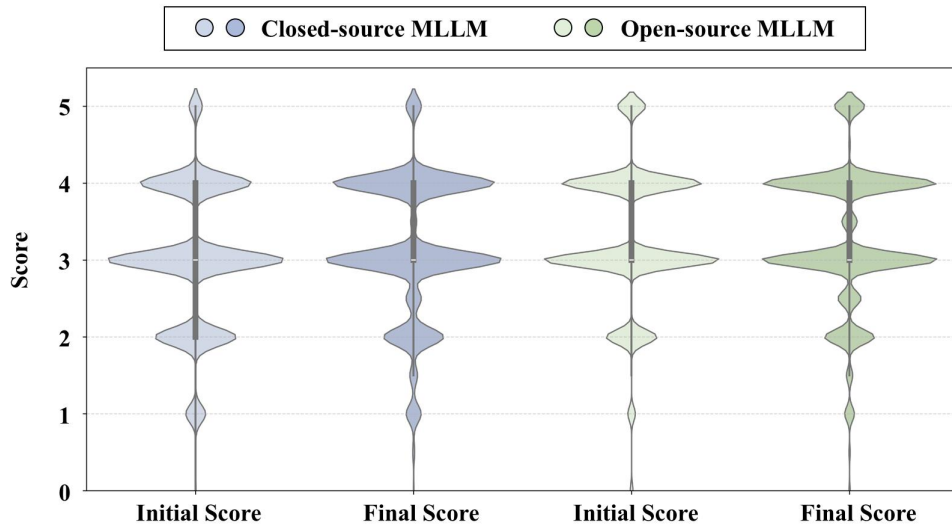
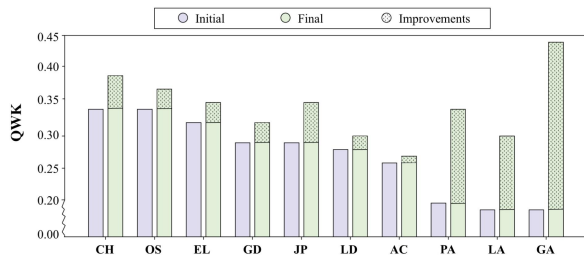
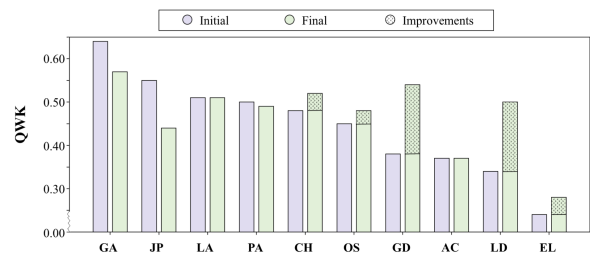


Figure 13: Score distributions of closed-source and open-source MLLMs at both the initial scoring stage and after revision through the CAFES framework.



(a) Qwen2.5-VL-3B



(b) GPT-4o-mini

Figure 14: Improvements of QWK score across all traits based on different student MLLM.

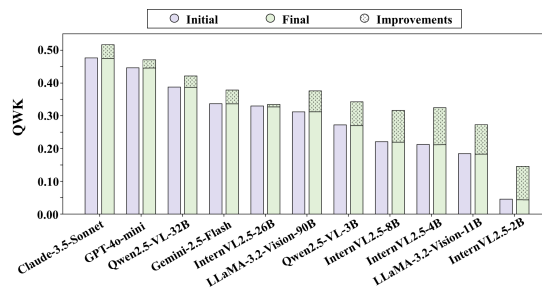


Figure 15: Improvements of average QWK score across ten traits of all student MLLMs.

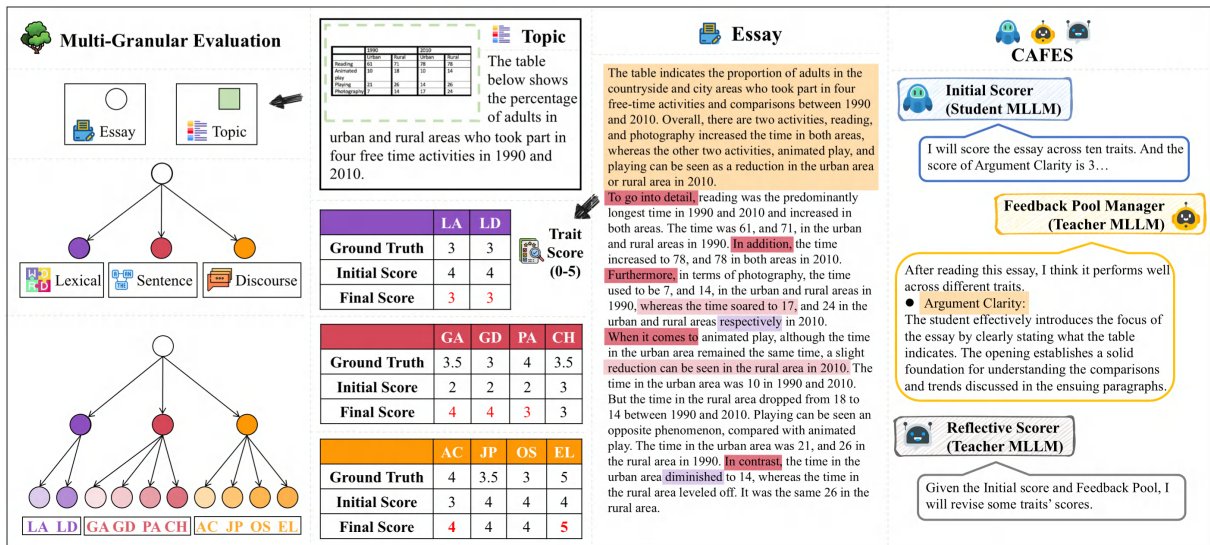


Figure 16: A case study illustrating CAFES's score revision process. And the student MLLM is Claude-3.5-Sonnet, and the teacher MLLM is GPT-4o.

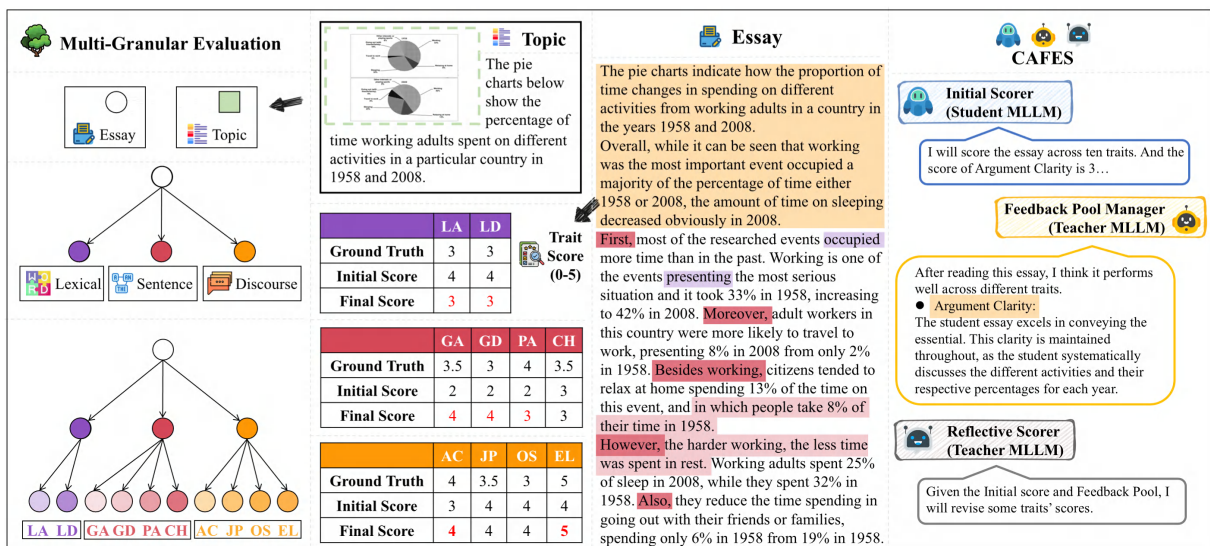


Figure 17: A case study illustrating CAFES's score revision process. And the student MLLM is GPT-4o-mini, and the teacher MLLM is GPT-4o.

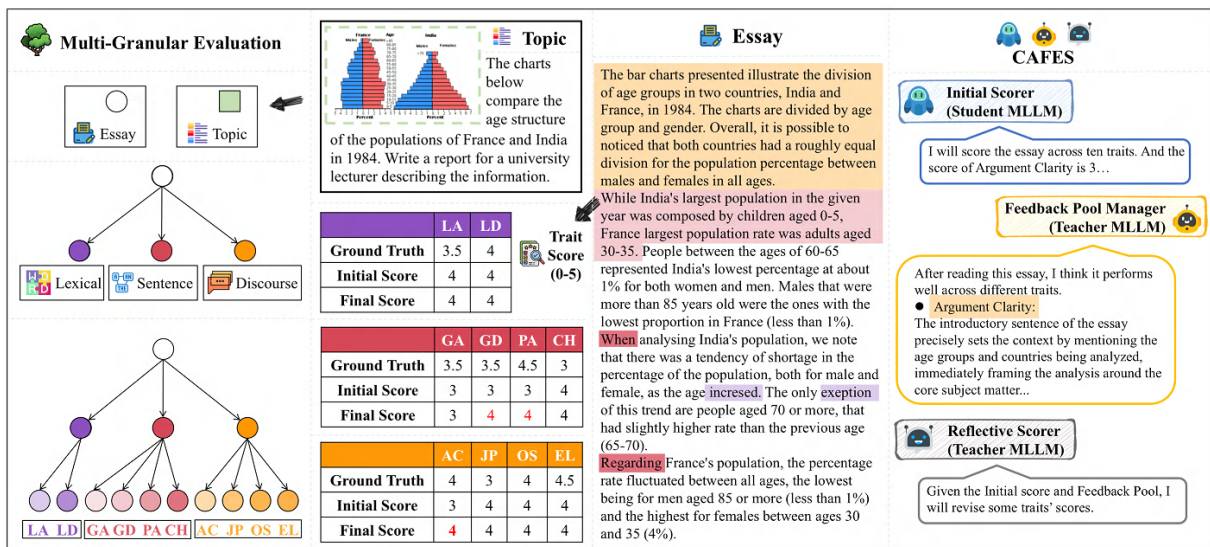


Figure 18: A case study illustrating CAFES 's score revision process. And the student MLLM is Qwen2.5-VL-32B, and the teacher MLLM is GPT-4o.

# GM-PRM: A Generative Multimodal Process Reward Model for Multimodal Mathematical Reasoning

Jianghangfan Zhang<sup>1</sup>, Yibo Yan<sup>1,2</sup>, Kening Zheng<sup>3</sup>, Xin Zou<sup>1</sup>, Song Dai<sup>1</sup>, Xuming Hu<sup>1,2\*</sup>

<sup>1</sup>Hong Kong University of Science and Technology (Guangzhou),

<sup>2</sup>Hong Kong University of Science and Technology

<sup>3</sup>University of Illinois Chicago

Correspondence: xuminghu@hkust-gz.edu.cn

## Abstract

Multimodal Large Language Models (MLLMs) demonstrate remarkable capabilities but often struggle with complex, multi-step mathematical reasoning, where minor errors in visual perception or logical deduction can lead to complete failure. While Process Reward Models (PRMs) offer step-by-step supervision, existing multimodal PRMs are limited to being binary verifiers that can identify but not correct errors, offering little explanatory power. To address these deficiencies, we introduce the **Generative Multimodal Process Reward Model (GM-PRM)**, a novel paradigm that transforms the PRM from a passive judge into an active reasoning collaborator. Instead of a simple scalar score, GM-PRM provides a fine-grained, interpretable analysis of each reasoning step, evaluating its step intent, visual alignment, and logical soundness. More critically, GM-PRM is trained to generate a corrected version of the first erroneous step it identifies. This unique corrective capability enables our new test-time inference strategy, Refined Best-of-N (Refined-BoN). This framework actively enhances solution quality by using the PRM’s generated correction to guide the policy model toward a more promising reasoning trajectory, thereby improving the diversity and correctness of the solution pool. We demonstrate that GM-PRM achieves state-of-the-art results on multiple multimodal math benchmarks, significantly boosting policy model performance with remarkable data efficiency, requiring only a 20K-sample training dataset. Our code will be released upon acceptance.

## 1 Introduction

The advent of Multimodal Large Language Models (MLLMs) has marked a significant milestone in artificial intelligence, demonstrating remarkable capabilities in integrating and understanding visual and textual information (Caffagni et al., 2024;

\*Corresponding author

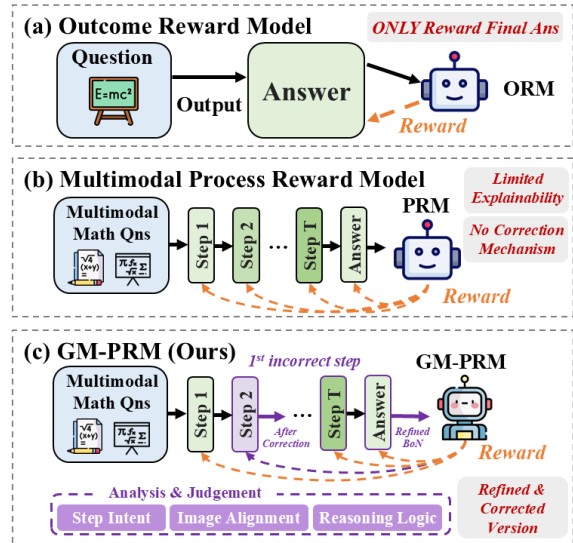


Figure 1: Comparison among ORM (a), PRM (b), and our proposed GM-PRM (c) for multimodal math reasoning.

Yan et al., 2024c; Yan and Lee, 2024; Huo et al., 2024; Zheng et al., 2024b). While these models excel at general-purpose tasks such as image captioning and visual question answering, they often falter when confronted with complex, multi-step reasoning challenges, particularly within specialized domains like mathematics (Wang et al., 2024; Yan et al., 2024a, 2025a; Ahn et al., 2024). Solving multimodal mathematical problems requires not only accurate perception of visual elements (e.g., geometric figures, function graphs) but also a rigorous, step-by-step logical deduction process (Shi et al., 2024; Zhuang et al., 2025; Yan et al., 2025b). *Minor errors in either image interpretation or logical inference can cascade, leading to entirely incorrect final answers.*

To mitigate these reasoning deficiencies, Process Reward Models (PRMs) have emerged as a promising paradigm (Gao et al., 2024; Zhong et al., 2025). Unlike outcome-based models that only re-

ward correct final answers (shown in Figure 1 (a)), PRMs provide fine-grained supervision by evaluating the correctness of each intermediate step in a reasoning chain (Zheng et al., 2024a; Lambert et al., 2024; Yan et al., 2024b), as shown in Figure 1 (b). This approach has proven effective in the language domain (Zeng et al., 2025; Yuan et al., 2024; Zhang et al., 2025a). However, extending PRMs to the multimodal context presents unique challenges (Miao et al., 2025; Du et al., 2025; Li et al., 2025b; Cao et al., 2025). Existing multimodal PRMs often function as binary classifiers, assigning a simple correct/incorrect label to each step, which offers *limited explanatory power*. Furthermore, they typically possess the ability to identify errors but *lack the mechanism to correct them*, leaving the reasoning process fundamentally broken. This limitation constrains their utility, especially within mechanisms like Best-of-N (BoN) sampling, which remain passive selection processes over a static set of potentially flawed solutions.

In this work, we introduce a novel **Generative Multimodal Process Reward Model (GM-PRM)** to address these limitations, as illustrated in Figure 1 (c). Our model transcends the role of a simple verifier and acts as an active reasoning collaborator. Instead of merely outputting a scalar score, our GM-PRM leverages its generative capabilities to produce a detailed, interpretable analysis for each reasoning step. This analysis deconstructs the step into three critical aspects: its fundamental **step intent**, the correctness of its **image alignment**, and the soundness of its **reasoning logic**. More importantly, our model is trained not only to identify errors but also to **generate a refined, corrected version** of the first incorrect step it encounters.

This unique corrective capability enables us to propose a new test-time inference strategy: the **Refined Best-of-N (Refined-BoN)** process. This dynamic framework moves beyond passive selection by actively enhancing the quality of candidate solutions. When our GM-PRM identifies a flawed step within a generated solution, it intervenes by providing a corrected step, which is then used to guide the policy model in generating a new, more promising reasoning trajectory. This iterative refinement process significantly improves the diversity and correctness of the solution pool, leading to a substantial boost in the policy model’s problem-solving performance. Furthermore, we demonstrate that this powerful capability can be achieved with remarkable data efficiency, requiring a significantly

smaller training dataset than previous approaches. Our primary contributions are as follows:

- We develop a generative multimodal PRM that **provides fine-grained, interpretable feedback for mathematical reasoning**. It analyzes each step’s purpose, image alignment, and logical validity, moving beyond simple binary classification to offer deeper insight into the model’s thought process.
- We introduce a novel Refined-BoN framework that **leverages the PRM’s generative power to actively correct errors at test time**. It enhances the policy model’s ability to find correct solutions by iteratively improving flawed reasoning paths.
- We demonstrate the **effectiveness and data efficiency** of GM-PRM, achieving state-of-the-art results on **multiple multimodal math benchmarks**. Our approach requires only a 20K sample dataset, highlighting the quality of data curation and the power of generative supervision strategy.

## 2 Related Work

### 2.1 Process Reward Models (PRMs)

PRMs have been proposed to evaluate the fine-grained step level for model reasoning. During the implementation process, annotating and obtaining a high-quality training dataset incurs a high cost. PRM800K (Lightman et al., 2023) is the first process supervision dataset completely annotated by humans. To mitigate annotation costs, MathShepherd (Wang et al., 2023) proposes Monte Carlo (MC) estimation, while OmegaPRM (Luo et al., 2024) leverages Monte Carlo Tree Search (MCTS) to automatically evaluate each reasoning step, both utilizing the generation capabilities of Large Language Models (LLMs). Subsequent research has enhanced the effectiveness of PRMs through various methods, including VerSaPRM (Zeng et al., 2025), Implicit PRM (Yuan et al., 2024), OpenPRM (Zhang et al., 2025a), PQM (Li and Li, 2024), PAV (Setlur et al., 2024), and others. Furthermore, GenRM (Zhao et al., 2025), ThinkPRM (Khalifa et al., 2025) and R-PRM (She et al., 2025) extend the method of using model generation analysis to evaluate steps to PRMs. Recently, DeepSeek-GRM (Liu et al., 2025) analyzes responses from multiple perspectives and directly generates and aggregates the

scores to achieve the estimation of the entire reasoning process. There are also many studies on benchmarks of PRMs such as ProcessBench (Zheng et al., 2024a), PRMBench (Song et al., 2025), and Socratic-PRMBench (Li et al., 2025a).

## 2.2 Multimodal PRMs

After achieving certain results and progress in the research of language modality in PRMs, research on PRMs has also begun to shift towards multimodal tasks. M-STAR (Liu et al., 2024a) proposes and implements multimodal PRM on multimodal problems. URSA (Luo et al., 2025) constructs a dataset by inserting errors and utilizes it to train a multimodal PRM. VisualPRM (Wang et al., 2025b) not only uses MC estimation to construct a multimodal VisualPRM400K training dataset, but also proposes a benchmark for multimodal PRMs called VisualProcessBench, which is entirely annotated by humans. Moreover, Athena-PRM (Wang et al., 2025a), PRM-BAS (Hu et al., 2025), MM-PRM (Du et al., 2025) and DreamPRM (Cao et al., 2025) also improve the capability of multimodal PRMs. Although several studies have explored multimodal PRMs, applying them to multimodal tasks effectively still poses certain challenges, such as insufficient interpretability of the labels assigned to each reasoning step and the inability to correct identified erroneous steps. In our work, we introduce a generative multimodal PRM, GM-PRM to solve the above problems.

## 3 Methodology

In this section, we first describe how to utilize PRMs and generative PRMs combined with the BoN method to improve the performance of policy models for mathematical problems in Section 3.1. Then, we introduce our process to implement multimodal generative PRM, including data construction and model training in Section 3.2. Finally, we propose a novel Refined-BoN framework for PRMs to enhance its performance beyond traditional BoN method in Section 3.3.

### 3.1 PRMs for Mathematical Problem

In this section, we present the implementation methods of PRM and GM-PRM, and provide formal and detailed explanations of their usage through mathematical notation.

#### 3.1.1 Problem and Reasoning Steps Generation

Let  $Q$  denote a mathematical problem. Firstly, an LLM  $\pi$  is involved in solving the mathematical problem  $Q$ . To facilitate reasoning, the problem is combined with a prompt  $P$ , which includes specific instructions guiding the generation of a step-by-step reasoning process and a final answer. This composite input is then fed into the LLM. When generating a response,  $\pi$  generates a sequence of reasoning steps, denoted as  $R = \{r_1, r_2, \dots, r_T\}$ , where  $T$  represents the total number of reasoning steps to the given mathematical problem. The above process can be explained as follows:

$$R = \pi(Q \parallel P), \quad (1)$$

where  $\parallel$  denotes the concatenation of the problem  $Q$  and the prompt  $P$ , and  $\pi(\cdot)$  represents the inference process of LLM.

#### 3.1.2 PRM

A single instance in a training dataset  $\mathcal{D}$  to train a PRM comprises three components: (1) a problem statement, (2) a generated response consisting of multiple inference steps, and (3) a corresponding set of binary labels, each taking a value of either 0 or 1, indicating whether the associated reasoning step is incorrect or correct, respectively.

During training, the PRM is optimized using cross-entropy loss and supervised to align its predictions with the ground-truth labels. After being trained, the PRM model is capable of processing new reasoning steps generated by the LLM in response to a given mathematical problem, which means that the PRM is able to assign a scalar score to each individual reasoning step, reflecting the model’s confidence in the correctness of each step:

$$f_{\text{PRM}} : (Q, R) \mapsto (s_1, s_2, \dots, s_T), \quad (2)$$

where  $f_{\text{PRM}} : (\cdot)$  represents the inference of PRM,  $s_i \in [0, 1]$  denotes the confidence score assigned to the  $i$ -th reasoning step  $r_i$ , and  $T$  denotes the number of reasoning steps.

For generative PRM, the binary labels in the training dataset are replaced with textual analyses and judgments, each formulated as a textual choice such as “incorrect” or “correct”. During inference, generative PRM also generates textual critiques and judgments for each step.

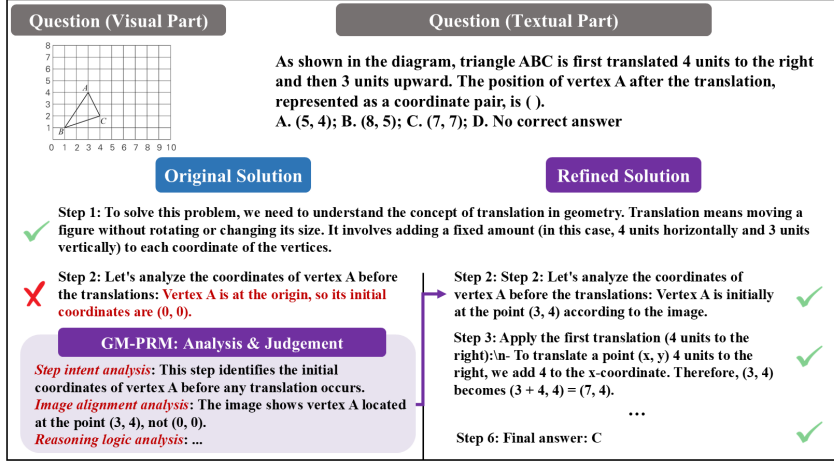


Figure 2: The illustration of a representative example *before* and *after* applying GM-PRM. In particular, GM-PRM first judges the steps of the original solution generated by the policy model. Subsequently, GM-PRM finds that the second step is incorrect and refines the second step to generate the correct version. The correct steps are input to the policy model to generate the refined solution, and finally the correct answer is obtained.

### 3.1.3 GM-PRM

By extending generative PRMs from the textual modality to a multimodal setting, we introduce GM-PRM. In this setting, mathematical problems are represented using both textual and visual information. The input to the policy model comprises the image of the problem, its textual description and task-specific instructions, which are processed jointly to generate reasoning steps. Similarly, during both training and inference, it is essential to provide GM-PRM with inputs from both visual and textual modalities, enabling it to perform cross-modal analysis when assigning correctness labels to each reasoning step:

$$f_{\text{GM-PRM}} : (Q, I, R) \mapsto (c_1, j_1, \dots, c_T, j_T), \quad (3)$$

where  $f_{\text{GM-PRM}} : (\cdot)$  represents the inference of GM-PRM,  $I$  denotes the image of the mathematical problem,  $c_i$  denotes the critique of the  $i$ -th reasoning step  $r_i$ , and  $j_i$  denotes the textual judgment assigned to the  $i$ -th reasoning step  $r_i$ .

## 3.2 Data Construction

In this section, we present our methodology employed to construct the training data for GM-PRM. The process consists of three key stages: (1) the selection of appropriate types and quantities of question data from the VisualPRM400K dataset (Wang et al., 2025b); (2) the generation of textual analysis and judgment data using GPT-4o; and (3) the filtering process of the generated data through MC estimation and LLM-as-a-judge techniques to ensure quality and reliability.

### 3.2.1 Data Selection

VisualPRM400K is a large-scale dataset containing approximately 400,000 multimodal process supervision samples. In our work, we select plane geometry- and function-related problems from VisualPRM400K to construct a specialized subset and supplement it with corresponding textual analysis for training GM-PRM. This targeted subset with textual critiques supports the effective training of GM-PRM, yielding strong performance on geometric and function-based mathematical reasoning tasks.

### 3.2.2 Generation of Analysis and Judgment

To obtain textual analyses and judgments, we employ GPT-4o to critique each reasoning step from 4 key aspects: **step intent**, **image alignment**, **reasoning logic**, and **step refinement**.

The aspect of **step intent** indicates identifying the purpose of each reasoning step. This initial analysis establishes a foundation that allows GM-PRM to interpret and evaluate each reasoning step in context more effectively. Furthermore, this level of understanding facilitates subsequent error detection and correction tasks, thereby enhancing the overall effectiveness of GM-PRM.

The second aspect is **image alignment**. When MLLMs are used for inference in solving multimodal problems, MLLMs often make errors in image alignment, such as misidentifying parallel relationships or incorrectly annotating angles, which leads to flawed solutions. To address this, we employ GPT-4o to produce textual analysis and judg-

ments in image alignment for inference steps, to form the dataset for training GM-PRM.

**Reasoning logic** is an indispensable presence in the step-by-step problem-solving process of MLLMs. However, the occurrences of logical inconsistencies and errors, such as miscalculations and incorrect inferences significantly impact the correctness of the reasoning steps and the final answers. Therefore, it is crucial for GM-PRM to be capable of identifying such logical flaws and making accurate judgments regarding the validity of the reasoning logic for each step. In our work, we employ GPT-4o to generate textual analysis and judgments of each step in reasoning logic to form the training dataset. The above process can be formulated as follows:

$$\mathcal{F} : (Q, I, R \parallel P) \mapsto \{SI_i, IA_i, RL_i, FJ_i\}_{i=1}^t, \quad (4)$$

where  $\mathcal{F} : (\cdot)$  represents the inference of GPT-4o,  $SI_i$  denotes the textual analysis of step intent for the  $i$ -th reasoning step,  $t$  denotes the number of the first incorrect step or the last step,  $1 \leq t \leq T$ ,  $IA_i = \{IAC_i, IAJ_i\}$  is the analysis which contains critique  $IAC_i$  and judgment  $IAJ_i$  in image alignment of the  $i$ -th reasoning step,  $RL_i = \{RLC_i, RLJ_i\}$  denotes the analysis which contains critique  $RLC_i$  and judgment  $RLJ_i$  in reasoning logic of the  $i$ -th reasoning step,  $FJ_i$  denotes the final judgment of the  $i$ -th reasoning step.

Building on aforementioned three aspects, we further aim for GM-PRM to **correct the first identified erroneous step**. The above information enables GM-PRM to generate corrected reasoning steps that are logically coherent, visually accurate, and semantically aligned with the original step intent. The resulting corrected steps can then be used to construct more diverse and accurate inference solutions and ultimately produce more reliable final answers. In our work, we employ GPT-4o to generate a corrected version of the first identified error step in a reasoning process if the first error step is detected to exist by GPT-4o:

$$\mathcal{F} : (Q, I, R \parallel P) \mapsto \begin{cases} RS, & \text{if incorrect step exists,} \\ \emptyset, & \text{otherwise.} \end{cases} \quad (5)$$

where  $RS$  denotes refined step of the first error step in a reasoning process.

In summary, we design a structured prompt for GPT-4o to generate comprehensive analysis data across four dimensions based on the provided prob-

lems, associated images, and step-by-step solutions:

$$\mathcal{F} : (Q, I, R \parallel P) \mapsto \mathcal{D}, \quad (6)$$

where  $\mathcal{D}$  denotes the generated training dataset:

$$\mathcal{D} = \{(\{SI_i^k, IA_i^k, RL_i^k, FJ_i^k\}_{i=1}^t, RS^k)\}_{k=1}^K, \quad (7)$$

where  $k \in \{1, 2, \dots, K\}$  represents the  $k$ -th sample in the dataset, and  $K$  denotes the number of the training instances.

### 3.2.3 Data Filtering

The process of constructing training data using GPT-4o can be regarded as an implementation of LLM-as-a-judge methodology. Inspired by the combination of LLM-as-a-judge and MC estimation techniques (Zhang et al., 2025b), we employ the MC estimation technique proposed by Math-Shepherd (Wang et al., 2023) to effectively filter and curate the generated data.

Monte Carlo estimation is a strategy for automated annotation that leverages LLMs or MLLMs to generate multiple subsequent solutions for each step. When applying MC estimation to evaluate a step  $r_i$ , we use an LLM or an MLLM as a ‘completer’ to finalize multiple subsequent reasoning processes from this step:

$$f_{completer} \mapsto \{(r_{i+1}^j, \dots, r_{L_j}^j, a^j)\}_{j=1}^m, \quad (8)$$

where  $a^j$  is the final answer of the  $j$ -th finalized solution and  $L_j$  is the total number of steps.

Within MC estimation, one type of evaluation method is commonly applied: hard estimation. In hard estimation, a step  $r_i$  is deemed correct if at least one subsequent solution reaches the correct final answer  $a^*$ ; otherwise, it is incorrect:

$$l_i^{HE} = \begin{cases} 1, & \exists a_j, a_j = a^*, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

In our data construction process, we employ hard estimation to label the correctness of individual reasoning steps. By integrating LLM-as-a-judge technique and MC estimation, we compare the labels acquired by MC estimation and judgments generated by GPT-4o. Data samples that receive consistent evaluations from both methods are selected as our final training dataset. By integrating these two methods, we aim to further enhance the reliability and quality of the training data, ensuring better performance of GM-PRM.

### 3.3 Refined-BoN Process

When applying Test-time Scaling (TTS) for LLMs and MLLMs, a widely adopted method is Best-of-N (BoN) approach. In the BoN process, a policy model is employed to generate  $N$  candidate solutions, which are then evaluated by reward models or self-consistency to select the optimal solution. However, during the BoN process, policy models are under identical prompting conditions when generating multiple solutions, which leads to the problem that the solutions often lack diversity and may exhibit limited correctness. In our work, we propose a novel Refined-BoN framework utilizing TTS techniques to enhance the diversity and accuracy of generated solutions, thereby improving the reasoning capabilities of policy models.

#### 3.3.1 Refined-BoN Method

As shown in Figure 2, in Refined-BoN process, we first employ an MLLM as the policy model to generate  $N/2$  initial solutions to a multimodal problem, and then these solutions are evaluated step-by-step by GM-PRM. For the subsequent  $N/2$  solutions, the policy model generates them under varying conditions, informed by the evaluation of the preceding  $N/2$  solutions: If GM-PRM identifies an incorrect reasoning step within a solution, it stops evaluating and refines the first erroneous step by generating a corrected version. This corrected step, along with all previously validated correct steps, is then input back into the policy model to continue the solution generation process. Conversely, if GM-PRM determines that all steps in a particular solution are correct, we employ the policy model to generate a new solution using the same prompt. Through this regeneration mechanism, we obtain the additional  $N/2$  solutions. Then, we employ GM-PRM to evaluate the subsequent  $N/2$  solutions.

#### 3.3.2 Solution Selection

After applying the Refined-BoN process, we obtain  $N$  solutions for each problem, each accompanied by step-level correctness judgments. Moreover, we divide all the solutions into two categories: one where GM-PRM judges that it contains incorrect steps, and the other where GM-PRM judges that all its steps are correct. Furthermore, we take the corresponding probability of GM-PRM generating the associated “Correct” and “Incorrect” tokens as the score of each step.

Among the  $N$  generated solutions, if there exist solutions in which all reasoning steps are judged

correct, we calculate the average of the scores of all steps in these solutions as the overall score of the solution, and select the solution with the highest average score as the optimal solution.

For  $N$  solutions to the problem, if GM-PRM determines that all  $N$  solutions contain incorrect steps, we calculate the average score of all steps in each solution as the overall score of the solution, and select the solution with the highest overall score as the final answer.

## 4 Experiments

In this section, we introduce our experimental setup to assess GM-PRM under the Refined-BoN process on five multimodal mathematical benchmarks in Section 4.1. In addition, we present the results of our experiments and three conclusions analyzed from the results in Section 4.2. Finally, we show the ablation studies in Section 4.3.

### 4.1 Experimental Setup

#### 4.1.1 Benchmarks

We evaluate GM-PRM across five datasets, including MathVista (Lu et al., 2023), MathVision (Wang et al., 2024), MathVerse (Zhang et al., 2024), DynaMath (Zou et al., 2024) and WeMath (Qiao et al., 2024). The datasets contain diverse problem types, such as plane geometry, functions, puzzle tests, *etc.* We use Vision-Only subset of MathVerse dataset and Plane-Geometry subset of DynaMath.

#### 4.1.2 Settings

We employ GM-PRM as the critic model for Refined-BoN evaluation and set  $N$  to 8 by default. We select six models as the policy models to generate step-by-step reasoning processes. When reasoning, we set the temperature of the policy models to 0.7 and top-p to 0.9. For comparison, we use the average accuracy of  $N$  sets of answers generated by policy models as baselines.

#### 4.1.3 Training Details

To train GM-PRM, we use Qwen2.5-VL-7B-Instruct as our base model and perform supervised fine-tuning (SFT) with all parameters trainable but the frozen Vision Transformer (ViT) encoder. During the training process, we utilize bfloat16 mixed-precision and DeepSpeed with zero3 technology and set the training consists of 2 epochs. For batch size, the batch size on each training device is set to 2, and through gradient accumulation, the effective batch size is extended to 16. Moreover, we use two

MLLMs	MathVista	MathVision	MathVerse	DynaMath	WeMath	Average
MiniCPM-V2.6-8B	44.3	16.0	18.9	22.6	38.6	28.1
+ GM-PRM (Ours)	<b>51.0</b>	<b>18.1</b>	<b>24.4</b>	<b>25.7</b>	<b>51.0</b>	<b>34.0</b>
Improvements	<u>+6.7</u>	<u>+2.1</u>	<u>+5.5</u>	<u>+3.1</u>	<u>+12.4</u>	<u>+5.9</u>
Llama-3.2-11B-Vision	44.5	14.3	16.5	28.4	46.1	30.0
+ GM-PRM (Ours)	<b>49.5</b>	<b>18.2</b>	<b>18.8</b>	<b>32.7</b>	<b>53.4</b>	<b>34.5</b>
Improvements	<u>+5.0</u>	<u>+3.9</u>	<u>+2.3</u>	<u>+4.3</u>	<u>+7.3</u>	<u>+4.5</u>
Qwen2.5-VL-7B	63.2	25.1	32.8	35.0	60.6	43.3
+ GM-PRM (Ours)	<b>65.0</b>	<b>28.2</b>	<b>37.4</b>	<b>39.2</b>	<b>69.0</b>	<b>47.8</b>
Improvements	<u>+1.8</u>	<u>+3.1</u>	<u>+4.6</u>	<u>+4.2</u>	<u>+8.4</u>	<u>+4.5</u>
InternVL3-8B	50.6	20.3	25.0	27.0	50.9	34.8
+ GM-PRM (Ours)	<b>55.7</b>	<b>22.2</b>	<b>31.7</b>	<b>33.4</b>	<b>59.2</b>	<b>40.4</b>
Improvements	<u>+5.1</u>	<u>+1.9</u>	<u>+6.7</u>	<u>+6.4</u>	<u>+8.3</u>	<u>+5.6</u>
InternVL3-38B	68.3	34.9	37.8	40.1	66.4	49.5
+ GM-PRM (Ours)	<b>69.9</b>	<b>37.0</b>	<b>39.1</b>	<b>43.1</b>	<b>72.9</b>	<b>52.4</b>
Improvements	<u>+1.6</u>	<u>+2.1</u>	<u>+1.3</u>	<u>+3.0</u>	<u>+6.5</u>	<u>+2.9</u>
InternVL3-78B	68.0	34.6	36.0	38.1	65.7	48.5
+ GM-PRM (Ours)	<b>70.7</b>	<b>37.1</b>	<b>40.6</b>	<b>39.9</b>	<b>72.2</b>	<b>52.1</b>
Improvements	<u>+2.7</u>	<u>+2.5</u>	<u>+4.6</u>	<u>+1.8</u>	<u>+6.5</u>	<u>+3.6</u>

Table 1: Percentage accuracy scores (%) of multiple MLLMs across five datasets. For each MLLM, the first row shows the baseline, the second shows the final result with GM-PRM, and the third shows the improvement. Only positive improvements are underlined. The best results are highlighted in **bold**. All values are reported after rounding to one decimal place.

A800 GPUs to train GM-PRM, and the AdamW optimizer is used with an initial learning rate of  $1 \times 10^{-5}$ . The learning rate schedule involves a linear warm-up with the warm-up ratio equal to 0.05 followed by linear decay.

## 4.2 Main Results

As shown in Table 1, integrating GM-PRM with the Refined-BoN process consistently improves performance across five different benchmark datasets for six different MLLMs. On average, our method yields notable accuracy gains, with improvements of +5.9 for MiniCPM-V2.6-8B, +4.5 for Llama-3.2-11B-Vision, +4.5 for Qwen2.5-VL-7B, and +5.6 for InternVL3-8B.

A closer look at dataset-level results reveals that the improvements are not uniform. The WeMath benchmark shows the most significant enhancement, with MiniCPM-V2.6-8B improving by +12.4 points, highlighting the ability of our method to strengthen mathematical reasoning on challenging problems. Similarly, MathVerse and DynaMath exhibit consistent gains of +4.5–6.7 points across multiple models, suggesting that our approach particularly benefits datasets requiring complex symbolic manipulation and multi-step reasoning. In contrast, MathVision improvements are more modest (+1.9–3.9), indicating that the visual reasoning component may already be relatively strong in baseline models.

**GM-PRM combined with the Refined-BoN process demonstrates strong generalization across diverse multimodal mathematical problems, with particularly remarkable gains in plane geometry tasks.** As illustrated in Figure 4, even after excluding plane geometry and function problems, policy models still achieve notable improvements across the datasets. This indicates that although GM-PRM is primarily trained on a dataset composed of plane geometry and function problems, it generalizes effectively to other types of multimodal mathematical problems. Moreover, as shown by the averaged results in Figure 4, the improvements achieved by GM-PRM with Refined-BoN on plane geometry problems consistently exceed those on the overall dataset, function problems, and other categories, underscoring the exceptional effectiveness of our method in tackling plane geometry tasks.

**The Refined-BoN process yields disproportionately larger gains for models with lower baseline performance.** As shown in Table 1, InternVL3-38B starts with the highest initial average accuracy among all policy models (49.5%) and achieves a modest improvement of +2.9 points (+5.9%). In contrast, Qwen2.5-VL-7B, which has the highest baseline accuracy (43.3%) among models with fewer than 12 billion parameters, improves by +4.5 points (+10.4%), surpassing the relative gains of InternVL3-38B. Notably, MiniCPM-V-2.6-

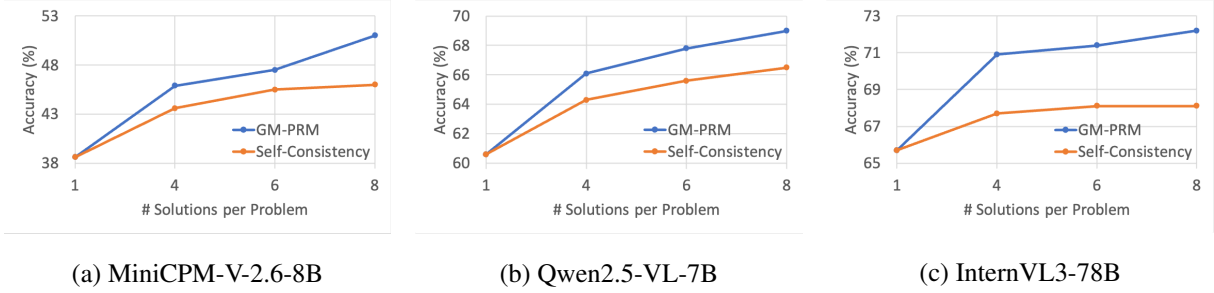


Figure 3: The results of changing the value of  $N$  in the Refined-BoN process on the WeMath across different policies. As  $N$  increases, the effectiveness of GM-PRM in enhancing accuracy improves and surpasses that of Self-Consistency.

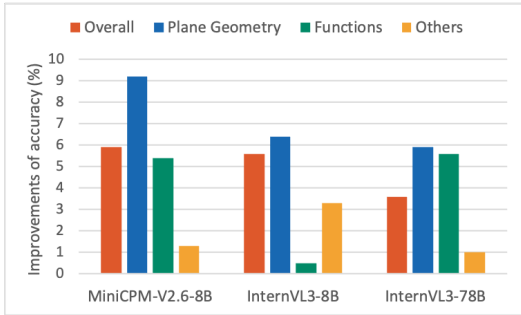


Figure 4: Improvements of the average percentage accuracy (%) of multiple MLLMs across different question types in MathVista, MathVision and MathVerse.

8B demonstrates the most significant relative improvement, achieving +5.9 points (+21.0%), despite its lower initial score. These results suggest that models with weaker baseline performance benefit more from the refinement mechanism of GM-PRM with Refined-BoN, likely because the process effectively corrects errors in reasoning steps, leaving greater room for improvement.

### 4.3 Number of solution samples $N$ in Refined-BoN

Following Test-time Scaling technique, we vary the number of  $N$  in the Refined-BoN process to evaluate the performance of GM-PRM in comparison to the Self-Consistency baseline.

Figure 3 depicts WeMath accuracy as the number of sampled solutions per problem ( $N$ ) increases from 1 to 8. Across all three backbones—MiniCPM-V-2.6-8B, Qwen2.5-VL-7B, and InternVL3-78B—both GM-PRM and the Self-Consistency (SC) baseline benefit from a larger sampling budget, yet GM-PRM exhibits a noticeably steeper growth curve.

Under the widely adopted Best-of-8 setting, GM-PRM delivers gains of 4.9 and 3.5 over SC on

MiniCPM-V-2.6-8B and Qwen2.5-VL-7B, respectively. Even for the 78B-parameter InternVL3, GM-PRM maintains a substantial 4.1 margin. These results indicate that the proposed refinement strategy in this paper not only scales to larger models but also converts additional candidate solutions into accuracy more effectively than Self-Consistency, thereby underscoring the robustness and versatility of GM-PRM.

Furthermore, for MiniCPM-V-2.6-8B, GM-PRM surpasses the self-consistency baseline by 2.1, 2.2, and 4.9 points under the Best-of-4, Best-of-6, and Best-of-8 settings, respectively, indicating a steadily increasing performance gap between GM-PRM and self-consistency as  $N$  increases.

## 5 Conclusion

In this work, we introduced GM-PRM, a novel paradigm that transforms the reward model from a passive judge into an active reasoning collaborator for multimodal mathematics problem solving. By providing fine-grained, interpretable analysis and, more critically, generating step-level corrections for erroneous steps, GM-PRM moves beyond simple binary verification. This unique corrective capability powers our Refined Best-of- $N$  (Refined-BoN) framework, which actively improves flawed reasoning trajectories during inference at test time. Our experiments demonstrate that this approach achieves state-of-the-art results on multiple benchmarks, significantly boosting policy model performance with remarkable data efficiency. The consistent gains across diverse models and problem types underscore the robustness and generalizability of our method. This shift from passive error detection to generative, collaborative correction represents a crucial advance in multimodal reasoning.

## Limitations

Our approach has several limitations that are important to note. First, GM-PRM is trained on a curated subset of VisualPRM400K focusing primarily on plane geometry and function-related problems. Although we observe encouraging transfer to other categories (e.g., Figure 4), performance may degrade under larger domain shifts such as puzzle-style questions, diagram-heavy word problems, or open-ended proof-like reasoning that requires global restructuring rather than local step repair. Second, our training signals depend on GPT-4o-generated analyses/refinements and are further filtered using hard Monte Carlo (MC) estimation; both components can introduce systematic label noise and biases (e.g., overconfidence on visually ambiguous cues or preference for certain solution styles), which may be inherited by GM-PRM. Finally, our refinement mechanism corrects only the first detected erroneous step; when errors are entangled across multiple steps, or when fixing one step necessitates revising earlier assumptions or downstream derivations, the refinement may be insufficient and can occasionally yield plausible but incorrect corrections. In addition, we freeze the ViT encoder during SFT, which may limit gains on inputs where stronger low-level visual perception is the primary bottleneck.

## Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grant No.62506318); Guangdong Provincial Department of Education Project (Grant No.2024KQNCX028); CAAI-Ant Group Research Fund; Scientific Research Projects for the Higher-educational Institutions (Grant No.2024312096), Education Bureau of Guangzhou Municipality; Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2025A03J3957), Education Bureau of Guangzhou Municipality.

## References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu,

Keming Lu, and 5 others. 2023. [Qwen technical report](#). *Preprint*, arXiv:2309.16609.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 4 others. 2025. [Qwen2.5-vl technical report](#). *Preprint*, arXiv:2502.13923.

Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. 2024. The revolution of multimodal large language models: a survey. *arXiv preprint arXiv:2402.12451*.

Qi Cao, Ruiyi Wang, Ruiyi Zhang, Sai Ashish Somayajula, and Pengtao Xie. 2025. Dreamprm: Domain-reweighted process reward model for multimodal reasoning. *arXiv preprint arXiv:2505.20241*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024. [Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks](#). *Preprint*, arXiv:2312.14238.

Lingxiao Du, Fanqing Meng, Zongkai Liu, Zhixiang Zhou, Ping Luo, Qiaosheng Zhang, and Wenqi Shao. 2025. Mm-prm: Enhancing multimodal mathematical reasoning with scalable step-level supervision. *arXiv preprint arXiv:2505.13427*.

Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. 2024. On designing effective rl reward at training time for llm reasoning. *arXiv preprint arXiv:2410.15115*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, and 1 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.

Pengfei Hu, Zhenrong Zhang, Qikai Chang, Shuhang Liu, Jiefeng Ma, Jun Du, Jianshu Zhang, Quan Liu, Jianqing Gao, Feng Ma, and 1 others. 2025. Prm-bas: Enhancing multimodal reasoning through prm-guided beam annealing search. *arXiv preprint arXiv:2504.10222*.

Jiahao Huo, Yibo Yan, Boren Hu, Yutao Yue, and Xuming Hu. 2024. Mmneuron: Discovering neuron-level domain-specific interpretation in multimodal large language model. *arXiv preprint arXiv:2406.11193*.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. [Process reward models that think](#). *Preprint*, arXiv:2504.16828.

- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, L Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, and 1 others. 2024. Rewardbench: Evaluating reward models for language modeling. arxiv 2024. *arXiv preprint arXiv:2403.13787*.
- Qingyun Li, Zhe Chen, Weiyun Wang, Wenhai Wang, Shenglong Ye, Zhenjiang Jin, Guanzhou Chen, Yinan He, Zhangwei Gao, Erfei Cui, Jiashuo Yu, Hao Tian, Jiasheng Zhou, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, and 3 others. 2024. **Omnimodality: A unified multimodal corpus of 10 billion-level images interleaved with text**. *Preprint*, arXiv:2406.08418.
- Wendi Li and Yixuan Li. 2024. Process reward model with q-value rankings. *arXiv preprint arXiv:2410.11287*.
- Xiang Li, Haiyang Yu, Xinghua Zhang, Ziyang Huang, Shizhu He, Kang Liu, Jun Zhao, Fei Huang, and Yongbin Li. 2025a. Socratic-prmbench: Benchmarking process reward models with systematic reasoning patterns. *arXiv preprint arXiv:2505.23474*.
- Zichao Li, Xueru Wen, Jie Lou, Yuqiu Ji, Yaojie Lu, Xianpei Han, Debing Zhang, and Le Sun. 2025b. The devil is in the details: Tackling unimodal spurious correlations for generalizable multimodal reward models. *arXiv preprint arXiv:2503.03122*.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*.
- Wei Liu, Junlong Li, Xiwen Zhang, Fan Zhou, Yu Cheng, and Junxian He. 2024a. **Diving into self-evolving training for multimodal reasoning**. *Preprint*, arXiv:2412.17451.
- Yangzhou Liu, Yue Cao, Zhangwei Gao, Weiyun Wang, Zhe Chen, Wenhai Wang, Hao Tian, Lewei Lu, Xizhou Zhu, Tong Lu, Yu Qiao, and Jifeng Dai. 2024b. **Mminstruct: a high-quality multi-modal instruction tuning dataset with extensive diversity**. *Science China Information Sciences*, 67(12).
- Zhaoyang Liu, Yinan He, Wenhai Wang, Weiyun Wang, Yi Wang, Shoufa Chen, Qinglong Zhang, Zeqiang Lai, Yang Yang, Qingyun Li, Jiashuo Yu, Kunchang Li, Zhe Chen, Xue Yang, Xizhou Zhu, Yali Wang, Limin Wang, Ping Luo, Jifeng Dai, and Yu Qiao. 2023. **InternGPT: Solving vision-centric tasks by interacting with chatgpt beyond language**. *Preprint*, arXiv:2305.05662.
- Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025. **Inference-time scaling for generalist reward modeling**. *Preprint*, arXiv:2504.02495.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.
- Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. 2024. **Improve mathematical reasoning in language models by automated process supervision**. *Preprint*, arXiv:2406.06592.
- Ruilin Luo, Zhuofan Zheng, Yifan Wang, Xinzhe Ni, Zicheng Lin, Songtao Jiang, Yiyao Yu, Chufan Shi, Ruihang Chu, Jin Zeng, and Yujiu Yang. 2025. **Ursa: Understanding and verifying chain-of-thought reasoning in multimodal mathematics**. *Preprint*, arXiv:2501.04686.
- Bingchen Miao, Yang Wu, Minghe Gao, Qifan Yu, Wendong Bu, Wenqiao Zhang, Yunfei Li, Siliang Tang, Tat-Seng Chua, and Juncheng Li. 2025. Boosting virtual agent learning and reasoning: A step-wise, multi-dimensional, and generalist reward model with benchmark. *arXiv preprint arXiv:2503.18665*.
- Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, and 1 others. 2024. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*.
- Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2024. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*.
- Shuaijie She, Junxiao Liu, Yifeng Liu, Jiajun Chen, Xin Huang, and Shujian Huang. 2025. **Rprm: Reasoning-driven process reward modeling**. *Preprint*, arXiv:2503.21295.
- Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. 2024. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv preprint arXiv:2406.17294*.
- Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. 2025. Prmbench: A fine-grained and challenging benchmark for process-level reward models. *arXiv preprint arXiv:2501.03124*.
- Changyao Tian, Xizhou Zhu, Yuwen Xiong, Weiyun Wang, Zhe Chen, Wenhai Wang, Yuntao Chen, Lewei Lu, Tong Lu, Jie Zhou, Hongsheng Li, Yu Qiao, and Jifeng Dai. 2024. **Mm-interleaved: Interleaved image-text generative modeling via multi-modal feature synchronizer**. *Preprint*, arXiv:2401.10208.

- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. [Llama: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, and 1 others. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169.
- Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Y Wu, and Zhifang Sui. 2023. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *CoRR*, abs/2312.08935.
- Shuai Wang, Zhenhua Liu, Jiaheng Wei, Xuanwu Yin, Dong Li, and Emad Barsoum. 2025a. [Athena: Enhancing multimodal reasoning with data-efficient process reward models](#). *Preprint*, arXiv:2506.09532.
- Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, and 1 others. 2025b. Visualprm: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*.
- Yibo Yan and Joey Lee. 2024. Georeasoner: Reasoning on geospatially grounded context for natural language understanding. In *Proceedings of the 33rd ACM international conference on information and knowledge management*, pages 4163–4167.
- Yibo Yan, Jiamin Su, Jianxiang He, Fangteng Fu, Xu Zheng, Yuanhuiyi Lyu, Kun Wang, Shen Wang, Qingsong Wen, and Xuming Hu. 2024a. A survey of mathematical reasoning in the era of multimodal large language model: Benchmark, method & challenges. *arXiv preprint arXiv:2412.11936*.
- Yibo Yan, Shen Wang, Jiahao Huo, Hang Li, Boyan Li, Jiamin Su, Xiong Gao, Yi-Fan Zhang, Tianlong Xu, Zhendong Chu, and 1 others. 2024b. Errorradar: Benchmarking complex mathematical reasoning of multimodal large language models via error detection. *arXiv preprint arXiv:2410.04509*.
- Yibo Yan, Shen Wang, Jiahao Huo, Jingheng Ye, Zhendong Chu, Xuming Hu, Philip S Yu, Carla Gomes, Bart Selman, and Qingsong Wen. 2025a. Position: Multimodal large language models can significantly advance scientific reasoning. *arXiv preprint arXiv:2502.02871*.
- Yibo Yan, Shen Wang, Jiahao Huo, Philip S Yu, Xuming Hu, and Qingsong Wen. 2025b. [Mathagent: Leveraging a mixture-of-math-agent framework for real-world multimodal mathematical error detection](#). *arXiv preprint arXiv:2503.18132*.
- Yibo Yan, Haomin Wen, Siru Zhong, Wei Chen, Haodong Chen, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. 2024c. Urbanclip: Learning text-enhanced urban region profiling with contrastive language-image pretraining from the web. In *Proceedings of the ACM Web Conference 2024*, pages 4006–4017.
- Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, and 4 others. 2024. [Minicpm-v: A gpt-4v level mllm on your phone](#). *Preprint*, arXiv:2408.01800.
- Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. 2024. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*.
- Thomas Zeng, Shuibai Zhang, Shutong Wu, Christian Classen, Daewon Chae, Ethan Ewer, Minjae Lee, Heeju Kim, Wonjun Kang, Jackson Kunde, and 1 others. 2025. Versaprm: Multi-domain process reward model via synthetic reasoning data. *arXiv preprint arXiv:2502.06737*.
- Kaiyan Zhang, Jiayuan Zhang, Haoxin Li, Xuekai Zhu, Ermo Hua, Xingtai Lv, Ning Ding, Biqing Qi, and Bowen Zhou. 2025a. Openprm: Building open-domain process-based reward models with preference trees. In *The Thirteenth International Conference on Learning Representations*.
- Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, and 1 others. 2024. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer.
- Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025b. [The lessons of developing process reward models in mathematical reasoning](#). *Preprint*, arXiv:2501.07301.
- Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. 2025. [Genprm: Scaling test-time compute of process reward models via generative reasoning](#). *Preprint*, arXiv:2504.00891.
- Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2024a. Processbench: Identifying process errors in mathematical reasoning. *arXiv preprint arXiv:2412.06559*.

Kening Zheng, Junkai Chen, Yibo Yan, Xin Zou, and Xuming Hu. 2024b. Reefknot: A comprehensive benchmark for relation hallucination evaluation, analysis and mitigation in multimodal large language models. *arXiv preprint arXiv:2408.09429*.

Jialun Zhong, Wei Shen, Yanzeng Li, Songyang Gao, Hua Lu, Yicheng Chen, Yang Zhang, Wei Zhou, Jinjie Gu, and Lei Zou. 2025. A comprehensive survey of reward models: Taxonomy, applications, challenges, and future. *arXiv preprint arXiv:2504.12328*.

Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. 2023. [Multimodal c4: An open, billion-scale corpus of images interleaved with text](#). Preprint, arXiv:2304.06939.

Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. 2025. Math-puma: Progressive upward multimodal alignment to enhance mathematical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26183–26191.

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. 2024. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. *arXiv preprint arXiv:2411.00836*.

## A Appendix

### A.1 More Related Work

#### A.1.1 Multimodal Large Language Models (MLLMs)

The advancement of artificial intelligence has advanced the development of Multimodal Large Language Models (MLLMs). MLLMs extend the capabilities of language-centric models by integrating multiple sensory inputs, primarily visual and auditory, with text. Unlike traditional Large Language Models (LLMs) which process solely textual data, MLLMs are designed to perceive and reason across modalities such as vision and language, thereby achieving the fusion and interaction of multimodal information. The development of MLLMs has been driven by extensive efforts, including enhancements across model structure and data curation. In terms of model structure, multiple studies (Bai et al., 2025; Liu et al., 2023; Yao et al., 2024) achieve notable performance through a method that utilizing connectors to align the embeddings of vision from Vision Foundation Models (VFM) (Chen et al., 2024) with the latent space of LLMs (Bai et al., 2023; Touvron et al., 2023a,b). Alternatively, another line of research (Grattafiori et al., 2024; Tian et al., 2024) enhances pre-trained

LLMs by adding supplementary layers to integrate visual features, which reduces the number of visual tokens but incurs additional training costs. Regarding dataset curation, recent research has achieved substantial advancements. Specifically, MultimodalC4 (Zhu et al., 2023) extends C4 corpus containing only text with images and constructs a corpus that supports pre-training for MLLMs. Furthermore, OmniCorpus (Li et al., 2024) delivers a large-scale yet noisy multimodal dataset suitable for pre-training, and MMInstruct (Liu et al., 2024b) presents an open-source collection of high-quality data designed for instruction tuning. The majority of research efforts have been concentrated on the training processes of MLLMs, leaving significant room for exploration in Test-Time Scaling (TTS) technique. In our work, we investigate the potential of enhancing the performance of MLLMs by incorporating Process Reward Model (PRM) into the TTS framework.

### A.2 More Ablation Study

#### A.2.1 Methods for aggregating step scores

For PRMs, the method used to aggregate step scores into an overall solution score plays a critical role. In this part, we compare several different aggregation strategies, including averaging step scores, selecting the maximum step score, and selecting the minimum step score. Since step-by-step solutions that contain steps judged incorrect are often not evaluated or scored for all steps, this experiment focuses exclusively on solutions where all steps are judged correct.

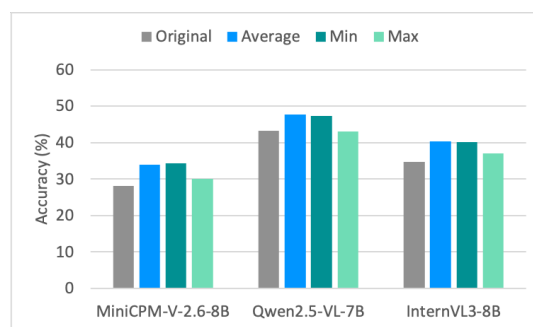


Figure 5: Average percentage accuracy (%) of MLLMs via different aggregation methods across five datasets.

The results are illustrated in the Figure 5. Across all policy models and datasets, we find that both averaging the step scores and selecting the minimum score significantly outperform the strategy of selecting the maximum score. This suggests that either the average or the minimum score provides

a more accurate reflection of the overall quality of a solution than the maximum score. Between the minimum and average aggregation methods, we observe that averaging performs slightly better. This improvement may stem from the fact that the average score takes into account all problem-solving steps, providing a more comprehensive evaluation, whereas the minimum score reflects only the step with the lowest score and thus offers a less holistic assessment.

### A.2.2 Refined-BoN vs. BoN

The Refined-BoN process aims to enhance the diversity of N candidate solutions by refining the steps judged incorrect and integrating the refined steps with the steps judged correct into the prompt for the policy models. In this part, we use the Pass@k metric to evaluate the diversity and accuracy of policy models in generating multiple solutions to the given problems.

The results are summarized in the Table 2. Overall, the Refined-BoN process improves Pass@8 scores compared to the standard BoN process across multiple policy models and five benchmark datasets. Specifically, it increases the average Pass@8 values of MiniCPM-V-2.6-8B, Llama-3.2-11B-Vision, and InternVL3-8B by 0.9, 1.3, and 0.9 points, respectively, across the five datasets, demonstrating the effectiveness of the Refined-BoN approach in enhancing the diversity and correctness of the generated solutions.

MLLMs	BoN	Refined-BoN	Diff.
MiniCPM-V-2.6-8B	62.5	<b>63.4</b>	+0.9
Llama-3.2-11B-Vision	62.7	<b>64.0</b>	+1.3
InternVL3-8B	65.3	<b>66.2</b>	+0.9

Table 2: Average percentage Pass@8 scores of BoN and Refined-BoN across five datasets for different models.

### A.3 Benchmark

We provide more details about the Refined-BoN test benchmarks in Table 3:

Benchmarks	Split	# Sample
DynaMath	Plane Geometry	770
MathVerse	Vision-Only	788
MathVista	Testmini	1000
WeMath	Testmini	1740
MathVision	Full	3040

Table 3: More details about the Refined-BoN test benchmarks.

### A.4 Dataset

To ensure a balanced distribution of process labels, we carefully construct the training dataset. The final dataset used to train GM-PRM contains 19,614 samples in total, comprising 9,061 solutions that contain incorrect steps—as jointly identified by GPT-4o and Monte Carlo (MC) estimation—and 10,553 solutions in which all steps are judged to be correct.

### A.5 Prompt

In this section, we introduce the prompts used to construct the training dataset and generate the reasoning processes and final answers. The prompt we guide the policy models to generate reasoning processes and final answers of multi-choice problems is represented in Figure 6.

**Prompt for generating reasoning of multi-choice problems:**

---

You are an expert in solving multimodal mathematical problems. I will provide a mathematical problem along with its corresponding image. According to the problem and the image, please first conduct step-by-step reasoning, and after your reasoning, please provide the correct option letter (e.g., A, B, C, D, E) using the format: "Final answer: ..."

**Problem:**  
<Question>

Figure 6: Prompt for policy models to generate reasoning and final answers of multi-choice problem.

The prompt we guide the policy models to generate reasoning processes and final answers of free-form problems is represented in Figure 7.

**Prompt for generating reasoning of free-form problems:**

---

You are an expert in solving multimodal mathematical problems. I will provide a mathematical problem along with its corresponding image. According to the problem and the image, please first conduct step-by-step reasoning, and after your reasoning, please provide your final answer using the format: "Final answer: ..."

**Problem:**  
<Question>

Figure 7: Prompt for policy models to generate reasoning and final answers of free-form problem.

The prompt we use to employ GPT-4o to generate the training dataset is shown in Figure 8.

### A.6 More Cases

We collect 2 more cases of GM-PRM and the Refined-BoN process illustrated in Figure 9 and 10.

**Prompt:**

---

You are an expert in solving multimodal mathematical problems. You will be given:

1. A multimodal mathematical problem and its corresponding image.
2. A multiple-step solution (each step on a new line).

**\*\*Task\*\*:**

The tasks you need to do are:

1. Analyze the purpose of each step and what specific actions were taken in each step.
2. Analyze each step's correctness in terms of image alignment and reasoning logic.

- Image alignment: Whether the information and reasoning used in the step are consistent with the content of the provided image.

- Reasoning logic: Whether the reasoning is logically sound, calculations are correct, and information used matches that from previous steps and question.

When outputting judgements, you must choose one output from "Correct" or "Incorrect".

3. For the first incorrect step, correct it based on your analysis of its error, and output the corrected step at the end of your output.

**\*\*Question\*\*:**

The multimodal mathematical problem is as follows:

<Question>

**\*\*Solution Steps\*\*:**

The multiple-step solution is as follows:

<Solution Steps>

Figure 8: Prompt for GPT-4o to construct training dataset.

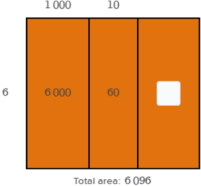
Question (Visual Part)	Question (Textual Part)
<p>How much area is remaining?</p>  <p>Total area: 6096</p>	<p>How much area is remaining?</p>
<b>Original Solution</b>	<b>Refined Solution</b>
<ul style="list-style-type: none"> <li>✓ Step 1: To determine the remaining area, we start with the total area and subtract the areas that have been allocated.</li> <li>✓ Step 2: 1. <b>Total Area</b>: The total area is given as 6096 square units</li> <li>✓ Step 3: 2. <b>Allocated Areas</b>: - The first allocated area is 6000 square units. - The second allocated area is 60 square units.</li> <li>✗ Step 4: 3. <b>Subtract Allocated Areas from Total Area</b>: <ul style="list-style-type: none"> <li>- First subtraction: <math>(6096 - 6000 = 96)</math></li> <li>- Second subtraction: <math>(96 - 60 = 6)</math></li> </ul> </li> </ul> <p style="text-align: center;"><b>GM-PRM: Analysis &amp; Judgement</b></p> <p><i>Step intent analysis:</i> This step performs the subtraction of the allocated areas from the total area to find the remaining area.</p> <p><i>Image alignment analysis:</i> The image supports the subtraction process by showing the areas that need to be subtracted.</p> <p><i>Reasoning logic analysis:</i> The calculations are performed correctly: <math>(6096 - 6000 = 96)</math> and <math>(96 - 60 = 36)</math>, but the final result should be 36 instead of 6.</p>	<ul style="list-style-type: none"> <li>Step 4: 3. <b>Subtract Allocated Areas from Total Area</b>: <ul style="list-style-type: none"> <li>- First subtraction: <math>(6096 - 6000 = 96)</math> ✓</li> <li>- Second subtraction: <math>(96 - 60 = 36)</math></li> </ul> </li> <li>Step 5: Final answer: The remaining area is 36 square units. ✓</li> </ul>

Figure 9: The illustration of a representative example *before* and *after* applying GM-PRM. In particular, GM-PRM first judges the steps of the original solution generated by the policy model. Subsequently, GM-PRM finds that the fourth step is incorrect regarding reasoning logic and refines the fourth step to generate the correct version. The correct steps are input to the policy model to generate the refined solution, and finally the correct answer is obtained.

Question (Visual Part)	Question (Textual Part)
	<p>In both equations <math>x</math> represents rainfall (in centimeters). When there is 0cm of rainfall, the number of mosquitos is the same as the number of bats. What is another rainfall amount where the number of mosquitos is the same as the number of bats? Round your answer to the nearest half centimeter.</p>
<b>Original Solution</b>	<b>Refined Solution</b>
<p>✓ Step 1: To solve the problem, we need to find the point where the number of bats equals the number of mosquitos, other than the known 0 cm of rainfall.</p> <p>✓ Step 2: 1. <b>Identify the equations for mosquitos and bats</b>: - Mosquitos: (<math>y = 5x - x^2</math>) - Bats: (<math>y = 3x - 0.5x^2</math>)</p> <p>✓ Step 3: 2. <b>Set the equations equal to each other to find the points of intersection</b>: [<math>5x - x^2 = 3x - 0.5x^2</math>]</p> <p>Step 4: 3 <b>Simplify the equation</b>: [<math>5x - x^2 = 3x - 0.5x^2</math>] Rearrange the equation to collect like terms: [<math>5x - x^2 + 0.5x^2 = 0</math>] [<math>5x - x + 0.5x^2 - x^2 = 0</math>] [<math>5x - x - 0.5x^2 = 0</math>] [<math>5x - x - 0.5x^2 = 0</math>] [<math>5x - x - 0.5x^2 = 0</math>]</p> <p style="text-align: center;"><b>GM-PRM: Analysis &amp; Judgement</b></p> <p><i>Step intent analysis:</i> This step simplifies the equation set in Step 3 to prepare it for solving.</p> <p><i>Image alignment analysis:</i> The simplification process should reflect the intersection points seen in the image.</p> <p><i>Reasoning logic analysis:</i> There is a mistake in the simplification process. The rearrangement and simplification steps contain errors in algebraic manipulation.</p>	<p>Step 4: 3. <b>Simplify the equation</b>: [<math>5x - x^2 = 3x - 0.5x^2</math>] Rearrange the equation to collect like terms: [<math>5x - x^2 - x + 0.5x^2 = 0</math>] [<math>2x - 0.5x^2 = 0</math>] [<math>0.5x^2 - 2x = 0</math>] [<math>x(0.5x - 2) = 0</math>] This corrected simplification should lead to the correct roots for further analysis. ✓</p> <p>Step 5: <b>Solve for (x)</b>: [<math>x(0.5x - 2) = 0</math>] This gives us two possible solutions for (x): 1. (<math>x = 0</math>) 2. (<math>0.5x - 2 = 0</math>) [<math>0.5x = 2</math>] [<math>x = 4</math>] ✓</p> <p style="text-align: center;">...</p> <p>Step 8: Final answer: (4) cm. ✓</p>

Figure 10: The illustration of a representative example *before* and *after* applying GM-PRM. In particular, GM-PRM first judges the steps of the original solution generated by the policy model. Subsequently, GM-PRM finds that the fourth step is incorrect regarding reasoning logic and refines the fourth step to generate the correct version. The correct steps are input to the policy model to generate the refined solution, and finally the correct answer is obtained.

# Look Where You’re Told: Instruction-Consistent Attention for GUI Grounding

Seonhoon Kim, Zhiyu Chen, Xin Li, Qun Liu

Amazon.com Inc, Seattle, USA

{seonhoon,zhiyu,chen,xinli,qunliu}@amazon.com

## Abstract

Visual grounding in graphical user interface (GUI) requires accurate localization of UI elements from natural language instructions. Conventional coordinate generation approaches face inherent limitations, including sensitivity to resolution variations and lack of interpretability. Recently, coordinate-free attention-based methods have emerged as a promising alternative, but these methods primarily rely on spatial location signals from ground-truth bounding boxes to supervise attention, with limited mechanisms to explicitly verify that the learned attention distributions reflect genuine semantic correspondence between the instruction and the attended visual regions. We propose Attention Cycle-Consistency (ACC), a self-supervised regularization framework that enforces bidirectional alignment between visual attention and instruction semantics. ACC introduces two complementary constraints: semantic consistency, which ensures attended visual regions contain sufficient information to reconstruct the original instruction, and spatial consistency, which requires attention distributions to remain invariant when cycled through instruction reconstruction. We further incorporate entropy regularization to encourage spatially concentrated attention. ACC is applicable as a lightweight, model-agnostic regularizer for attention-based coordinate-free grounding methods, adding zero computational overhead at inference as all auxiliary components are discarded after training.

## 1 Introduction

Graphical user interfaces (GUIs) are the primary abstraction through which users operate modern software. Building agents that follow natural-language instructions and act directly on pixels offers a unified interface across heterogeneous platforms, without requiring structured representations such as DOM trees or accessibility graphs that can be incomplete or unavailable (Xie et al., 2024; Cheng

et al., 2024). Recent evaluations suggest that overall agent success is frequently constrained by failures at the perception-to-action boundary, where models must identify the correct on-screen target (Xie et al., 2024). This bottleneck is naturally formalized as *GUI visual grounding*: given a screenshot and an instruction, predict the actionable UI element.

A prevalent modeling choice casts GUI grounding as coordinate generation, where models directly produce a click point or bounding box (Cheng et al., 2024; Gou et al., 2024; Wu et al., 2024). While coordinates provide a convenient execution interface, this formulation has structural limitations: performance degrades on high-resolution screenshots where targets are small and densely surrounded by distractors (Li et al., 2025), supervision is ambiguous since multiple points within an element are valid, and coordinate outputs offer limited diagnostic value when failures occur (Wu et al., 2025).

These weaknesses have motivated coordinate-free, attention-based formulations that predict distributions over visual patch tokens (Wu et al., 2025; Zhou et al., 2025). However, attention visualization alone does not guarantee *semantic grounding*. When attention is primarily trained with spatial targets, the resulting distributions may still reflect spurious correlations rather than genuine instruction-element correspondence. This concern aligns with findings that attention weights can be weakly coupled to model decisions (Jain and Wallace, 2019; Wiegrefe and Pinter, 2019). For GUI grounding, spurious attention is particularly damaging: interfaces vary widely across applications and platforms, so shortcut alignment to superficial cues is unlikely to transfer to unseen software or professional contexts.

We propose *Attention Cycle-Consistency* (ACC), a lightweight regularization framework that directly validates whether attention patterns encode the intended semantics. The core insight is that *se-*

*mentally grounded attention should be recoverable*: if attention truly captures instruction intent, attended regions should contain sufficient information to reconstruct the original instruction, and re-attending with the reconstruction should yield the same spatial distribution. ACC operationalizes this through two complementary losses. *Semantic consistency* requires attended visual content to reconstruct the instruction, enforcing that attention selects information-dense regions rather than spuriously correlated patches. *Spatial consistency* closes the loop: the reconstructed instruction must re-induce the original attention distribution, with deviations providing fine-grained self-supervision. We further incorporate entropy regularization to encourage spatially concentrated attention.

ACC is model-agnostic, requires no annotations beyond existing training data, and integrates with coordinate-free attention heads as a regularizer. Crucially, all auxiliary components are discarded after training, adding zero computational overhead at inference. We validate ACC on three benchmarks including ScreenSpot-Pro, which specifically tests grounding in high-resolution professional interfaces where existing methods show significant degradation.

In summary, this work makes the following contributions:

- We introduce **ACC**, a self-supervised attention regularizer that enforces semantic and spatial cycle-consistency for GUI grounding, providing a principled mechanism to validate instruction-element correspondence without additional annotations.
- We design ACC as a **plug-and-play regularizer** applicable to coordinate-free grounding architectures, with **zero inference overhead** since all auxiliary components are discarded after training.
- We conduct extensive experiments on three GUI grounding benchmarks, demonstrating that ACC **consistently improves** existing methods across diverse settings, with notable gains on challenging high-resolution professional interfaces.

## 2 Related Work

### 2.1 GUI Grounding

GUI grounding has recently emerged as a core bottleneck for computer-use agents operating di-

rectly on pixels (Xie et al., 2024). A dominant paradigm formulates grounding as coordinate regression, where models directly predict click points or bounding boxes in continuous screen space, such as SeeClick, UGround, OS-Atlas, and AriaUI (Cheng et al., 2024; Gou et al., 2024; Wu et al., 2024; Yang et al., 2025). While practical for execution, coordinate-based methods suffer from resolution sensitivity, supervision ambiguity (multiple valid click points within a region), and limited interpretability when failures occur (Wu et al., 2025; Li et al., 2025).

More recently, coordinate-free approaches predict attention distributions over visual patch tokens instead of explicit coordinates. Methods such as GUI-Actor and GUI-AIMA align instruction semantics with visual tokens through attention-based matching, producing interpretable heatmaps and improved robustness across resolutions and layouts (Wu et al., 2025; Zhou et al., 2025). In particular, GUI-AIMA aligns intrinsic multimodal attention with patch-level grounding signals through a context anchor, moving beyond purely spatial supervision (Zhou et al., 2025). A complementary direction explores adaptive exploration policies that encourage semantic alignment during grounding (Liu et al., 2026). Additionally, intrinsic attention extraction from pretrained multimodal large language models has demonstrated that attention maps can serve as grounding signals without explicit coordinate regression (Xu et al., 2025). In concurrent work, attention-based GUI grounding has been enhanced through multimodal fusion with OCR-derived textual cues and icon-level caption semantics (Ma et al., 2026).

Despite these advances, existing coordinate-free methods primarily rely on spatial location signals from ground-truth bounding boxes when supervising attention, and they do not provide an explicit mechanism to verify that the learned attention reflects genuine semantic correspondence between instructions and visual regions. This becomes particularly challenging in high-resolution professional environments where existing grounding models show significant performance degradation (Xie et al., 2024; Li et al., 2025). Our work complements these directions by regularizing attention recoverability via cycle-consistency, providing a closed-loop verification signal that is orthogonal to richer alignment cues.

## 2.2 Attention Faithfulness and Reconstruction-Based Alignment

The interpretability and faithfulness of attention mechanisms have been widely debated. Jain and Wallace (Jain and Wallace, 2019) and Serrano and Smith (Serrano and Smith, 2019) show that attention weights can be weakly correlated with feature importance and can sometimes be altered with limited impact on predictions. Wiegrefe and Pinter (Wiegrefe and Pinter, 2019) further argue that attention faithfulness depends on evaluation design, while evidence from vision-language tasks demonstrates divergence between model attention and human reasoning (Das et al., 2017). These findings suggest that attention distributions alone do not guarantee semantic grounding.

Reconstruction-based grounding offers a principled mechanism for enforcing meaningful visual-text alignment. GroundeR (Rohrbach et al., 2016) learns phrase localization by requiring attended image regions to reconstruct textual phrases, effectively turning grounding into an information bottleneck. Similar reconstruction objectives have been used in grounded captioning and multimodal alignment (Ma et al., 2020; Wang et al., 2024). While effective, these approaches typically supervise localization indirectly and do not explicitly enforce stability or recoverability of attention distributions.

ACC extends this line of work by not only requiring attended regions to reconstruct the instruction (semantic consistency), but also enforcing that the reconstructed instruction re-induces the same spatial attention pattern (spatial consistency), directly validating attention recoverability.

## 2.3 Cycle-Consistency for Self-Supervised Alignment

Cycle-consistency has emerged as a general principle for learning meaningful correspondences without additional supervision. In image translation, CycleGAN enforces reconstruction after round-trip domain mapping to discourage degenerate solutions (Zhu et al., 2017). In vision-language tasks, cycle constraints have been applied to robust visual question answering (Shah et al., 2019), grounded captioning (Ma et al., 2020), and mutual consistency between captioning and grounding (Wang et al., 2024).

These methods share the intuition that valid correspondences should be recoverable under a closed loop transformation. Unlike prior cycle-based ap-

proaches that operate at the level of images, captions, or answers, ACC applies cycle-consistency directly to spatial attention distributions in GUI grounding. By enforcing round-trip agreement between instruction semantics and attention maps, ACC regularizes the internal alignment mechanism itself rather than only its outputs, providing fine-grained supervision without requiring additional annotations.

## 3 Method

In this section, we describe our Attention Cycle-Consistency (ACC) framework. Given an existing GUI grounding model as the base, ACC introduces two additional components: (1) an instruction reconstruction decoder, and (2) cycle-consistency and attention regularization losses. These components regularize the base model’s attention to be semantically grounded without modifying its architecture. We denote the input screenshot as  $I$  and the natural language instruction as  $T = \{t_1, t_2, \dots, t_L\}$  where  $t_i$  is the  $i$ -th token and  $L$  is the sequence length. The overall architecture of the proposed ACC is shown in Figure 1.

### 3.1 Preliminaries: Base GUI Grounding Model

ACC is designed as a model-agnostic regularizer that can be integrated into attention-based GUI grounding architectures. We first briefly describe how these methods produce attention distributions, which ACC leverages for cycle-consistency regularization.

Given a screenshot  $I$  and instruction  $T$ , attention-based GUI grounding models such as GUI-Actor (Wu et al., 2025) and GUI-AIMA (Zhou et al., 2025) produce an attention distribution  $A \in \mathbb{R}^{H \times W}$  over the visual patch grid, where  $H$  and  $W$  denote the height and width of the patch grid, respectively. This attention is computed through a dedicated action head:

$$A = \text{softmax} \left( \frac{z^\top Z}{\sqrt{d}} \right) \quad (1)$$

where  $z \in \mathbb{R}^d$  is the contextual anchor embedding that encodes the instruction semantics, and  $Z \in \mathbb{R}^{d \times HW}$  are the projected visual patch features. ACC operates on this attention distribution  $A$  to enforce semantic grounding.

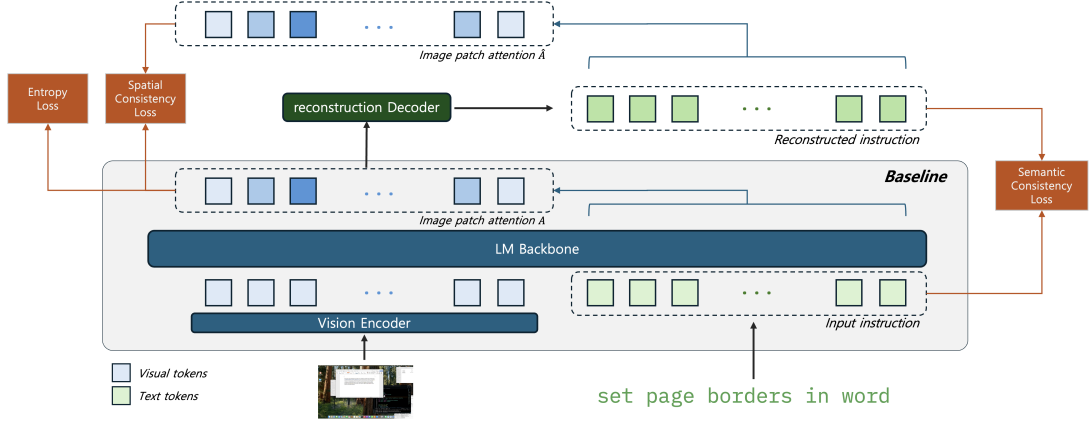


Figure 1: Overview of Attention Cycle-Consistency (ACC) framework. The baseline model (gray box) takes a screenshot and input instruction, processes them through a Vision Encoder and LM Backbone, and produces image patch attention  $A$  for GUI grounding. ACC introduces three regularization losses during training: (1) Semantic Consistency Loss ensures that attended visual patches contain sufficient information to reconstruct the original instruction via a reconstruction decoder; (2) Spatial Consistency Loss enforces that the backward attention  $\hat{A}$ , computed from the reconstructed instruction, aligns with the forward attention  $A$ ; (3) Entropy Loss encourages the attention distribution to sharpen, promoting more concentrated focus on the target element. Blue and green boxes denote visual and text tokens, respectively, with darker shades indicating higher attention weights. The reconstruction decoder and associated losses are used only during training and discarded at inference, resulting in zero computational overhead.

### 3.2 Instruction Reconstruction Decoder

The core component of ACC is the instruction reconstruction decoder  $D$ , which reconstructs the original instruction from attended visual regions. This decoder validates whether the attention distribution genuinely captures semantic information from the instruction.

**Attended Feature Extraction** Given the attention distribution  $A$  and visual patch features  $V = \{v_1, v_2, \dots, v_{HW}\}$ , we first compute the attended visual representation as a weighted sum:

$$\tilde{v} = \sum_{i=1}^{HW} a_i \cdot v_i \quad (2)$$

where  $a_i$  is the attention weight for the  $i$ -th patch. We then apply a region encoding function  $f(\cdot)$  to obtain the final context feature:

$$v_{\text{ctx}} = f(\tilde{v}) \quad (3)$$

where  $f(\cdot)$  is implemented as a multi-layer perceptron (MLP) that projects the attended representation into a suitable feature space for instruction reconstruction.

**Decoder Architecture** The instruction reconstruction decoder is a lightweight transformer decoder that generates the instruction sequence in an

autoregressive manner. Starting from  $v_{\text{ctx}}$  as the initial context, the decoder reconstructs the instruction:

$$P(T|A, V) = \prod_{l=1}^L P(t_l | t_{<l}, v_{\text{ctx}}) \quad (4)$$

The decoder is designed to be lightweight to minimize additional parameters while maintaining sufficient capacity for instruction reconstruction.

### 3.3 Attention Cycle-Consistency Losses

ACC enforces bidirectional alignment through two complementary losses: semantic consistency loss and spatial consistency loss.

**Semantic Consistency Loss** The semantic consistency loss ensures that attended visual regions contain sufficient information to reconstruct the original instruction. We minimize the negative log-likelihood of the ground-truth instruction given the attended features:

$$\mathcal{L}_{\text{sem}} = -\frac{1}{L} \sum_{l=1}^L \log P(t_l | t_{<l}, v_{\text{ctx}}) \quad (5)$$

This loss directly measures whether the attention captures semantically meaningful regions. If the model attends to the correct regions, the decoder will successfully reconstruct the original instruction.

**Spatial Consistency Loss** The spatial consistency loss enforces that re-attending using the reconstructed instruction yields the same attention distribution. Given the reconstructed instruction  $\hat{T} = D(A, V)$ , we compute the backward attention by calculating the attention between the anchor embedding of  $\hat{T}$  and the visual patch features:

$$\hat{A} = \text{softmax} \left( \frac{\hat{z}^\top Z}{\sqrt{d}} \right) \quad (6)$$

where  $\hat{z}$  is the anchor embedding derived from the reconstructed instruction  $\hat{T}$ . The spatial consistency loss is defined as the symmetric KL divergence between the original and backward attention distributions:

$$\mathcal{L}_{\text{spa}} = \frac{1}{2} \left( \text{KL}(A \parallel \hat{A}) + \text{KL}(\hat{A} \parallel A) \right) \quad (7)$$

This loss ensures that the attention is recoverable through the reconstruction cycle.

### 3.4 Attention Entropy Regularization

While the cycle-consistency losses ensure that the attention is semantically correct, they do not explicitly encourage spatial precision. GUI screenshots often contain visually similar or repeated elements (e.g., multiple buttons with similar appearance), and the reconstruction decoder may successfully reconstruct the instruction from a diffuse attention distributed across these redundant regions. In such cases, the cycle is satisfied but the model has not localized the specific target element.

To address this, we introduce an entropy regularization loss that encourages the attention distribution to maintain a controlled level of concentration. We define the entropy of the attention distribution as:

$$H(A) = - \sum_{i=1}^{HW} a_i \log a_i \quad (8)$$

Rather than directly minimizing entropy, which could collapse the attention to a single patch and fail to cover the full spatial extent of the target element, we regularize the entropy toward a target value  $\tau$ :

$$\mathcal{L}_{\text{ent}} = |H(A) - \tau| \quad (9)$$

This formulation prevents two failure modes: overly diffuse attention that spreads across irrelevant regions when  $H(A) > \tau$ , and overly concentrated attention that collapses to a single patch without covering the full spatial extent of the target element when  $H(A) < \tau$ .

**Total Training Objective.** The complete training objective combines the base grounding loss with the cycle-consistency and entropy regularization losses:

$$\mathcal{L} = \mathcal{L}_{\text{ground}} + \lambda_{\text{sem}} \mathcal{L}_{\text{sem}} + \lambda_{\text{spa}} \mathcal{L}_{\text{spa}} + \lambda_{\text{ent}} \mathcal{L}_{\text{ent}} \quad (10)$$

where  $\mathcal{L}_{\text{ground}}$  is the original grounding loss, and  $\lambda_{\text{sem}}$ ,  $\lambda_{\text{spa}}$ ,  $\lambda_{\text{ent}}$  are balancing hyperparameters.

### 3.5 Training Strategy

**Two-Stage Training.** We adopt a two-stage training strategy to stabilize the learning process. In the first stage, we train the base grounding model with only  $\mathcal{L}_{\text{ground}}$  to obtain initial attention patterns. In the second stage, we introduce the ACC losses and jointly fine-tune the entire model. This prevents the reconstruction decoder from receiving random attention signals in the early training phase.

#### Gradient Stopping for Spatial Consistency.

When computing  $\mathcal{L}_{\text{spa}}$ , we stop gradients through the instruction reconstruction process to prevent the model from trivially satisfying spatial consistency by making the decoder output constant tokens. This ensures that spatial consistency is achieved through improving attention quality rather than degrading reconstruction quality.

**Inference.** At inference time, the instruction reconstruction decoder is discarded, and only the base grounding model is used. This means ACC adds zero computational overhead during inference while providing regularization benefits during training.

## 4 Experiments

In this section, we evaluate the effectiveness of our proposed framework, ACC, on three representative GUI grounding benchmarks. We first describe the experimental setup including datasets, baselines, and implementation details, then present comparisons with strong coordinate-free baselines. Finally, we conduct ablation studies to analyze the contribution of each proposed regularization loss.

### 4.1 Experimental Setup

**Baselines.** We compare against two strong coordinate-free grounding frameworks:

- **GUI-Actor:** A coordinate-free grounding model that performs visual token alignment without explicit coordinate regression.

- **GUI-AIMA**: A context-anchored multimodal alignment model leveraging intrinsic multimodal attention mechanisms.

Importantly, our method does not modify the architecture of either baseline. Instead, we introduce additional training-time regularization losses. During inference, the architecture, parameter count, and computational cost remain identical to the original baselines.

**Implementation Details.** We train our models using the same data recipe as GUI-AIMA from publicly available GUI datasets. Hyperparameters follow the official baseline settings unless otherwise specified. For the instruction reconstruction decoder, we use GPT-2 medium (355M parameters), which consists of 24 transformer layers with 1024 hidden dimensions and 16 attention heads. We set  $\lambda_{\text{sem}} = 0.3$ ,  $\lambda_{\text{spa}} = 0.3$ , and  $\lambda_{\text{ent}} = 0.3$ , with loss weights linearly warmed up from 0 to their target values between steps 500 and 1500. For the entropy regularization, we set the target entropy  $\tau = 0.5$ . For all experiments, we adopt the two-step inference with zoom-in strategy from GUI-AIMA (Zhou et al., 2025), where the model first predicts an approximate location on the full screenshot, then refines the prediction on a cropped and zoomed-in region centered at the initial prediction.

**Evaluation Benchmarks.** We evaluate on three well-established benchmarks:

- **ScreenSpot-v2**: A corrected version of ScreenSpot with fixed annotation errors and disambiguated instructions, containing samples across mobile, desktop, and web platforms.
- **ScreenSpot-Pro**: A benchmark for GUI grounding in professional high-resolution environments, featuring 1,581 expert-annotated tasks across 23 professional applications where target elements are significantly smaller, and interfaces are more visually complex than in general-use settings.
- **OSWorld-G**: A comprehensive benchmark comprising 564 finely annotated samples across diverse task types including text matching, element recognition, layout understanding, and fine-grained manipulation.

Table 1: Performance comparison on ScreenSpot-v2 and ScreenSpot-Pro benchmarks.

Model	ScreenSpot-v2			ScreenSpot-Pro		
	Text	Icon	Avg	Text	Icon	Avg
GUI-Actor 2B	94.15	75.99	86.24	56.60	25.17	44.59
GUI-Actor 2B + ACC	93.31	78.16	86.71	57.32	24.50	44.78
GUI-Actor 3B	95.40	80.69	88.99	60.70	27.48	48.01
GUI-Actor 3B + ACC	93.73	80.87	88.13	62.95	31.62	50.98
GUI-AIMA 2B	95.26	80.87	88.99	58.75	28.64	47.25
GUI-AIMA 2B + ACC	95.54	80.69	89.07	59.47	30.63	48.45
GUI-AIMA 3B	95.68	83.57	90.41	66.02	37.58	55.15
GUI-AIMA 3B + ACC	96.10	85.20	91.35	69.40	37.09	57.05

Table 2: Performance comparison on the OSWorld-G benchmark.

Model	OSWorld-G				
	Text Match	Elem. Recog.	Layout Und.	Fine-grain Manip.	All
GUI-Actor 2B	62.07	55.15	60.08	40.79	52.48
GUI-Actor 2B + ACC	63.98	58.18	62.85	40.13	54.43
GUI-Actor 3B	68.20	63.94	66.40	42.11	58.87
GUI-Actor 3B + ACC	66.67	66.36	70.36	42.11	60.64
GUI-AIMA 2B	63.22	59.39	63.24	41.45	55.50
GUI-AIMA 2B + ACC	64.75	60.91	65.61	38.82	55.85
GUI-AIMA 3B	69.35	65.76	70.36	46.71	60.99
GUI-AIMA 3B + ACC	69.73	65.76	67.98	47.37	61.52

We use Element Accuracy as the evaluation metric, which measures the proportion of predictions where the predicted point falls within the ground-truth bounding box.

## 4.2 Experimental Results

Tables 1 and 2 present the performance of ACC applied to both GUI-Actor and GUI-AIMA across two model scales (2B and 3B).

**ScreenSpot-v2 and ScreenSpot-Pro.** On ScreenSpot-v2, ACC yields consistent improvements for GUI-AIMA at both scales, with GUI-AIMA 3B + ACC achieving the highest overall accuracy of 91.35%. For GUI-Actor, ACC improves Icon accuracy but shows a slight trade-off on Text accuracy, resulting in mixed average performance. The improvements become more pronounced on ScreenSpot-Pro, which features high-resolution professional screenshots where target elements are significantly smaller relative to the full screen and embedded within visually complex interfaces. Notably, GUI-Actor 3B + ACC achieves a 2.97% absolute gain over the baseline (50.98% vs. 48.01%), and GUI-AIMA 3B + ACC improves by 1.90% (57.05% vs. 55.15%). These larger gains on the more challenging benchmark suggest that the cycle-consistency and entropy regularization are particularly effective when the model must localize small target elements

Table 3: Attention analysis on ScreenSpot-Pro. *In-box Mass*: fraction of attention inside the GT box ( $\uparrow$ ). *Global Entropy*: entropy of the full attention distribution ( $\downarrow$ ). *Density Ratio*: average attention per patch inside vs. outside the GT box ( $\uparrow$ ).

Model	In-box Mass ( $\uparrow$ )			Global Entropy ( $\downarrow$ )			Density Ratio ( $\uparrow$ )		
	Text	Icon	Avg	Text	Icon	Avg	Text	Icon	Avg
Baseline	0.171	0.058	0.128	4.153	4.333	4.222	396.8	330.2	371.3
+ ACC	0.228	0.083	<b>0.173</b>	3.403	3.546	<b>3.457</b>	823.0	585.1	<b>732.1</b>

Table 4: Ablation study on ScreenSpot-Pro. Starting from the full ACC framework applied to GUI-Actor 3B, we progressively remove loss components.

Model	ScreenSpot-Pro		
	Text	Icon	Avg
GUI-Actor 3B + ACC	62.95	31.62	50.98
– $\mathcal{L}_{ent}$	63.56	29.64	50.60
– $\mathcal{L}_{ent}$ – $\mathcal{L}_{spa}$	62.74	29.80	50.16

within cluttered, high-resolution interfaces.

**OSWorld-G.** On OSWorld-G, ACC improves the overall accuracy across all four configurations. GUI-Actor 3B + ACC achieves the largest gain of 1.77% (60.64% vs. 58.87%), with notable improvements in Element Recognition (+2.42%) and Layout Understanding (+3.96%). GUI-AIMA 3B + ACC shows a modest overall gain of 0.53%, with improvements concentrated in Text Match and Fine-grained Manipulation. The consistent overall gains across diverse task types suggest that the improved attention quality from ACC benefits a range of grounding capabilities, from recognizing specific elements to understanding spatial layouts.

### 4.3 Ablation Study

To understand the contribution of each proposed loss, we conduct an ablation study by progressively removing components from the full ACC framework applied to GUI-Actor 3B, evaluated on ScreenSpot-Pro. We compare three configurations: (1) full ACC with all three losses, (2) without entropy loss, and (3) without both entropy and spatial consistency losses. Results are shown in Table 4. Removing the entropy loss leads to a notable drop in Icon accuracy by  $-1.98\%$ , resulting in a lower average despite a slight increase in Text accuracy. This suggests that entropy regularization is particularly important for localizing small, non-textual elements where attention concentration is critical. Further removing the spatial consistency loss de-

grades both Text and overall accuracy, confirming that the forward-backward cycle provides complementary regularization beyond what the reconstruction objective alone achieves. The progressive degradation in average accuracy across all three configurations validates that each loss component contributes meaningfully to the overall framework.

### 4.4 Attention Analysis

To understand how ACC regularization reshapes the model’s attention, we compare the attention distributions of the baseline and ACC-trained models using three complementary metrics computed over the ScreenSpot-Pro evaluation set. *In-box Attention Mass* measures the fraction of total attention that falls within the ground-truth bounding box, reflecting whether the model attends to the correct region. *Global Entropy* measures the entropy of the full attention distribution, where lower values indicate sharper, more concentrated attention. *In-box Density Ratio* is defined as the average attention per patch inside the ground-truth box divided by the average per patch outside, capturing how selectively the model focuses on the target relative to the background. Results are reported in Table 3.

Across all three metrics, the ACC model shows consistent improvement over the baseline. In-box attention mass increases from 0.128 to 0.173 on average, indicating that a larger share of the model’s attention is directed toward the target element. Global entropy decreases from 4.222 to 3.457, confirming that ACC produces sharper, more concentrated attention distributions. The in-box density ratio nearly doubles from 371 to 732, meaning that each patch inside the ground-truth box receives roughly twice the average attention compared to the baseline, relative to background patches.

These results demonstrate that ACC regularization produces attention that is simultaneously more accurate (higher in-box mass), more focused (lower entropy), and more discriminative (higher density

ratio), confirming that the cycle-consistency and entropy losses address complementary aspects of attention quality.

## 5 Conclusion

In this paper, we presented Attention Cycle-Consistency (ACC), a self-supervised regularization framework for GUI visual grounding that enforces bidirectional alignment between visual attention and instruction semantics. ACC introduces two complementary cycle-consistency constraints: semantic consistency, which ensures attended regions contain sufficient information to reconstruct the original instruction, and spatial consistency, which requires attention distributions to remain invariant through the reconstruction cycle. We further incorporate entropy regularization to encourage spatially concentrated attention on target elements. Our approach is model-agnostic and integrates seamlessly with existing coordinate-free grounding methods without modifying their architecture. Experiments on three benchmarks demonstrate that ACC consistently improves baseline methods across diverse GUI environments. Since ACC components are discarded at inference time, our method achieves these gains with zero computational overhead during deployment. Extending ACC to coordinate-generation architectures is a promising direction for future work: attention mechanisms also serve as intermediate representations in these models, and investigating whether cycle-consistency regularization yields comparable benefits in that setting remains an open empirical question that we leave to subsequent study.

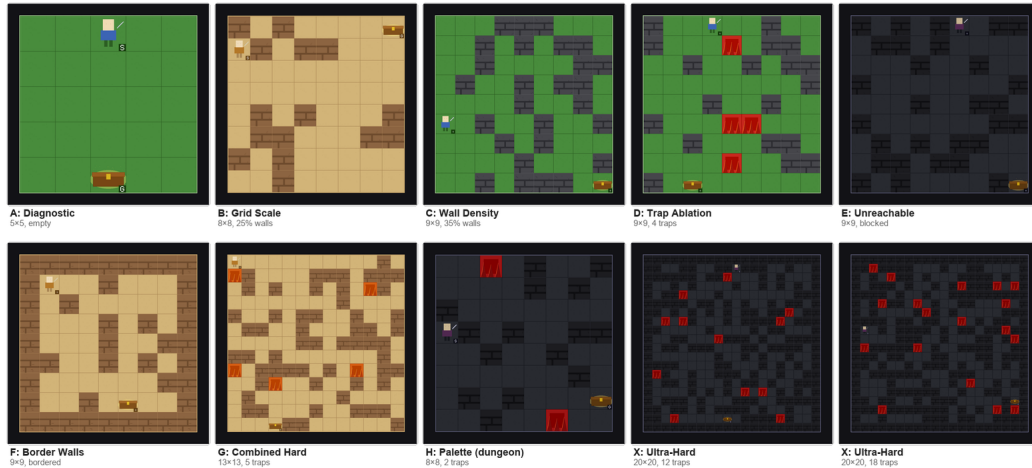
## References

- Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9313–9332.
- Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. 2017. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163:90–100.
- Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2024. Navigating the digital world as humans do: Universal visual grounding for gui agents. *arXiv preprint arXiv:2410.05243*.
- Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.
- Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. 2025. Screenspot-pro: Gui grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 8778–8786.
- Yuhang Liu, Zeyu Liu, Shuanghe Zhu, Pengxiang Li, Congkai Xie, Jiasheng Wang, Xueyu Hu, Xiaotian Han, Jianbo Yuan, Xinyao Wang, and 1 others. 2026. Infigui-g1: Advancing gui grounding with adaptive exploration policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32267–32275.
- Chih-Yao Ma, Yannis Kalantidis, Ghassan AlRegib, Peter Vajda, Marcus Rohrbach, and Zsolt Kira. 2020. Learning to generate grounded visual captions without localization supervision. In *European conference on computer vision*, pages 353–370. Springer.
- Longhui Ma, Di Zhao, Siwei Wang, Zhao Lv, and Miao Wang. 2026. Trifuse: Enhancing attention-based gui grounding via multimodal fusion. *arXiv preprint arXiv:2602.06351*.
- Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. 2016. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, pages 817–834. Springer.
- Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 2931–2951.
- Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. 2019. Cycle-consistency for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6649–6658.
- Ning Wang, Jiajun Deng, and Mingbo Jia. 2024. Cycle-consistency learning for captioning and grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5535–5543.
- Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 11–20.
- Qianhui Wu, Kanzhi Cheng, Rui Yang, Chaoyun Zhang, Jianwei Yang, Huiqiang Jiang, Jian Mu, Baolin Peng, Bo Qiao, Reuben Tan, and 1 others. 2025. Gui-actor: Coordinate-free visual grounding for gui agents. *arXiv preprint arXiv:2506.03143*.

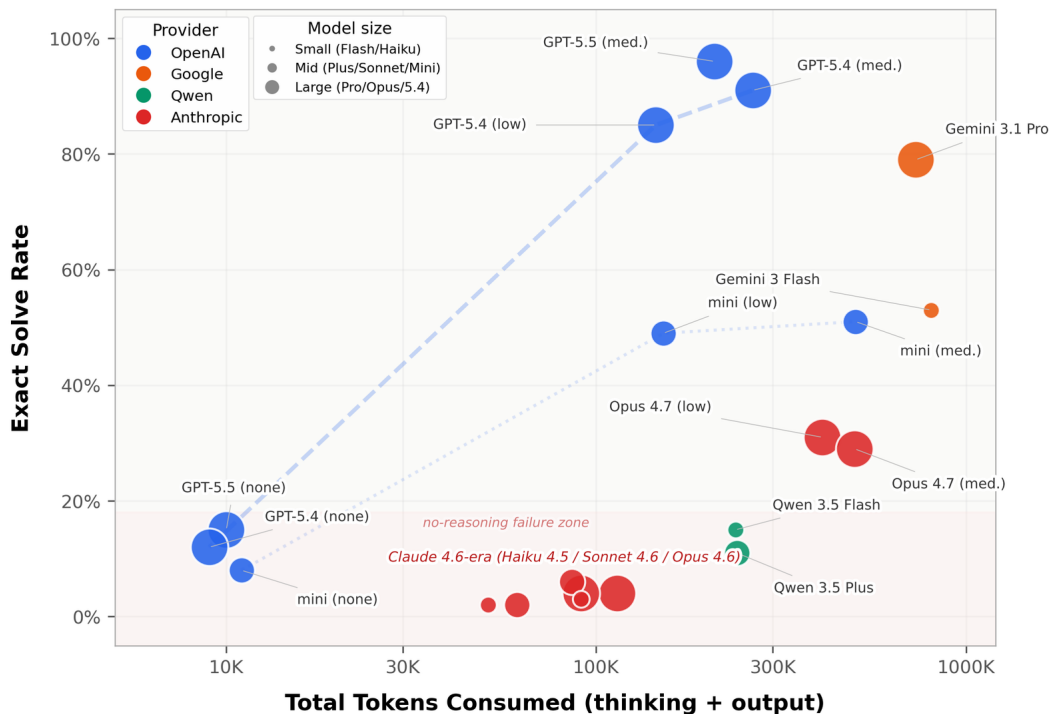
- Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and 1 others. 2024. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*.
- Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, and 1 others. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.
- Hai-Ming Xu, Qi Chen, Lei Wang, and Lingqiao Liu. 2025. Attention-driven gui grounding: Leveraging pretrained multimodal large language models without fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 8851–8859.
- Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. 2025. Aria-ui: Visual grounding for gui instructions. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22418–22433.
- Shijie Zhou, Viet Dac Lai, Hao Tan, Jihyung Kil, Wanrong Zhu, Changyou Chen, and Ruiyi Zhang. 2025. Gui-aima: Aligning intrinsic multimodal attention with a context anchor for gui grounding. *arXiv preprint arXiv:2511.00810*.
- Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

# From Pixels to BFS: High Maze Accuracy Does Not Imply Visual Planning

Alberto Rodríguez Salgado Gonzalo  
Independent Researcher  
alberto.rodriguez.salgado.97@gmail.com



## Reasoning Token Efficiency Across Frontier Models



**Figure 1: Top:** Mazes from MAZEBENCH at increasing difficulty: 5×5 (A) to 20×20 (X). **Bottom:** Solve rate vs. total tokens. The newly released GPT-5.5 sets a new Pareto point (92% at 1,096 tokens/solve); Claude 4.6-era models (red) cluster at 2–6% and Claude Opus 4.7 reaches 29–31%, still trailing the frontier on image input.

### Abstract

How do multimodal models solve visual spatial tasks—through genuine planning, or through brute-force search in token space? We introduce MAZEBENCH, a benchmark of 110 procedurally generated maze images across nine controlled groups, and evaluate 19 model configurations from OpenAI, Anthropic, Google, and Alibaba. The newly released GPT-5.5 solves 96% of the 100-maze core set and GPT-5.4 91%, but these scores are misleading: mod-

urally generated maze images across nine controlled groups, and evaluate 19 model configurations from OpenAI, Anthropic, Google, and Alibaba. The newly released GPT-5.5 solves 96% of the 100-maze core set and GPT-5.4 91%, but these scores are misleading: mod-

els typically translate images into text grids and then enumerate paths step by step, consuming 1,096–22,818 tokens per solve for a task humans do quickly. Without added reasoning budgets, all configurations score only 2–15%; on 20×20 ultra-hard mazes, even GPT-5.5 medium hits token limits and solves only 4/10. Qualitative traces reveal a common two-stage strategy: image-to-grid translation followed by token-level search, effectively BFS in prose. A text-grid ablation shows Claude Sonnet 4.6 rising from 6% to 80% and the newer Claude Opus 4.7 rising from 31% to 90% when given the correct grid, isolating weak visual extraction from downstream search across two Claude generations. When explicitly instructed not to construct a grid or perform graph search, models still revert to the same enumeration strategy. MAZEBENCH therefore shows that high accuracy on visual planning tasks does not imply human-like spatial understanding.

## 1 Introduction

Multimodal large language models (MLLMs) have achieved impressive performance across a wide range of vision-language tasks, from visual question answering to diagram understanding and mathematical reasoning in visual contexts (Lu et al., 2024; Yue et al., 2024; Fu et al., 2024a). Recent benchmarks have begun probing deeper visual capabilities, finding that models struggle with tasks that humans—even young children—solve effortlessly (Fu et al., 2024b; Tong et al., 2024; Chen et al., 2026). Yet when models *do* score well on such tasks, a crucial question remains: **does a high accuracy score mean the model actually understands the task, or is it achieving the right answer through a fundamentally different—and far less efficient—mechanism?**

We investigate this question through visual maze solving, a task that is conceptually simple for humans: given a pixel-art maze image, find the shortest path from the player to the treasure. A human glances at even a complex 20×20 maze and traces the path visually in seconds. We find that frontier MLLMs can also solve many of these mazes—GPT-5.5 achieves 96%, GPT-5.4 91%, Gemini 3.1 Pro 79%—but they do so in a qualitatively different way. Rather than spatial planning, models translate the image into a token-level grid representation and then perform *serial path enumeration*: they brute-force the solution step by step in natural language, consuming thousands of reasoning tokens for a task that requires no deliberation for a human. When

the path is too long to enumerate within the token budget, the model gives up—not because it cannot see the maze, but because it runs out of space to think.

This finding has implications beyond mazes. It suggests that benchmark accuracy alone can be misleading about the nature of model capabilities: a model may score 90% on a task while using a fundamentally different—and far more costly—cognitive strategy than the one the benchmark was designed to measure.

We design MAZEBENCH with three properties that make this analysis possible:

### **Controlled difficulty via procedural generation.**

We build a procedural maze generator producing mazes with controlled grid size (5×5 to 20×20), wall density (0–55%), trap count (0–25), border walls, and varied start/goal positions. Ground-truth shortest paths are computed via BFS. The 110 mazes are organized into nine experimental groups—including diagnostic, grid scale, wall density, trap ablation, and ultra-hard—enabling clean ablation studies (Figure 1).

**Reasoning effort as a controlled variable.** We systematically vary the reasoning budget via frontier API controls (OpenAI’s `reasoning_effort`, Anthropic’s `adaptive thinking`), producing a scaling curve from no thinking to medium effort on the *same* visual inputs.

**Token efficiency as a window into strategy.** We report the total tokens consumed (thinking + output) per solve. This reveals *how* models solve mazes, not just whether they do: GPT-5.5 low requires 1,096 tokens per solve, GPT-5.4 low 1,710, and Gemini 3 Flash 15,171—all reaching correct answers but through vastly different amounts of brute-force enumeration (Figure 1).

Our main findings are:

- 1. High scores mask brute-force strategies.** GPT-5.5 solves 96% of mazes and GPT-5.4 91%, but each consumes thousands of tokens per solve (2,172 and 2,913 respectively) in serial enumeration, with no observable spatial planning behavior. On 20×20 mazes where paths exceed the token budget, even GPT-5.5 medium drops to 4/10 (Section 5.6).
- 2. Without added reasoning budgets, performance remains very low.** All 19 configurations score only 2–15% in their no-thinking or

lowest-budget settings, even when some parse the grid correctly. For the stronger models, the failure is primarily in path planning rather than basic visual parsing (Section 5.3).

3. **The same task, radically different costs.** Models vary by  $14\times$  in tokens per solve (1,096 for GPT-5.5 low vs. 15,171 for Gemini 3 Flash), revealing that “solving a maze” means very different things computationally across providers (Section 5.5).
4. **Within a generation, size alone does not buy spatial reasoning.** Within the Claude 4.6 generation, Opus 4.6 (the largest) solves the same 4% as Haiku 4.5 (the smallest). The newer Claude Opus 4.7 closes much of the gap to frontier models (31% at low effort), and GPT-5.5 advances the OpenAI ceiling from 91% to 96% while cutting tokens-per-solve at low effort by 36%—showing that generational improvements, not raw scale within a fixed generation, are what actually move this benchmark (Section 6).

## 2 Related Work

**Visual perception gaps in MLLMs.** Several recent benchmarks have documented systematic failures in visual perception. BLINK (Fu et al., 2024b) reformats 14 classic computer vision tasks as multiple-choice questions and finds that GPT-4V achieves only 51% versus 96% for humans, concluding that perception tasks “resist mediation through natural language.” Tong et al. (2024) identify “CLIP-blind pairs”—images that vision encoders conflate despite clear visual differences—and construct the MMVP benchmark exposing failures on basic visual patterns. BabyVision (Chen et al., 2026) tests core visual abilities that human children master by age 3–6, finding that even Gemini 3 Pro scores only 49.7 versus 94.1 for adults. Notably, BabyVision includes a maze-tracing task in which models select which entrance connects to an exit from a multiple-choice list—a *perceptual tracking* task. Our benchmark differs fundamentally: rather than choosing among predefined options, models must *generate* the complete shortest path as an exact sequence of moves (U/D/L/R), requiring both visual parsing and multi-step spatial planning. This distinction allows us to show that failures compound across *both* stages: some models (Claude) fail primarily at visual grid extraction, while others (GPT-5.4, Gemini) parse the

grid correctly but still resort to brute-force token enumeration rather than spatial planning.

**Multimodal reasoning benchmarks.** MathVista (Lu et al., 2024) and MathVerse (Zhang et al., 2024) evaluate mathematical reasoning in visual contexts, finding that models often rely on textual cues rather than diagram understanding. EMMA (Hao et al., 2025) tests cross-modal reasoning across math, physics, chemistry, and coding, reporting that even chain-of-thought prompting and test-time compute scaling underperform (ICML 2025). MMMU (Yue et al., 2024) and MME (Fu et al., 2024a) provide comprehensive evaluation suites spanning dozens of disciplines, while MMEvalPro (Huang et al., 2025) addresses systematic biases in multiple-choice evaluation by introducing perception prerequisite questions. Chen et al. (2024b) audit evaluation methodology itself, questioning whether current benchmarks measure the capabilities they claim to. Our benchmark complements this body of work by targeting a single, tightly controlled task—visual pathfinding—that isolates spatial reasoning from domain knowledge.

**Spatial reasoning in MLLMs.** Spatial reasoning has emerged as a key evaluation axis for multimodal models. SpatialVLM (Chen et al., 2024a) endows VLMs with metric spatial reasoning via synthetic data, while SpatialRGPT (Cheng et al., 2024) grounds spatial reasoning in depth-aware representations. SpatialBench (Xu et al., 2025) decomposes spatial intelligence into five cognitive levels and finds that models fail at high-level planning. SpatiaLab (Tong et al., 2026) evaluates spatial reasoning in unconstrained real-world images, finding that even GPT-5-mini scores only 41% versus 65% for humans. GSR-Bench (Rajabi and Kosecka, 2024) evaluates grounded spatial relationship understanding across 27 models at NeurIPS 2024. VGRP-Bench (Ren et al., 2025) is the most directly related benchmark to ours, testing vision-language models on grid-based visual reasoning puzzles. Our work differs in three ways: (1) we introduce reasoning effort as a controlled experimental variable, (2) we report thinking token efficiency as a metric, and (3) we demonstrate through qualitative analysis that models solve grid puzzles through brute-force token-level enumeration, with no evidence of human-like spatial planning.

**Test-time compute scaling.** Snell et al. (2025) demonstrate that scaling inference-time computa-

tion can be more effective than scaling model parameters for reasoning tasks (ICLR 2025). Agarwal et al. (2025) show that no single test-time scaling strategy universally dominates but that performance scales monotonically with compute budget. Chain-of-Visual-Thought (Qin et al., 2025) and related methods (Wang et al., 2025) extend chain-of-thought reasoning to continuous visual tokens. Our reasoning effort sweep provides direct empirical evidence for these theoretical results: GPT-5.4 improves from 12% to 85% to 91%, and GPT-5.5 from 15% to 92% to 96%, as reasoning effort increases from none to low to medium, with diminishing returns at higher budgets.

### 3 Benchmark Design

#### 3.1 Task Formulation

Given a pixel-art maze image, the model must return a JSON object containing: the grid size, whether the start and goal are found, whether a path exists (*reachable*), the shortest path length, and the exact path as a list of directional moves (U, D, L, R). A maze is scored as *solved* only when all three conditions hold: (1) reachability is correctly identified, (2) the shortest path length is correct, and (3) the returned path exactly matches one of the accepted shortest-path annotations. No partial credit is awarded.

#### 3.2 Procedural Maze Generation

We build a procedural generator that produces mazes with controlled parameters. Each maze is defined by a grid size ( $r \times c$ ), wall density  $d \in [0, 0.55]$  (fraction of candidate cells converted to walls), trap count  $t$  (impassable hazard tiles visually distinct from walls), and optional border walls (a wall ring around the outer edge). Start and goal positions are randomized on opposite edges with a minimum Manhattan distance of  $\lfloor (r + c)/3 \rfloor$ .

The generation algorithm places walls incrementally, verifying after each placement that the maze remains reachable (unless the maze is designated unreachable). Traps are placed similarly with reachability checks. Ground-truth shortest paths are computed via breadth-first search with multi-parent tracking, enumerating all optimal paths (capped at 50). All mazes render as  $1024 \times 1024$  pixel PNG images using procedurally generated pixel-art sprites across four visual palettes (forest, desert, dungeon, meadow).

#### 3.3 Dataset Structure

The benchmark contains 110 mazes organized into nine groups:

- **Group A: Diagnostic** (8 mazes). Empty or near-empty grids with straight-line paths. If a model fails here, the bottleneck is visual parsing, not reasoning.
- **Group B: Grid Scale** (15). Constant wall density (25%), grid sizes from  $5 \times 5$  to  $13 \times 13$ . Isolates the effect of spatial scale.
- **Group C: Wall Density** (15). Constant  $9 \times 9$  grid, density swept from 0% to 45%. Isolates obstacle complexity.
- **Group D: Trap Ablation** (12). Six matched pairs sharing the same random seed—one with traps, one without—isolating trap recognition.
- **Group E: Unreachable** (14). All unreachable, spanning  $5 \times 5$  to  $13 \times 13$ . Tests false-positive rate for reachability claims.
- **Group F: Border Walls** (10). Five matched pairs with/without border walls. Tests whether visual framing affects parsing.
- **Group G: Combined Hard** (16). Large grids ( $9 \times 9$ – $13 \times 13$ ), high density, traps, and borders combined.
- **Group H: Palette Stress** (10). Same maze structure rendered in all four palettes. Tests visual style sensitivity.
- **Group X: Ultra-Hard** (10).  $20 \times 20$  grids with 8–25 traps, 35–55% wall density, and shortest paths of 28–42 steps.

Overall the dataset contains 79 reachable mazes (72% of the 110) and 31 unreachable (28%). Shortest path lengths range from 4 to 42 moves (mean 13.5, median 12).

### 4 Experimental Setup

**Models.** We evaluate models from four providers: **OpenAI:** GPT-5.4 and GPT-5.4-mini, each at three reasoning effort levels (none, low, medium), plus GPT-5.5 (released April 2026) at none, low, and medium effort; **Anthropic:** Claude Opus 4.6, Sonnet 4.6, and Haiku 4.5, each at no-thinking and low-effort configurations, plus Claude Opus 4.7 (released April 2026) at low- and medium-effort; **Google:** Gemini 3.1 Pro Preview and Gemini 3 Flash Preview; **Alibaba/Qwen:**

Qwen 3.5 Plus and Qwen 3.5 Flash via DashScope. This yields 19 model configurations in total.

**Protocol.** Every configuration receives the same fixed prompt instructing JSON-only output with no tool use. Images are sent as base64-encoded data URLs. We disable structured output enforcement and tool calling across all APIs so that models must reason freely. Failed JSON parses are retried up to twice.

**Reasoning effort control.** For OpenAI, we use the `reasoning_effort` parameter (none/low/medium). For Anthropic, we test both no-thinking (omitting the thinking configuration) and adaptive thinking with low effort. We evaluate Claude 4.6-era models only at no-thinking and low-effort because the text-grid ablation (Section 6) establishes that their failure is in visual extraction, not downstream search—additional reasoning tokens applied to a misidentified grid yield no improvement. For Claude Opus 4.7 we additionally include a medium-effort run, since its substantially higher solve rate (29–31% vs. 4% for Opus 4.6) suggests the visual-extraction bottleneck has partially lifted and makes effort scaling on the image task a meaningful comparison. Gemini and Qwen models use default configurations; notably, Gemini performs hidden internal reasoning (visible via `thoughtsTokenCount` in the API response) that cannot be disabled.

**Reproducibility.** The main experiments were conducted between March 20–23, 2026, with the Claude Opus 4.7 runs added on April 16, 2026 and the GPT-5.5 runs on April 24–25, 2026, shortly after each model’s release. The following API model identifiers were used: `gpt-5.5`, `gpt-5.4`, and `gpt-5.4-mini` (OpenAI Responses API), `claude-opus-4-7`, `claude-opus-4-6`, `claude-sonnet-4-6`, and `claude-haiku-4-5-20251001` (Anthropic Messages API, version 2023-06-01), `gemini-3.1-pro-preview` and `gemini-3-flash-preview` (Gemini REST API), and `qwen3.5-plus` and `qwen3.5-flash` (DashScope API). Temperature was set to 0.0 for all non-thinking configurations; Anthropic requires temperature 1.0 when thinking is enabled.

Model	Solved	Reach.%	Lat.(s)
<b>GPT-5.5 (medium)</b>	<b>96/100</b>	<b>99</b>	28.6
GPT-5.5 (low)	92/100	99	13.5
GPT-5.4 (medium)	91/100	95	38.1
GPT-5.4 (low)	85/100	92	23.8
Gemini 3.1 Pro	79/100	86	51.9
Gemini 3 Flash	53/100	82	32.3
GPT-5.4-mini (medium)	51/100	86	21.8
GPT-5.4-mini (low)	49/100	89	7.3
Opus 4.7 (low)	31/100	74	29.1
Opus 4.7 (medium)	29/100	72	39.7
GPT-5.5 (none)	15/100	74	2.4
Qwen 3.5 Flash	15/100	71	12.2
GPT-5.4 (none)	12/100	71	2.1
Qwen 3.5 Plus	11/100	78	23.7
GPT-5.4-mini (none)	8/100	79	1.3
Sonnet 4.6 (none)	6/100	70	16.9
Opus 4.6 (low)	4/100	67	21.3
Opus 4.6 (none)	4/100	71	19.8
Haiku 4.5 (low)	3/100	63	9.1
Sonnet 4.6 (low)	2/100	60	14.7
Haiku 4.5 (none)	2/100	68	9.4

Table 1: Main leaderboard on the 100-maze core set. **Solved** requires correct reachability, correct shortest path length, and exact path match. **Reach.%** is reachability accuracy. **Lat.** is average latency per maze.

## 5 Results

### 5.1 Main Results

Table 1 presents the full leaderboard ranked by solve rate on the 100-maze core set (excluding Group X ultra-hard).

The results reveal a clear hierarchy. The newly released GPT-5.5 with medium reasoning takes the top spot at 96%, followed by GPT-5.5 low (92%), GPT-5.4 medium (91%), GPT-5.4 low (85%), and Gemini 3.1 Pro (79%). GPT-5.5 advances the OpenAI ceiling by 5 points over GPT-5.4 at matched effort while also raising reachability accuracy to 99% (vs. 95% for GPT-5.4 medium). All Claude 4.6-era models—Opus 4.6, Sonnet 4.6, and Haiku 4.5—score between 2–6% regardless of size or thinking configuration, and enabling low-effort thinking sometimes *degrades* performance (Sonnet drops from 6% to 2%). The subsequently released Claude Opus 4.7 changes this picture partially: it jumps to 31% at low effort and 29% at medium effort—a roughly 8× improvement over Opus 4.6 at matched effort—but still trails GPT-5.5 low (92%) and GPT-5.4 low (85%) by a wide margin. Curiously, Opus 4.7 also exhibits inverse effort scaling in this range: medium effort solves *fewer* mazes than low while consuming 22% more tokens, echoing the pattern seen in the 4.6-era Sonnet low/none comparison.

Model	Reason.	Solves per Group								Total
		A	B	C	D	E	F	G	H	
		/8	/15	/15	/12	/14	/10	/16	/10	
GPT-5.5	med.	8	13	13	12	14	10	16	10	<b>96</b>
GPT-5.5	low	8	15	11	10	14	10	14	10	<b>92</b>
GPT-5.5	none	8	2	0	0	2	0	3	0	15
GPT-5.4	med.	7	14	13	12	13	9	13	10	<b>91</b>
GPT-5.4	low	8	13	10	11	12	8	13	10	<b>85</b>
GPT-5.4	none	8	1	1	1	1	0	0	0	12
GPT-5.4-mini	med.	7	11	8	6	7	3	5	4	51
GPT-5.4-mini	low	6	9	8	5	11	3	3	4	49
GPT-5.4-mini	none	5	0	0	1	2	0	0	0	8
Gemini 3.1 Pro	default	8	12	14	9	9	7	10	10	<b>79</b>
Gemini 3 Flash	default	7	7	9	7	6	5	7	5	<b>53</b>
Qwen 3.5 Flash	default	8	1	2	1	2	0	1	0	15
Qwen 3.5 Plus	default	6	1	4	0	0	0	0	0	11
Sonnet 4.6	none	3	0	1	1	0	0	0	1	6
Sonnet 4.6	low	1	0	1	0	0	0	0	0	2
Opus 4.6	none	2	0	1	0	1	0	0	0	4
Opus 4.6	low	2	0	1	0	1	0	0	0	4
Haiku 4.5	none	1	1	0	0	0	0	0	0	2
Haiku 4.5	low	3	0	0	0	0	0	0	0	3

Table 2: Per-group solve counts for all 19 model configurations. Groups: **A**=Diagnostic, **B**=Grid Scale, **C**=Wall Density, **D**=Trap Ablation, **E**=Unreachable, **F**=Border Walls, **G**=Combined Hard, **H**=Palette Stress. GPT-5.5 medium is the first configuration to clear Group G (Combined Hard) at full 16/16. GPT-5.4’s none→low transition (+73 solves) is the largest single improvement; GPT-5.5 shows a similar +77 jump (15→92). All Claude models remain at 2–6 regardless of reasoning effort. Gemini models achieve strong results via hidden internal thinking.

## 5.2 Reasoning Effort Scaling

Table 2 presents per-group solve counts across all 16 model configurations, revealing both the effect of reasoning effort and stark cross-provider differences.

For GPT-5.4, the transition from no reasoning to low effort produces a dramatic +73 solve improvement (12→85), while increasing to medium adds only +6 (85→91), exhibiting clear diminishing returns. The hard group (G) plateaus at 13/16 at both low and medium, suggesting that the remaining failures require qualitatively different capabilities rather than more reasoning tokens. GPT-5.5 shows the same scaling shape but at a higher ceiling: none→low adds +77 (15→92) and low→medium adds another +4 (92→96), with the hard group (G) finally clearing at 16/16 at medium effort—the first model in our benchmark to do so.

In contrast, all Claude models remain flat at 2–6 solves regardless of reasoning configuration—Sonnet actually *degrades* from 6 to 2 with low effort enabled. Gemini models, which perform hidden reasoning by default, achieve results between GPT-5.4’s low and medium configurations without any user-controllable effort setting. Qwen models show modest performance (11–15), with

the smaller Flash variant slightly outperforming Plus—the only family where the smaller model does better.

## 5.3 Diagnostic Analysis: Vision Works, Reasoning Doesn’t

Group A (diagnostic) serves as a critical control. These eight mazes have zero or near-zero walls with straight-line paths—any model that can parse the grid should solve them. Both GPT-5.4 and GPT-5.5 achieve 8/8 *even without reasoning*, confirming that their vision encoders correctly identify the grid, start position, goal position, and tile types. GPT-5.4’s collapse from 8/8 on diagnostics to 1/15 on Group B (which adds only 25% wall density) demonstrates that the failure is entirely in path planning, not visual parsing; GPT-5.5 follows the same pattern (8/8 → 2/15 without reasoning).

Interestingly, Claude models struggle even on diagnostics: Opus solves only 2/8 and Haiku 1/8 without thinking. This suggests that Claude’s vision pipeline has additional limitations in grid parsing that compound with the reasoning deficit.

Model	Solved	Tot. Tok.	Tok/Solve
<b>GPT-5.5 (medium)</b>	<b>96</b>	209K	2,172
<b>GPT-5.5 (low)</b>	<b>92</b>	101K	<b>1,096</b>
GPT-5.4 (medium)	91	265K	2,913
GPT-5.4 (low)	85	145K	1,710
Gemini 3.1 Pro	79	731K	9,250
Gemini 3 Flash	53	804K	15,171
GPT-5.4-mini (medium)	51	503K	9,872
GPT-5.4-mini (low)	49	152K	3,105
Opus 4.7 (low)	31	409K	13,189
Opus 4.7 (medium)	29	500K	17,239
GPT-5.5 (none)	15	10K	668
Qwen 3.5 Flash	15	238K	15,835
GPT-5.4 (none)	12	9K	782
Qwen 3.5 Plus	11	240K	21,804
GPT-5.4-mini (none)	8	11K	1,427
Opus 4.6 (none)	4	91K	22,818

Table 3: Thinking token efficiency. **Tot. Tok.** is total thinking + output tokens across all 100 mazes. **Tok/Solve** is the average total tokens consumed per correctly solved maze. Lower is more efficient.

#### 5.4 Unreachable Detection

Unreachable mazes probe a different failure mode: can models recognize when no path exists? Without reasoning, all models exhibit a strong bias toward claiming reachability—GPT-5.4 produces 25/28 false positives (89% false-positive rate). With medium reasoning, this drops to 6/28 (21%), and unreachable detection reaches 93% recall. This suggests that detecting impossibility is itself a reasoning-intensive task: the model must exhaustively verify that no path exists rather than optimistically reporting one.

#### 5.5 Thinking Token Efficiency

Figure 1 and Table 3 report total tokens consumed (thinking + output) per solve, revealing order-of-magnitude differences in reasoning efficiency.

GPT-5.5 at low effort is the new Pareto-optimal configuration: 92 solves at **1,096** tokens per solve—a 36% reduction over the previous Pareto point (GPT-5.4 low, 1,710 tok/solve) while solving 7 more mazes. Medium effort on GPT-5.5 buys another 4 solves (92→96) at roughly twice the token cost (1,096→2,172). GPT-5.4 medium remains a strong configuration but is dominated by GPT-5.5 on both axes: GPT-5.5 low ties or beats GPT-5.4 medium’s solve rate (92 vs. 91) at 38% fewer tokens per solve. Gemini 3 Flash consumes 7,186 thinking tokens per maze internally (visible via the API’s `thoughtsTokenCount` field) but achieves only 53% solve rate—14× more total tokens than GPT-5.5 low for 39 fewer solves. Claude 4.6-era

Maze	GT	Pred.	Path	Solved	Lat.
101	✓	–	40 → –	△	137s
102	✓	✓	30 → 30		75s
103	✓	✓	42 → 42		103s
104	×	×	–	✓	101s
105	✓	–	32 → –	△	87s
106	✓	✓	37 → 37	✓	88s
107	×	×	–	✓	61s
108	✓	✓	28 → 28		100s
109	✓	×	41 → –		77s
110	×	×	–	✓	87s
Total solved				<b>4/10</b>	91s

Table 4: GPT-5.5 (medium) on the 10 ultra-hard 20×20 mazes. **GT** = ground-truth reachability; **Pred.** = model prediction; **Path** = GT length → predicted length. ✓ = correct, × = wrong, △ = correct reachability but incomplete output (hit token limit). The model solves one reachable maze (37 steps) plus all three unreachable cases; it falsely declares the 41-step reachable maze unreachable and runs out of tokens on the 40- and 32-step mazes.

models are the least efficient, spending 22,000–30,000 tokens per solve on verbose but incorrect outputs; Opus 4.7 improves to 13,000–17,000 tokens per solve, narrowing but not closing the gap to GPT-5.5 low (1,096 tokens per solve).

A key revelation is that Gemini models perform hidden reasoning by default: Gemini 3 Flash uses a median of 7,861 thinking tokens per maze despite having no user-configurable reasoning toggle. The Gemini–Claude gap, however, is not solely explained by thinking tokens. As our qualitative analysis in Section 6 shows, Claude models produce inaccurate grid extractions (wrong dimensions, misplaced walls), meaning they brute-force on a *hallucinated* grid—a compounding failure where poor vision quality renders even extensive reasoning futile.

#### 5.6 Ultra-Hard Ceiling

Group X (20 × 20 grids, 8–25 traps, paths of 28–42 steps) tests the absolute ceiling of current models. Table 4 shows per-maze results for GPT-5.5 at medium effort—the best-performing configuration on the core set.

GPT-5.5 drops from 96% on the core set to **4/10** on ultra-hard—a one-solve improvement over GPT-5.4 medium (3/10) but the same qualitative ceiling. Average latency is 91 seconds (versus 28.6s on the core set), and 2/10 mazes hit the maximum output token limit (8,192 tokens). GPT-5.5 cleanly solves all three unreachable ultra-hard mazes (full

JSON output, correct reach); GPT-5.4 medium also identified them as unreachable but hit the token limit on two of them ( $\Delta$ ) and was credited with only one (107). Beyond that, GPT-5.5 still falsely declares the 41-step reachable maze unreachable and produces wrong paths on the 28-, 30-, and 42-step reachable mazes despite getting their lengths right. The pattern is consistent: longer reachable paths exceed the brute-force budget even for the strongest model.

This provides direct evidence that models solve mazes through serial enumeration bounded by token budget: when the path exceeds what can be brute-forced within the thinking allocation, the model fails—and the failure mode is the same for both GPT-5.4 and GPT-5.5.

## 6 Analysis and Discussion

**A universal two-stage strategy.** Examination of model outputs reveals that *all* models—regardless of provider or accuracy—follow the same two-stage strategy: (1) **Image-to-grid translation**: the model converts the visual maze into a textual row-column matrix, and (2) **Serial path enumeration**: the model attempts to trace paths step-by-step through this textual grid. What differentiates high-performing models from low-performing ones is primarily the quality of Stage 1, not the sophistication of Stage 2.

Claude models expose this strategy clearly because their reasoning traces are visible in the output. On `gen_maze_014` (an  $8 \times 8$  grid), Opus 4.6 outputs a full grid transcription before searching:

```
Row 0: W, open, W, W, open, open, ...G
Row 1: W, S, open, W, W, W, open, ...
[...maps all 10 rows...]
Path: R,R,R,...(1,3) wait, (1,3) is wall.
Alternative: (1,1)→(1,2)→(2,2)→(2,3)→...
That's: R, D, R, R, R, R, R, R, U, U = 11
Shorter: ... = 11 moves. Or: ... = 15, longer.
Stick with 11.
```

This is textbook brute-force search in natural language: try a path, hit a wall, backtrack, try another, count steps, compare. Notably, our prompt explicitly instructs models not to “use any external tools, code, search, calculators, or graph-search programs”—yet the models’ only available strategy is to *simulate* a graph-search algorithm (BFS) in natural language tokens, step by step. Critically, both Opus and Sonnet **misidentify the grid as  $10 \times 10$**  (it is actually  $8 \times 8$ ), leading them to rea-

son over a hallucinated grid and produce incorrect paths. GPT-5.4 correctly identifies  $8 \times 8$  and Gemini 3 Flash also reports the correct grid size—their Stage 1 is more accurate, which makes their Stage 2 brute-force search succeed on the correct grid.

**The performance gap is in vision, not reasoning strategy.** This two-stage analysis reframes the cross-provider performance differences. Claude models do not fail because they use a worse *reasoning* strategy—they use the same enumerate-and-backtrack approach as GPT-5.4. They fail because their *grid extraction* is unreliable: wrong grid dimensions, mislocated walls, and hallucinated openings. On the trivially empty diagnostic maze `gen_maze_001` ( $5 \times 5$ , zero walls), Sonnet 4.6 reports a “ $6 \times 6$  grid” instead of the correct  $5 \times 5$  and computes a path of length 4—the correct length, but from a misidentified grid. Opus 4.6 reports “ $7 \times 7$ ” and returns a path of length 5, both incorrect (the grid is  $5 \times 5$  with shortest path 4).

On the unreachable maze `gen_maze_057`, this vision deficit is catastrophic: Opus maps a  $9 \times 9$  grid as  $11 \times 11$  with incorrectly placed walls, then brute-forces a 14-step path *through cells that are actually walls*—confidently declaring the maze reachable when it is not. GPT-5.4 without reasoning correctly reports this maze as unreachable in 2.6 seconds, demonstrating superior visual parsing even without any chain-of-thought.

**High accuracy still means brute-force.** The crucial insight is that even the best-performing models—GPT-5.5 at 96%, GPT-5.4 at 91%, Gemini 3.1 Pro at 79%—are still brute-forcing. They simply brute-force on a *correct* grid. The evidence is threefold: (1) Performance scales with thinking token budget, not with any spatial heuristic—adding more tokens yields more solves, in a pattern consistent with longer serial search; this holds for GPT-5.5 (15→92→96) just as it does for GPT-5.4. (2) On ultra-hard  $20 \times 20$  mazes, GPT-5.5 medium still bottoms out at 4/10, with the same failure mode as GPT-5.4—hitting the token limit on the longest reachable paths and defaulting to “unreachable.” A human traces the same maze visually in seconds. (3) Token efficiency varies by  $14 \times$  across providers (1,096 for GPT-5.5 low vs. 15,171 for Gemini 3 Flash), consistent with different search strategies expressed in text but inconsistent with human-like spatial planning, which should not require proportionally more computation for the same visual structure.

This is fundamentally different from human maze solving. Humans engage spatial perception directly—tracing paths with their eyes, recognizing dead ends holistically, and planning routes without step-by-step enumeration. Models convert vision to language, then search through language. The conversion step is lossy (especially for Claude), and the search step is computationally expensive relative to the task’s intrinsic difficulty.

**Can prompting induce visual reasoning?** A natural question is whether the brute-force strategy is a prompt artifact—perhaps the standard prompt implicitly encourages grid-based reasoning. We test this with a modified “visual-intuition” prompt that explicitly instructs: *“Do NOT convert the maze into a text grid, matrix, or row/column representation. Do NOT perform step-by-step BFS, DFS, or any graph-search algorithm in text. Instead, solve this the way a human would: look at the image, visually trace the walkable path.”*

On 20 representative mazes, GPT-5.4 (low) scores 17/20 with the visual prompt versus 19/20 with the standard prompt—slightly *worse*, not better. More revealing is what happens when we apply the visual prompt to Claude Sonnet 4.6, whose reasoning traces are visible. Despite the explicit prohibition, Sonnet immediately falls back to grid enumeration:

```
Let me map the walls (brown bricks) and
open paths (tan/sand tiles):
Row 0: walls at col 0, col 2; open col 3-8
Row 1: Start at col 1; walls col 1-2, 3-5
[...continues mapping all rows...]
```

The model does not comply with the instruction to reason visually; it falls back to grid enumeration—exactly the strategy the prompt forbids—and silently violates the constraint rather than reporting that it cannot solve the task under the given restrictions. This raises a broader concern about instruction compliance: when a model lacks the capability to follow a constraint, it proceeds with a forbidden strategy rather than failing gracefully.

**Text-grid ablation confirms vision is the bottleneck—for both Claude generations.** To directly test whether Claude’s failure is in vision or reasoning, we bypass the image encoder entirely: we provide the model with the *correct text grid* (using S, G, ., #, T symbols) instead of the maze image. We run this ablation on Sonnet 4.6 and on

Model	Input	Solved	Tok/Solve	Lat.
Sonnet 4.6	Image (none)	6/100	14,313	16.9s
Sonnet 4.6	Text grid (low)	80/100	3,222	34.8s
Opus 4.7	Image (low)	31/100	13,189	29.1s
Opus 4.7	Text grid (low)	<b>90/100</b>	<b>2,285</b>	18.7s

Table 5: Image vs. text grid on 100 mazes for two Claude generations. Bypassing vision yields 13× more solves for Sonnet 4.6 and 2.9× more for Opus 4.7. Opus 4.7 on text grids reaches 90%—matching GPT-5.4 medium on images (91%) at *lower* tokens per solve (2,285 vs. 2,913).

the newly released Opus 4.7 to test whether the partial image-input improvement in the 4.7 generation reflects better vision, better reasoning, or both. Table 5 shows the results.

With low-effort reasoning on the text grid, Sonnet 4.6 solves **80/100** mazes (13× image input) and Opus 4.7 reaches **90/100**, matching GPT-5.4 medium on images at fewer tokens per solve (2,285 vs. 2,913). Token efficiency improves 4.4× for Sonnet (14,313→3,222) and 5.8× for Opus 4.7 (13,189→2,285). The Opus 4.7 result is especially informative: even though the 4.7 generation has clearly improved grid extraction (31% on images vs. 4% for Opus 4.6), its text-grid score still exceeds its image score by 59 points—showing that vision remains the dominant bottleneck. Across *both* Claude generations the reasoning engine is competitive with the best models when given accurate spatial input; the gap on images lives almost entirely in the image-to-grid translation stage.

**Generation, not scale, drives progress; vision still gates cross-provider ordering.** Within the Claude 4.6 generation, Opus 4.6 solves only 4/100—tied with Haiku 4.5 and consistent with Claude’s MMMU gap (Yue et al., 2024). The 4.7 generation closes much of this distance (31% on images, 90% on text grid), showing that progress comes from generational improvements in image-to-grid extraction, not scale within a fixed generation. Gemini’s advantage over 4.6-era Claude (53–79% vs. 2–6%) is also partly explained by hidden default reasoning—Gemini 3 Flash spends a median of 7,861 thinking tokens per maze—but the cross-provider ordering on image inputs still tracks vision quality more than raw reasoning capacity.

## 7 Conclusion

We introduced MAZEBENCH, a benchmark of 110 procedurally generated visual mazes that probes not just *whether* multimodal models can solve spatial tasks, but *how* they do so.

Our main message is simple: **high accuracy on visual planning tasks can be misleading**. The strongest models in our benchmark do solve many mazes, but they do not appear to solve them the way humans do. Instead, they first translate pixels into a textual grid and then search for a path in token space, step by step. This strategy is computationally expensive, brittle under scale, and qualitatively different from rapid visual path tracing.

This distinction matters for evaluation. GPT-5.5 reaches 96% on the 100-maze core set and GPT-5.4 91%, yet both still rely on serial token-level search and collapse on ultra-hard mazes when the search no longer fits within the token budget—GPT-5.5 medium solves only 4/10 ultra-hard mazes despite cutting tokens-per-solve by 36% on the core set. Claude models expose the same search strategy even more clearly, but often on incorrect grid extractions; when we replace the image with the correct text grid, Sonnet 4.6 jumps from 6% to 80% and Opus 4.7 from 31% to 90%, showing that weak performance can come from poor visual extraction while strong performance can still come from brute-force planning on a correct grid.

Taken together, our results suggest that benchmark scores should not be read as evidence of human-like spatial understanding. They are evidence only that a model can eventually produce the correct answer under a particular compute budget. To measure genuine multimodal progress, we need evaluations that track not just correctness, but also *strategy, efficiency, and failure mode*.

## Limitations

MAZEBENCH relies on procedurally generated pixel-art mazes and proprietary API-based models, and cross-provider reasoning controls (none/low/medium vs. Anthropic adaptive thinking vs. Gemini’s hidden internal reasoning) are not perfectly comparable. We do not include a human-timing baseline, and we evaluate each configuration with a single seed; the resulting solve counts therefore lack confidence intervals. Future work should extend the setup to open-weight models, human-timing baselines, multi-seed evaluation, and broader visual planning tasks.

## References

- Aradhye Agarwal, Ayan Sengupta, and Tanmoy Chakraborty. 2025. The art of scaling test-time compute for large language models. *arXiv preprint arXiv:2512.02008*.
- Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024a. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Liang Chen, Weichu Xie, Yiyang Liang, Hongfeng He, Hans Zhao, Zhibo Yang, and Ziqi Huang. 2026. BabyVision: Visual reasoning beyond language. *arXiv preprint arXiv:2601.06521*.
- Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. 2024b. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37.
- An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. 2024. SpatialRGPT: Grounded spatial reasoning in vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. 2024a. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.
- Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024b. BLINK: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision (ECCV)*, pages 148–166.
- Yunzhuo Hao, Jiawei Gu, Huichen Will Wang, Linjie Li, Zhengyuan Yang, Lijuan Wang, and Yu Cheng. 2025. Can MLLMs reason in multimodality? EMMA: An enhanced multimodal reasoning benchmark. In *International Conference on Machine Learning (ICML)*.
- Jinsheng Huang, Liang Chen, Taian Guo, Fu Zeng, Yusheng Zhao, Bohan Wu, Ye Yuan, Haozhe Zhao, Zhihui Guo, Yichi Zhang, Jingyang Yuan, Wei Ju, Luchen Liu, Tianyu Liu, Baobao Chang, and Ming Zhang. 2025. MMEvalPro: Calibrating multimodal benchmarks towards trustworthy and efficient evaluation. In *Proceedings of the Conference of the Nations of the Americas Chapter of the ACL (NAACL)*.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. MathVista: Evaluating mathematical reasoning of

- foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*.
- Yiming Qin, Bomin Wei, Jiaxin Ge, Konstantinos Kallidromitis, Stephanie Fu, Trevor Darrell, and Xudong Wang. 2025. Chain-of-visual-thought: Teaching VLMs to see and think better with continuous visual tokens. *arXiv preprint arXiv:2511.19418*.
- Navid Rajabi and Jana Kosecka. 2024. GSR-Bench: A benchmark for grounded spatial reasoning evaluation via multimodal LLMs. In *NeurIPS 2024 Workshop on Compositional Learning*.
- Yufan Ren, Konstantinos Tertikas, Shalini Maiti, Junlin Han, Tong Zhang, Sabine Süssstrunk, and Filippos Kokkinos. 2025. VGRP-Bench: Visual grid reasoning puzzle benchmark for large vision-language models. *arXiv preprint arXiv:2503.23064*.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. In *International Conference on Learning Representations (ICLR)*.
- Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. Eyes wide shut? Exploring the visual shortcomings of multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Shengbang Tong, Yuexiang Zhai, Lingjie Liu, Yann LeCun, Yi Ma, and Saining Xie. 2026. SpatialLab: Can vision-language models perform spatial reasoning in the wild? In *International Conference on Learning Representations (ICLR)*.
- Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. 2025. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*.
- Peiran Xu, Sudong Wang, Yao Zhu, Jianing Li, and Yunjian Zhang. 2025. SpatialBench: Benchmarking multimodal large language models for spatial cognition. *arXiv preprint arXiv:2511.21471*.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9556–9567.
- Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. 2024. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? In *European Conference on Computer Vision (ECCV)*, pages 169–186.

## A Complete Maze Dataset by Group

Figures 2–10 show every maze in the benchmark organized by experimental group.

### A: Diagnostic



Figure 2: Group A: Diagnostic (8 mazes). Empty or near-empty grids with trivial straight-line paths.

### B: Grid Scale

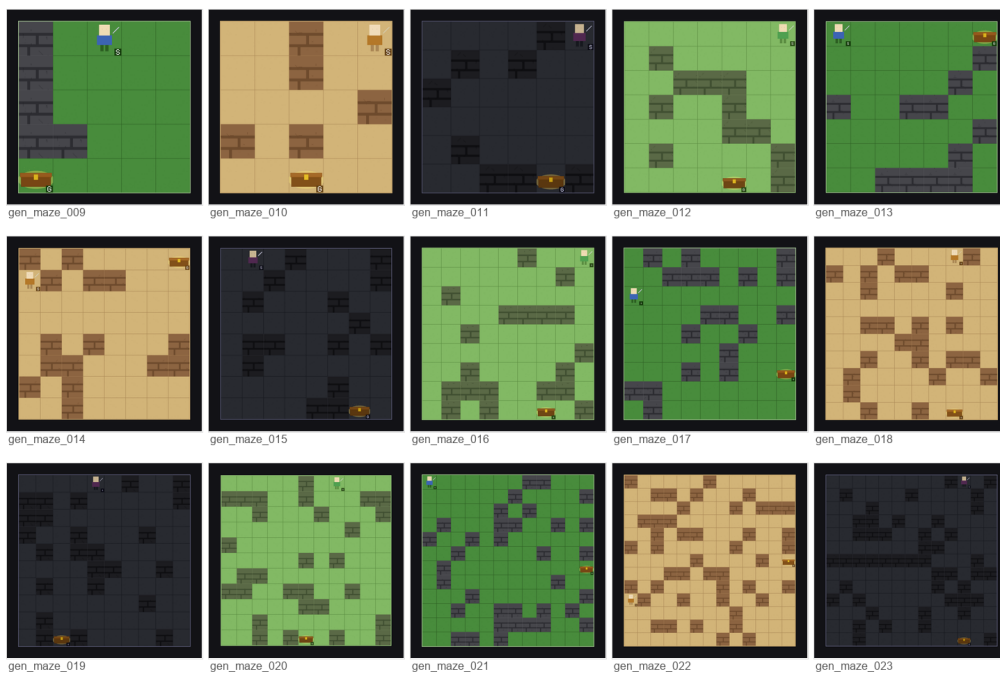


Figure 3: Group B: Grid Scale (15 mazes). Constant 25% wall density, grid sizes from  $5 \times 5$  to  $13 \times 13$ .

**C: Wall Density**

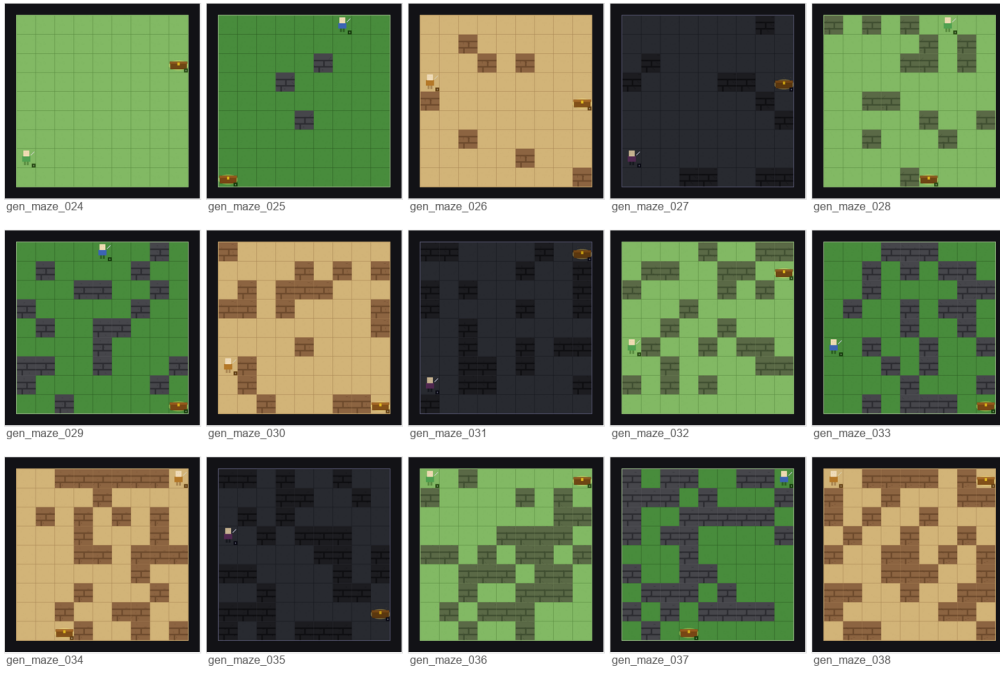


Figure 4: Group C: Wall Density (15 mazes). Constant  $9 \times 9$  grid, density from 0% to 45%.

**D: Trap Ablation**

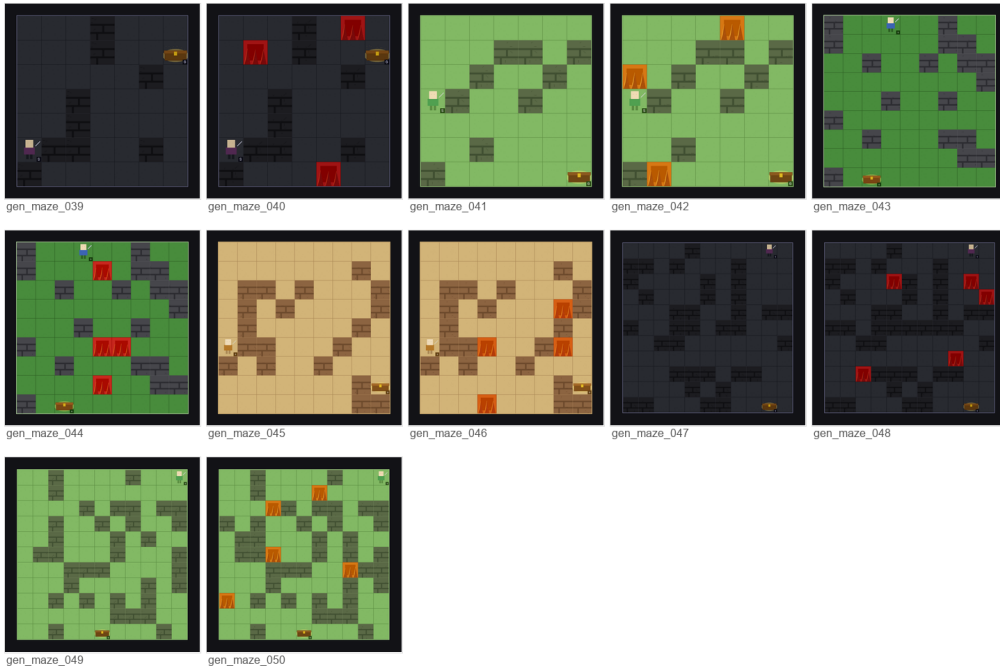


Figure 5: Group D: Trap Ablation (12 mazes). Six matched pairs—control (no traps) and treatment (with traps)—sharing the same random seed.

**E: Unreachable**

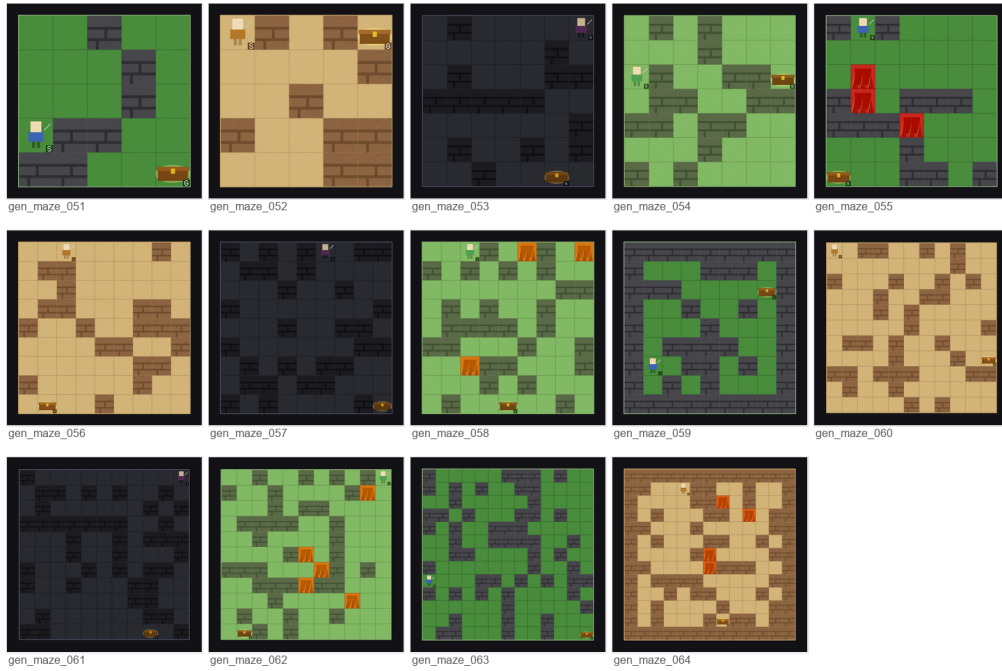


Figure 6: Group E: Unreachable (14 mazes). All mazes have no valid path from start to goal.

**F: Border Walls**

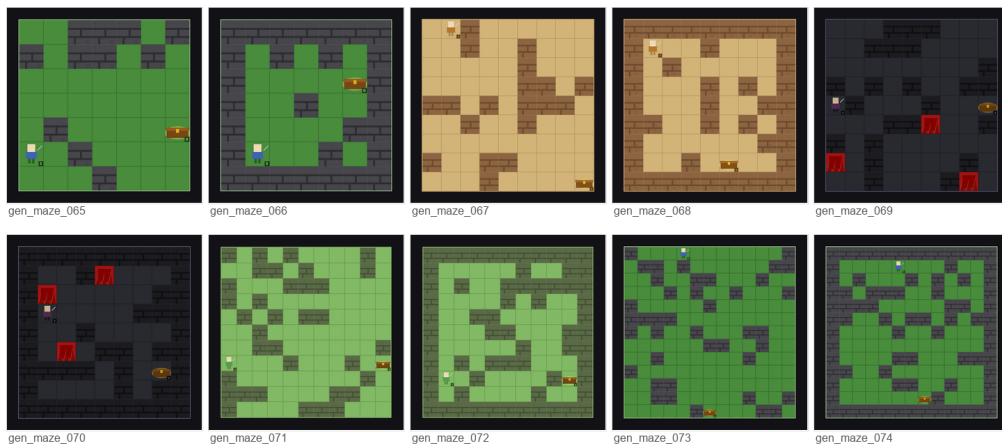


Figure 7: Group F: Border Walls (10 mazes). Five matched pairs with and without a wall border ring.

### G: Combined Hard

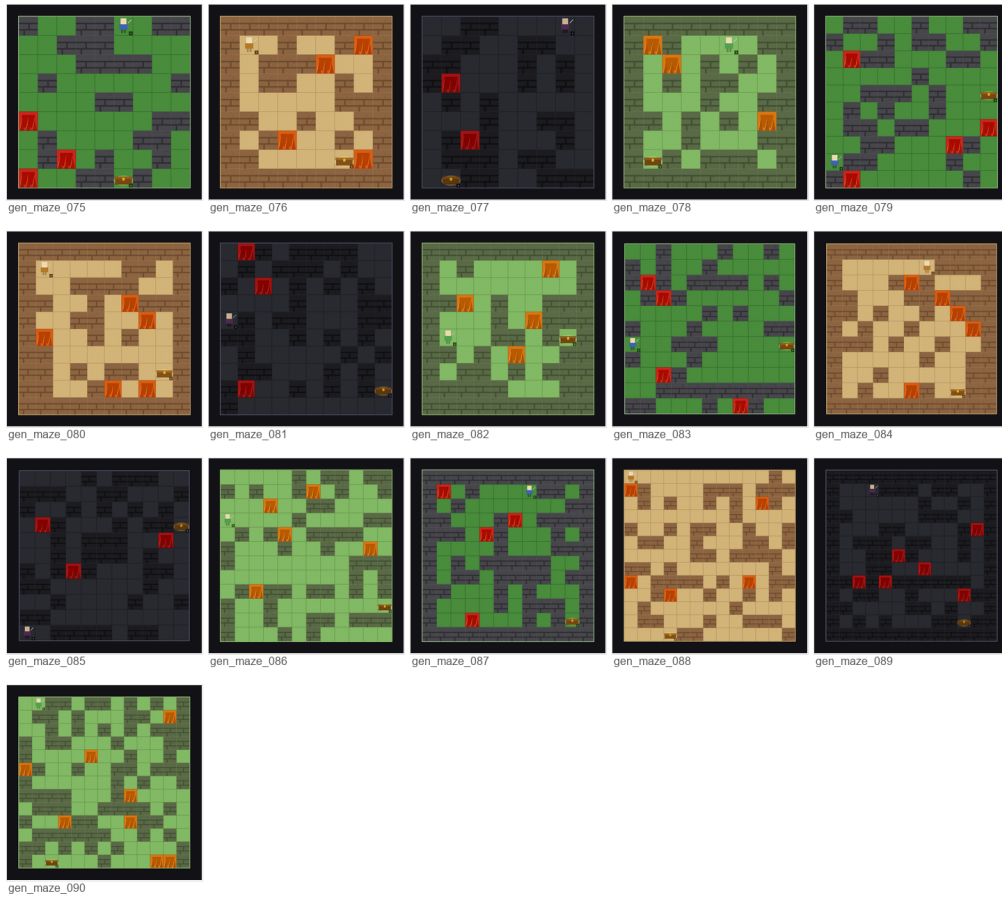


Figure 8: Group G: Combined Hard (16 mazes). Large grids with high wall density, traps, and borders.

### H: Palette Stress

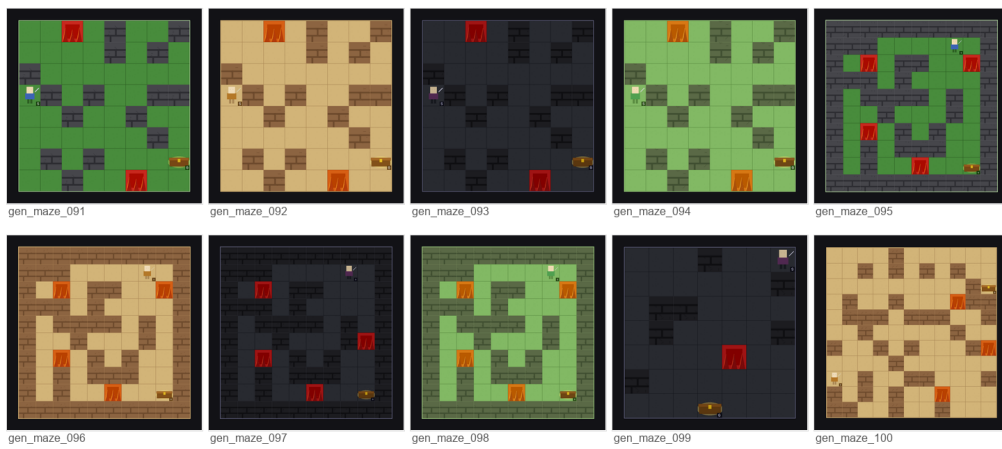


Figure 9: Group H: Palette Stress (10 mazes). Same maze structure rendered across four visual palettes.

**X: Ultra-Hard**

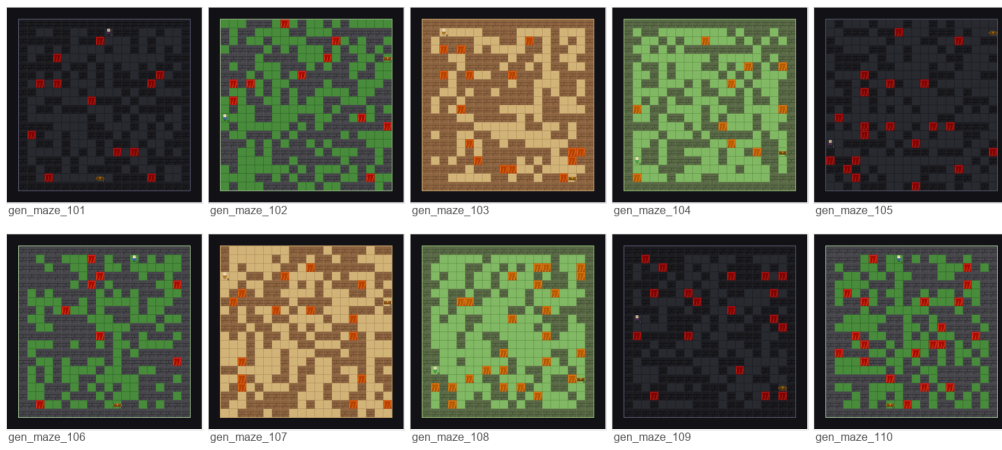


Figure 10: Group X: Ultra-Hard (10 mazes).  $20 \times 20$  grids with 8–25 traps and shortest paths of 28–42 steps.

# When Relations Break: Analyzing Relation Hallucination in Vision-Language Model Under Rotation and Noise

Philip Wootaeek Shin, Ajay Narayanan Sridhar, Sivani Devarapalli,  
Rui Zhang, Jack Sampson, Vijaykrishnan Narayanan

The Pennsylvania State University

{pws5345, afs6372, lfd5379, rmz5227, jms1257, vxn9}@psu.edu

## Abstract

Vision–language models (VLMs) achieve strong multimodal performance but remain prone to relation hallucination, which requires accurate reasoning over inter-object interactions. We study the impact of visual perturbations, specifically rotation and noise, and show that even mild distortions significantly degrade relational reasoning across models and datasets. We further evaluate prompt-based augmentation and preprocessing strategies (orientation correction and denoising), finding that while they offer partial improvements, they do not fully resolve hallucinations. Our results reveal a gap between perceptual robustness and relational understanding, highlighting the need for more robust, geometry-aware VLMs.

## 1 Introduction

Vision–language models (VLMs) have been widely deployed across a range of applications, including object recognition (Feng et al., 2025; Jin et al., 2024), scene understanding (Liao et al., 2024; Selvam et al., 2025), and multimodal reasoning (Jiang et al., 2025; Xu et al., 2025). Despite their impressive capabilities, these models are known to exhibit hallucinations, where the generated outputs are inconsistent with the visual input. Such hallucinations typically manifest in three forms: object hallucination, attribute hallucination, and relation hallucination (Bai et al., 2025). Among these, relation hallucination remains particularly challenging, as it requires accurately capturing interactions and spatial relationships between objects rather than simply identifying their presence or attributes (Zheng et al., 2025).

In this work, we observe that relation hallucination is highly sensitive to visual perturbations, such as image noise and rotation, as illustrated in Fig. 1. While often studied independently, rotation and noise represent complementary

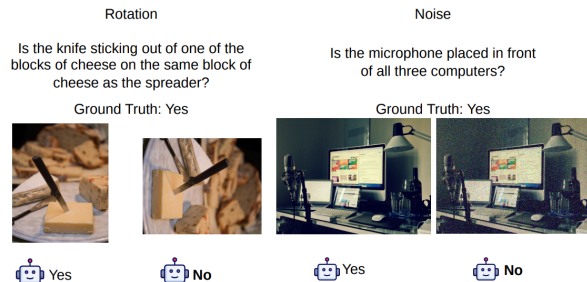


Figure 1: VLM response under visual perturbations.

failure modes—rotation disrupts geometric invariance, while noise degrades visual fidelity—and frequently co-occur in real-world settings, jointly affecting both structure and quality. While one might argue that rotating an image can alter perceived spatial relationships, humans naturally compensate for such transformations by mentally correcting orientation and still infer consistent object relations. In contrast, VLMs often fail to exhibit this invariance, leading to significantly degraded relational reasoning under such conditions.

To systematically investigate this phenomenon, we make the following contributions:

- **Comprehensive Analysis:** We conduct a systematic study of relation hallucination under varying noise and rotation—capturing both geometric and photometric perturbations—across multiple VLMs and datasets, providing a unified evaluation of robustness.
- **Prompt-Based Intervention:** We analyze how incorporating auxiliary prompts (e.g., rotation metadata or noise descriptions) influences model outcome and hallucination behavior.
- **Mitigation Strategy:** We propose a practical mitigation approach by introducing preprocessing modules—such as rotation correction and denoising—prior to VLM inference,

demonstrating improved robustness in relational reasoning.

## 2 Related Work

Hallucination in vision–language models can be categorized into object, attribute, and relation hallucination, with the latter being the most challenging due to its reliance on inter-object dependencies and spatial reasoning (Bai et al., 2025). Existing benchmarks study relation hallucination using yes/no and multiple-choice formats (Nie et al., 2025; Wu et al., 2024; Zheng et al., 2025), but largely focus on clean visual inputs.

Recent work has explored VLM robustness under visual perturbations, showing that noise and rotation can significantly degrade performance (Zhang et al., 2025; Shin et al., 2025; Niu et al., 2026). However, these studies consider perturbations in isolation and do not directly address their impact on relation hallucination.

While preprocessing methods such as orientation detection and denoising (Barbosa, 2025; Chen et al., 2025; Yu et al., 2024) improve visual quality, their effectiveness in mitigating hallucination remains unclear. In contrast, our work systematically studies the combined effects of noise and rotation on relation hallucination and evaluates both prompt-based and preprocessing-based mitigation strategies, bridging robustness analysis with hallucination-specific evaluation.

## 3 Rotation Analysis

To evaluate our hypothesis on the effect of rotation on relation hallucination, we utilize three benchmark datasets: MMRel (Nie et al., 2025), R-Bench (Wu et al., 2024), and Reefknot (Zheng et al., 2025). Given that each dataset exhibits distinct characteristics and potential overlaps in image sources and annotations, we carefully curate subsets to ensure a fair and non-redundant evaluation.

Specifically, we remove duplicate image–question pairs and avoid overlapping samples across datasets, while maintaining diversity in relational queries. Through this selective curation process, we construct evaluation sets comprising 1,632 image–question pairs for MMRel, 3,466 pairs for R-Bench, and Reefknot subsets with 1,185 multiple-choice questions and 2,922 binary (yes/no) questions. This rigorous selection enables a consistent and unbiased analysis of relation hallucination under rotational

Model	No Rotation (Orig)	90°	270°
Qwen2-VL 7B	78.02	72.07	72.65
InternVL2 8B	79.80	75.36	75.62
LLaVA-Next 8B	80.29	78.85	78.56
DeepSeek-Janus 7B	64.66	64.66	64.66
LLaMA-3.2 11B (Vision)	81.16	77.29	77.44

Table 1: Performance (%) of open-source vision–language models on R-Bench under different rotation settings.

perturbations.

### 3.1 Experimental Setup for Rotation-Induced Relation Hallucination

To investigate the effect of image rotation on relation hallucination, we apply controlled clockwise (90°) and counterclockwise (270°) rotations to the input images. We evaluate both rotation directions using five open-source vision–language models—Qwen2-VL 7B (Wang et al., 2024), InternVL2 8B (Chen et al., 2024), LLaVA-Next 8B (Liu et al., 2024), DeepSeek-Janus 7B (DeepSeek-AI, 2024), and LLaMA-3.2 11B (Vision) (Meta AI, 2024)—on the R-Bench dataset which we select as a representative benchmark due to its balanced coverage of relational reasoning tasks and manageable scale for controlled experimentation across multiple models.

As shown in Tab. 1, we observe no significant performance difference between 90 degrees clockwise and counterclockwise rotations. Therefore, for clarity and consistency, we report results using the clockwise rotation setting throughout the paper. We exclude 180° rotations, as such transformations are less representative of real-world viewing conditions and largely preserve global object configurations without introducing meaningful ambiguity in relational reasoning.

For evaluation, we employ three widely used closed-source vision–language models: GPT-5.1 (OpenAI, 2026), Gemini 2.5 Pro (DeepMind, 2026), and Claude Sonnet 4.5 (Anthropic, 2026). These models are selected due to their strong multimodal reasoning capabilities and broad adoption in recent studies.

During inference, each model is provided with the rotated image and the original question prompt, without any additional guidance or metadata. The generated responses are then compared against the ground-truth answers provided in each dataset. Quantitative results and comparative analyses are summarized in Tab. 2. Overall, we observe a con-

Category	GPT-5.1	Gemini 2.5 Pro	Claude Sonnet 4.5
Reefknot Y/N (orig)	76.61	75.41	72.25
Reefknot Y/N (cw)	73.03	71.73	65.20
Reefknot MCQ (orig)	88.51	86.08	80.08
Reefknot MCQ (cw)	82.61	80.92	73.33
R-Bench Y/N (orig)	80.15	75.97	79.69
R-Bench Y/N (cw)	78.51	48.93	72.74
MMRel Y/N (orig)	89.83	90.87	64.03
MMRel Y/N (cw)	58.82	53.25	30.45

Table 2: Accuracy (%) of vision-language models under original (orig) and clockwise rotated (cw) settings across Reefknot, R-Bench, and MMRel datasets.

sistent performance degradation under clockwise rotation across all models and datasets, confirming the sensitivity of VLMs to geometric perturbations in relational reasoning task. We focus on closed-source VLMs as they represent state-of-the-art, widely deployed systems and must be evaluated under realistic black-box conditions, ensuring our findings generalize to real-world applications.

### 3.2 Mitigating Rotation-Induced Relation Hallucination: Prompting vs. Preprocessing

From Tab. 1 and Tab. 2, we observe that image rotation consistently increases relation hallucination, motivating the need for effective mitigation strategies. To address this issue, we explore two approaches: (1) prompt-based augmentation using rotation metadata and (2) preprocessing via image orientation correction.

First, we investigate whether incorporating auxiliary prompts—specifically rotation information—can improve model robustness. Our hypothesis is that providing explicit orientation metadata may help VLMs compensate for geometric transformations and thus reduce hallucination. We conduct this analysis using GPT-5.1, which demonstrated relatively strong robustness on the Reefknot dataset under rotation. The Reefknot dataset contains both perceptual questions (orientation-related) and cognitive questions (action/interaction-related). However, as shown in Fig. 2, incorporating rotation metadata as an additional prompt yields minimal improvement in accuracy, suggesting that prompt-based guidance alone is insufficient to mitigate relation hallucination.

In contrast, we explore a preprocessing-based approach using a lightweight image orientation detector (Barbosa, 2025). We evaluate this detector on a modified Reefknot dataset with randomly applied  $90^\circ$  and  $270^\circ$  rotations. The detec-

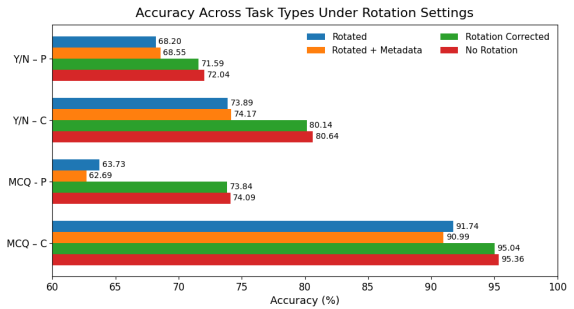


Figure 2: Effect of rotation metadata on VLM accuracy. (Reefknot, GPT-5.1)

tor achieves 99.66% accuracy on multiple-choice questions (MCQ) with a downstream accuracy of 95.04% (Cognitive) and 73.84% (Perceptual) for GPT-5.1 and 99.38% accuracy on yes/no (Y/N) questions with a downstream accuracy of 80.14% (Cognitive) and 71.59% (Perceptual) for GPT-5.1. These results indicate that correcting image orientation prior to VLM inference is a highly effective strategy, and is more reliable than relying on additional prompting to convey rotation information.

## 4 Noise Analysis

To evaluate noise effects on relation hallucination, we adopt 19 corruption types from (Zhang et al., 2025; Hendrycks and Dieterich, 2019) across Reefknot, R-Bench, and MMRel, grouped into six categories (A–F). We exclude weather (E) and geometric/structural (F) corruptions, as they introduce unrealistic artifacts and alter scene geometry, thereby invalidating ground-truth object relationships.

From the remaining geometry-preserving categories (A–D: noise-based, blur/distortion, compression/resolution, and photometric/color corruptions), we select one representative corruption each—Impulse Noise, Gaussian Blur, Pixelate, and Saturate—to enable a controlled and interpretable evaluation while preserving spatial relationships critical for assessing relation hallucination.

### 4.1 Impact of Noise Corruptions and Severity on Relation Hallucination

To examine the effect of noise on relation hallucination, we apply four representative corruptions—Impulse Noise, Gaussian Blur, Pixelate, and Saturate—at severity level 2. We hypothesize that introducing such perturbations would increase hallucination by degrading visual fidelity and disrupting relational cues. We evaluate this using the Reefknot dataset with GPT-5.1. As shown in Tab. 3, we

Category	Baseline	Gaussian Blur	Impulse Noise	Pixelate	Saturate
MC – Perceptual	74.09%	62.18%	64.77%	63.47%	63.73%
YN – Perceptual	72.04%	69.69%	69.36%	69.88%	70.22%
MC – Cognitive	95.36%	91.99%	91.86%	92.61%	92.49%
YN – Cognitive	80.64%	75.05%	76.14%	76.92%	77.19%

Table 3: Impact of noise corruptions (severity level 2) on relation hallucination performance using GPT-5.1 on the Reefknot dataset. Results are reported as correct predictions (accuracy %). All corruption types lead to performance degradation compared to the baseline.

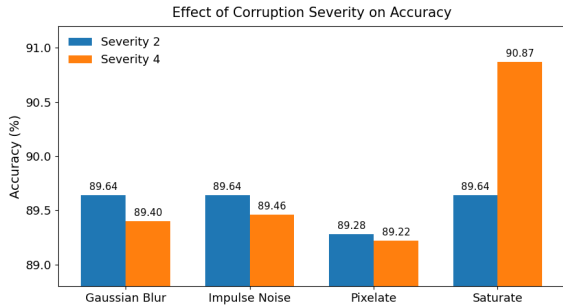


Figure 3: Effect of corruption severity on relation hallucination performance. (MMRel dataset, GPT-5.1)

observe a consistent decrease in accuracy across both yes/no and multiple-choice settings, as well as across perceptual and cognitive tasks, confirming the negative impact of noise on relational reasoning.

Furthermore, we analyze the effect of increasing corruption severity using GPT-5.1 on the MMRel dataset. As reported in Fig. 3, performance generally degrades as severity increases. However, *saturate* occasionally improves accuracy, likely because enhanced color contrast strengthens salient cues, making object relationships easier for VLMs to detect.

#### 4.2 Mitigating Noise-Induced Relation Hallucination: Prompting vs. Denoising

From Tab. 3 and Fig. 3, we observe that relation hallucination consistently worsens as image noise increases, confirming the sensitivity of VLMs to visual corruptions. Motivated by this, we explore two mitigation strategies: (1) prompt-based augmentation using noise information and (2) preprocessing via image denoising prior to model inference.

From Tab. 4, we compare the effectiveness of prompt-based augmentation and preprocessing-based denoising for mitigating noise-induced relation hallucination. Overall, both approaches provide limited but non-negligible improvements, with effectiveness varying across datasets and corruption types.

Dataset	Corruption	Acc (%)	Drop (pp)	Acc-D (%)	Drop-D (pp)
MMRel	Gaussian Blur	86.70	3.12	89.34	0.49
	Impulse Noise	87.32	2.51	88.97	0.86
	Pixelate	86.58	3.25	89.34	0.49
	Saturate	85.54	4.29	87.68	2.15
R-Bench	Gaussian Blur	80.84	8.99	80.06	9.76
	Impulse Noise	80.35	9.48	80.18	9.65
	Pixelate	80.24	9.59	80.18	9.65
	Saturate	78.04	11.78	77.44	12.39

Table 4: Comparison of noise corruption (S2) and preprocessing-based denoising on relation hallucination across MMRel and R-Bench using GPT-5.1. Acc/Drop denote results under corruption, while Acc-D/Drop-D denote results after denoising.

On R-Bench, performance drops remain large (9–12 pp) under both settings, indicating that denoising alone cannot fully recover relational reasoning under severe noise. In contrast, on MMRel, denoising yields smaller drops (0.5–4 pp) and generally outperforms prompt-based augmentation, suggesting benefits when underlying visual structure is preserved.

For denoising, we evaluate state-of-the-art restoration models (Chen et al., 2025; Yu et al., 2024) and select outputs using LPIPS (Zhang et al., 2018), PSNR, and SSIM (Wang et al., 2004). Despite improved perceptual quality, gains in relational reasoning are inconsistent, indicating a gap between low-level restoration and high-level understanding.

Overall, denoising is a useful but condition-dependent strategy: it helps when noise affects low-level features, but is less effective when relational cues are disrupted or task complexity is high

## 5 Conclusion

In this work, we study relation hallucination in VLMs under rotation and noise, showing that even simple perturbations significantly degrade relational reasoning, with larger effects on complex datasets. While prompting and preprocessing (orientation correction and denoising) provide partial improvements, they fail to fully recover performance, and gains in perceptual quality do not consistently translate to better reasoning. These results reveal a gap between perceptual robustness and relational understanding, motivating more robust, geometry-aware VLM designs.

## Acknowledgments

This work was supported in part by NSF Awards 2243979 and 2318101

## References

- Anthropic. 2026. Claude sonnet 4.5. <https://claude.ai>. Large language model.
- Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2025. [Hallucination of multimodal large language models: A survey](#).
- Duarte Barbosa. 2025. Deep image orientation detection. <https://github.com/duartebarbosadev/deep-image-orientation-detection>. GitHub repository.
- I Chen, Wei-Ting Chen, Yu-Wei Liu, Yuan-Chun Chiang, Sy-Yen Kuo, Ming-Hsuan Yang, and 1 others. 2025. Unirestore: Unified perceptual and task-oriented image restoration model using diffusion prior. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17969–17979.
- Zhe Chen, Wenhai Wang, Yue Cao, and 1 others. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*.
- Google DeepMind. 2026. Gemini 2.5 pro. <https://deepmind.google>. Large language model.
- DeepSeek-AI. 2024. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*.
- Yongchao Feng, Yajie Liu, Shuai Yang, Wenrui Cai, Jinqing Zhang, Qiqi Zhan, Ziyue Huang, Hongxi Yan, Qiao Wan, Chenguang Liu, and 1 others. 2025. Vision-language model for object detection and segmentation: A review and evaluation. *arXiv preprint arXiv:2504.09480*.
- Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.
- Chaoya Jiang, Yongrui Heng, Wei Ye, Han Yang, Haiyang Xu, Ming Yan, Ji Zhang, Fei Huang, and Shikun Zhang. 2025. Vlm-r3: Region recognition, reasoning, and refinement for enhanced multimodal chain-of-thought. *arXiv preprint arXiv:2505.16192*.
- Sheng Jin, Xueying Jiang, Jiaying Huang, Lewei Lu, and Shijian Lu. 2024. LLMs meet vlms: Boost open vocabulary object detection with fine-grained descriptors. *arXiv preprint arXiv:2402.04630*.
- Guibiao Liao, Jiankun Li, and Xiaoqing Ye. 2024. Vlm2scene: Self-supervised image-text-lidar learning with foundation models for autonomous driving scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3351–3359.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.
- Meta AI. 2024. Llama 3.2: Open multimodal foundation models. *arXiv preprint arXiv:2409.17146*.
- Jiahao Nie, Gongjie Zhang, Wenbin An, Yun Xing, Yap-Peng Tan, Alex C. Kot, and Shijian Lu. 2025. [Mmrel: Benchmarking relation understanding in multi-modal large language models](#). *Preprint*, arXiv:2406.09121.
- Tianyi Niu, Jaemin Cho, Elias Stengel-Eskin, and Mohit Bansal. 2026. [Rotbench: Evaluating multimodal large language models on identifying image rotation](#). *Preprint*, arXiv:2508.13968.
- OpenAI. 2026. Chatgpt-5.1. <https://chat.openai.com>. Large language model.
- Surya Selvam, Ravi K Rajendran, Murugan Sankaradas, Anand Raghunathan, and Srimat T Chakradhar. 2025. Simcache: similarity caching for efficient vlm-based scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3327–3336.
- Philip Wootae Shin, Jack Sampson, Vijaykrishnan Narayanan, Andres Marquez, and Mahantesh Halappanavar. 2025. [Losing the plot: How vlm responses degrade on imperfect charts](#). *Preprint*, arXiv:2509.18425.
- Peng Wang, Shuai Bai, Shengbang Tan, Shuai Wang, Zhihao Fan, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.
- Mingrui Wu, Jiayi Ji, Oucheng Huang, Jiale Li, Yuhang Wu, Xiaoshuai Sun, and Rongrong Ji. 2024. [Evaluating and analyzing relationship hallucinations in large vision-language models](#). *Preprint*, arXiv:2406.16449.
- Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, and 1 others. 2025. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. *arXiv preprint arXiv:2504.15279*.
- Fanghua Yu, Jinjin Gu, Zheyuan Li, Jinfan Hu, Xiangtao Kong, Xintao Wang, Jingwen He, Yu Qiao, and Chao Dong. 2024. [Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild](#). *Preprint*, arXiv:2401.13627.
- Hao Zhang, Wenqi Shao, Hong Liu, Yongqiang Ma, Ping Luo, Yu Qiao, Nanning Zheng, and Kaipeng Zhang. 2025. [B-avibench: Toward evaluating the robustness of large vision-language model on black-box adversarial visual-instructions](#). *Trans. Info. For. Sec.*, 20:1434–1446.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595.

Kening Zheng, Junkai Chen, Yibo Yan, Xin Zou, and Xuming Hu. 2025. Reefknot: A comprehensive benchmark for relation hallucination evaluation, analysis and mitigation in multimodal large language models. *Preprint*, arXiv:2408.09429.

# VLCE: An External Knowledge Framework for Contextual Image Captioning in Disaster Assessment

Md. Mahfuzur Rahman<sup>1</sup>, Marufa Kamal<sup>2</sup>, Fahad Rahman<sup>3</sup>, Sunzida Siddique<sup>4</sup>,  
Ahmed Rafi Hasan<sup>3</sup>, Mohd Ariful Haque<sup>1</sup>, Kishor Datta Gupta<sup>1</sup>, Roy George<sup>1</sup>

<sup>1</sup>Clark Atlanta University, USA,

<sup>2</sup>BRAC University, Bangladesh,

<sup>3</sup>United International University Bangladesh,

<sup>4</sup>Daffodil International University, Bangladesh,

Correspondence: [mdmahfuzur.rahman@students.cau.edu](mailto:mdmahfuzur.rahman@students.cau.edu)

## Abstract

General-purpose vision-language models (VLMs) such as LLaVA and QwenVL produce descriptions of disaster imagery that lack domain-specific vocabulary and actionable detail. We propose the Vision-Language Caption Enhancer (VLCE), a framework that integrates external semantic knowledge from ConceptNet and WordNet into the caption generation process for post-disaster satellite and UAV imagery. VLCE operates in two stages: first, a baseline VLM generates an initial caption conditioned on YOLOv8 object detections; second, a knowledge-enriched sequential model, a CNN-LSTM or a hierarchical cross-modal Transformer, refines the caption using a vocabulary augmented with 1,566 domain-relevant terms extracted from knowledge graphs. We evaluate VLCE on two disaster benchmarks: xBD (satellite, 6,369 images, 3 damage classes) and RescueNet (UAV, 4,494 images, 12 damage classes), using CLIPScore for semantic alignment and InfoMetIC for informativeness. On RescueNet with the Transformer decoder, VLCE with knowledge graph enrichment produces captions preferred over QwenVL baselines in 95.33% of image pairs on InfoMetIC and 73.64% on CLIPScore. Qualitative analysis shows that without knowledge graph integration, generated captions exhibit hallucinations, word repetition, and semantic incoherence, whereas knowledge-enriched captions maintain factual consistency and domain-appropriate vocabulary. intended as a continuous, extensible monitor of differential framing under changing real-world inputs.

## 1 Introduction

Natural disasters cause widespread destruction to infrastructure, ecosystems, and communities. Rapid and accurate assessment of post-disaster damage is critical for coordinating emergency response, allocating resources, and planning recovery efforts. Satellite and unmanned aerial vehicle



Figure 1: Post-hurricane disaster imagery from the RescueNet dataset.

(UAV) images are now the main sources of data for disaster assessment. They give clear pictures of large areas of land.

Recent advances in vision-language models (VLMs) have enabled the automatic generation of textual descriptions from images. Models such as LLaVA (Liu et al., 2023) and QwenVL (Bai et al., 2023) achieve strong performance on general image captioning benchmarks. However, when applied to post-disaster imagery, these models produce generic descriptions, lack domain-specific terminology, and fail to capture the nuanced damage indicators that responders require. A caption stating “buildings in a residential area” provides little actionable information compared to one identifying “severely damaged roof structures with debris fields and evidence of flooding in surrounding roadways.”

This gap arises because general-purpose VLMs are trained mainly on everyday imagery and lack exposure to the specialized vocabulary and visual patterns characteristic of disaster scenarios (Mason and Charniak, 2014). The value of contextual captioning is illustrated in Figure 1. A model without domain knowledge produces: “A satellite image depicts a community from an aerial perspective, revealing multiple dwellings and streets” In contrast, a knowledge enriched model generates: “The

*image shows the aftermath of Hurricane Michael, which inflicted major damage to infrastructure and the environment. Roads and streets are cluttered with debris from destroyed structures, such as fallen trees and scattered items. Trees in the region have suffered varied degrees of damage, with some seeming damaged or totally fallen. Buildings show different degrees of destruction, ranging from comparatively intact ones to those with obvious traces of catastrophic damage. The entire scene depicts the ongoing recovery and rebuilding activities in the aftermath of the catastrophe*”, defining the type of disaster, outlining the effects on infrastructure, and communicating the state of recovery. Adding domain-relevant outside knowledge to the caption generation pipeline is necessary to overcome this constraint.

### 1.1 Our Contributions

We propose VLCE (Vision-Language Caption Enhancer), a framework that enriches disaster image captioning through external knowledge graph integration. Our contributions are as follows:

#### 1. Knowledge-enriched captioning framework.

We introduce a two-stage pipeline that combines baseline VLM captions with knowledge graph-augmented sequential models. The system extracts keywords from training captions using RAKE, looks up related terms in ConceptNet and WordNet, and builds a vocabulary of 3,195 words (1,566 from knowledge graphs) that helps create more accurate descriptions of disasters.

#### 2. Dual architecture for satellite and UAV imagery.

We create two different decoder designs: one is a CNN-LSTM model that combines visual and text information, and the other is a hierarchical cross-modal Transformer that processes visual information. Each architecture is paired with a domain-specific image encoder: ResNet50-EuroSAT for satellite imagery and a ViT-based UAV classifier for drone imagery.

#### 3. Comprehensive evaluation on disaster benchmarks.

We evaluate xBD and RescueNet using CLIPScore and InfoMetIC, which are two complementary metrics that together capture both semantic alignment and informational content. We show that knowledge graph integration is essential, and without it, the quality of captions for UAV images drops significantly (less than 2% preference on both metrics for RescueNet),

but with it, the Transformer model gets up to 95.33% preference compared to standard VLMs on InfoMetIC.

## 2 Related Work

Satellite-based damage assessment has progressed from manual interpretation to automated deep learning pipelines. The xBD dataset (Gupta et al., 2019) established a large-scale benchmark for building damage classification from satellite imagery, while RescueNet (Rahnemoonfar et al., 2023) extended coverage to UAV-captured scenes with fine-grained damage categories. Alisjahbana et al. (Alisjahbana et al., 2024) proposed a two-step CNN for building damage segmentation (0.66 F1). Abbas and Dang (Abbas and Dang, 2023) investigated CNN-based caption generation for disaster areas (BLEU-1: 0.8731, CIDEr: 5.0908), and Chun et al. (Chun et al., 2022) applied deep learning for bridge damage descriptions (92.9% accuracy). Multimodal approaches combining image and text data have shown promise for disaster classification (Zou et al., 2021; Kota et al., 2022). Most existing methods focus on damage *classification* rather than free-form description generation.

LLaVA (Liu et al., 2023) pairs a CLIP ViT-L/14 encoder with a LLaMA decoder through visual instruction tuning. QwenVL (Bai et al., 2023) extends the Qwen language model with a vision encoder for multi-image understanding. Cornia et al. (Cornia et al., 2020) explored meshed-memory Transformers for captioning (CIDEr 132.7 on MS COCO). Chen and Li (Chen and Li, 2025) introduced multi-modal graph aggregation Transformers. While these models excel on general benchmarks, their application to specialized domains such as disaster assessment remains limited by training data distribution (Mason and Charniak, 2014).

External knowledge bases have been used to enhance language understanding across a range of tasks. ConceptNet (Schön et al., 2019) provides commonsense relational knowledge, while WordNet offers lexical relationships including synonymy and hypernymy. ConceptNet Numberbatch embeddings combine distributional semantics with structured knowledge, producing word vectors that encode both corpus statistics and relational information. Zhou et al. (Zhou et al., 2019) applied knowledge graphs to captioning on MS COCO and Visual Genome. Zhao and Wu (Zhao and Wu, 2024) proposed multi-modal KG methods for graph attention

captioning. Tang et al. (Tang et al., 2022) introduced KG-guided attention for CNN-LSTM models. Wang et al. (Wang et al., 2024) explored KG strategies for remote sensing captioning. Despite these advances, KG application to disaster-specific imagery has not been explored.

Pre-trained word embeddings serve as the bridge between discrete vocabulary tokens and continuous neural network inputs. Comparative analyses (Toshevska et al., 2020; Elbedwehy et al., 2023) have evaluated embeddings for captioning. CLIP-Score (Hessel et al., 2021) provides reference-free semantic alignment, while InfoMetIC (Hu et al., 2023) assesses informativeness. YOLOv8 (Talib et al., 2024) has been adopted for object detection in UAV (Zhai et al., 2023; Wang et al., 2023) and general scenarios (Rasheed and Zarkoosh, 2025).

### 3 Methodology

Figure 2 illustrates the VLCE pipeline. The framework works in two steps: 1) first, it creates basic captions using a ready-made VLM, and 2) it improves those captions with extra knowledge using a trained sequential model.

#### 3.1 Baseline Caption Generation

We generate initial captions using two VLMs with complementary characteristics. YOLOv8 (Talib et al., 2024) first performs object detection on each input image  $I$ , producing an annotation set  $\mathcal{A} = \{(b_i, \ell_i)\}_{i=1}^K$  where the bounding box coordinates  $b_i \in \mathbb{R}^4$  and the class label  $\ell_i \in \mathcal{L}$ . These annotations are integrated into a textual prompt  $P(\mathcal{A})$  describing the scene to guide the VLM.

**LLaVA-7B v1.5.** We use the LLaVA-7B v1.5 model (Liu et al., 2023), made of a CLIP ViT-L/14 visual encoder with a LLaMA language decoder. The encoder produces patch embeddings  $V = \phi(I) = [v_1, \dots, v_N] \in \mathbb{R}^{N \times D}$ , concatenated with the prompt into sequence  $s$ . The decoder generates the caption autoregressively:  $\hat{y} = \arg \max_y P(y | s)$ . Generation uses  $T = 0.2$  with a maximum output length of 4,000 tokens.

**QwenVL-7B Instruct.** We use the QwenVL-7B Instruct model (Bai et al., 2023) with greedy decoding and a maximum output length of 128 tokens. The vision encoder creates  $V = \text{QwenVL}_{\text{vision}}(I) \in \mathbb{R}^{H \times W \times D}$ . An event type  $e$  and annotations are included in a structured prompt  $P(\mathcal{A}, e)$ :  $\hat{y} = \arg \max_y P(y | V, P(\mathcal{A}, e))$ .

These two baselines differ substantially in output characteristics. LLaVA produces longer, more detailed descriptions, while QwenVL generates concise captions.

#### 3.2 Knowledge Graph Vocabulary Enrichment

The vocabulary enrichment process constructs a domain-specific lexicon from the training captions augmented with external knowledge:

**Step 1: Keyword extraction.** We apply the RAKE algorithm to all training captions. With it we extract ranked keyword phrases that capture disaster-relevant concepts such as “debris field,” “structural damage,” and “emergency response”.

**Step 2: ConceptNet expansion.** For each extracted keyword, we query the ConceptNet API, retrieving up to 10 semantically related terms per keyword. Relation filtering omits synonyms to focus on conceptual connections (e.g., “hurricane”  $\rightarrow$  “wind,” “flooding,” “evacuation”).

**Step 3: WordNet synonym enrichment.** Each keyword is additionally queried against WordNet to retrieve synonyms, providing lexical variety. Overlapping substrings are removed to prevent redundancy.

**Step 4: Vocabulary construction.** We filter out invalid English words and remove duplicates from the union of the original caption words, ConceptNet terms, and WordNet synonyms. This gives us a final vocabulary of 3,195 tokens, of which 1,566 (49%) are terms that were not in the original training captions but were added to the knowledge graph. Boundary tokens (startseq, endseq) are added, and sequences are padded to max length 192.

#### 3.3 Embedding Strategies

We evaluate two embedding strategies that differ in whether they encode structured knowledge:

**With knowledge graph (KG).** ConceptNet Numberbatch embeddings are 300-dimensional word vectors. They are pre-trained to encode both distributional statistics and relational knowledge from ConceptNet. Each term  $t$  is mapped to:

$$\mathbf{e}_t = \begin{cases} \text{find}(D, t) & \text{if } t \in D \\ \mathbf{e}_{\text{rand}} \sim \mathcal{U}(-\epsilon, \epsilon) & \text{otherwise} \end{cases}$$

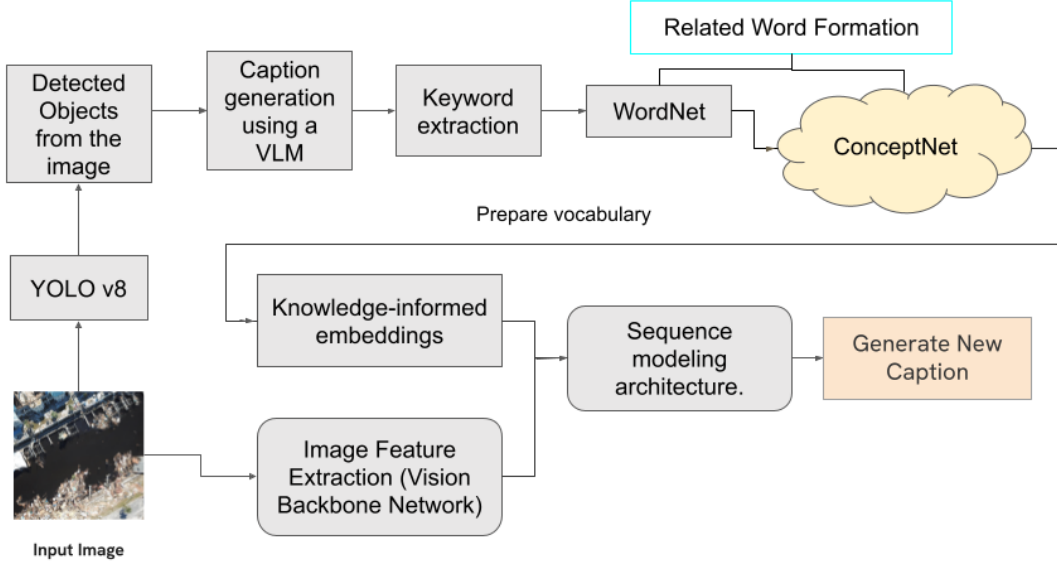


Figure 2: The VLCE pipeline. YOLOv8 generates scene-aware prompts for a baseline VLM (LLaVA or QwenVL) based on object detection in the input image. ConceptNet and WordNet queries are used to enhance keywords that were taken from training captions using RAKE, creating a vocabulary unique to the domain. The final caption is produced by training a sequential model with the enriched vocabulary and matching embeddings.

The embedding layer is initialized with these vectors. It remains frozen during training, preserving structured semantic relationships. The final embedding matrix is  $E \in \mathbb{R}^{|V| \times 300}$ .

**Without KG.** DistilBERT embeddings (768-dimensional) provide contextual representations learned from large-scale text corpora without explicit knowledge graph structure:  $e_w = \text{BERT}(w)_{[\text{CLS}]}$ . This serves as an ablation to isolate the contribution of structured knowledge.

### 3.4 Image Encoders

We select image encoders pre-trained on data distributions matching our target domains:

**ResNet50-EuroSAT.** For satellite imagery (xBD), we use a ResNet50 model pre-trained on the EuroSAT land-use classification dataset<sup>1</sup>, producing 2,048-dimensional feature vectors. Images are resized to  $336 \times 336$  pixels.

**ViT-UAV.** For UAV imagery (RescueNet), we use a Vision Transformer pre-trained on a UAV image classification task<sup>2</sup>, producing 768-dimensional

<sup>1</sup><https://huggingface.co/cm93/resnet50-eurosat>

<sup>2</sup><https://huggingface.co/SeyedAli/Remote-Sensing-UAV-image-classification>

feature vectors via global average pooling. Images are resized to  $224 \times 224$  pixels.

### 3.5 Sequential Decoders

We design two decoder architectures to investigate how different sequence modeling approaches interact with knowledge-enriched vocabularies.

#### 3.5.1 CNN-LSTM Decoder

The CNN-LSTM decoder (Figure 3) uses additive fusion of visual and textual features. Given an image feature vector  $\mathbf{v} \in \mathbb{R}^{d_v}$  (where  $d_v = 2048$  for ResNet50 or  $d_v = 768$  for ViT-UAV) and caption tokens  $w_1, \dots, w_T$ :

$$\begin{aligned}
 \mathbf{f}_{\text{img}} &= \text{ReLU}(\mathbf{W}_v \cdot \text{Dropout}_{0.5}(\mathbf{v})) \in \mathbb{R}^{256} \\
 \mathbf{f}_{\text{xt}} &= \text{LSTM}_{256}(\text{Dropout}_{0.5}(\mathbf{E}[w_1, \dots, w_T])) \in \mathbb{R}^{256} \\
 \mathbf{h} &= \text{ReLU}(\mathbf{W}_h \cdot (\mathbf{f}_{\text{img}} + \mathbf{f}_{\text{xt}})) \in \mathbb{R}^{256} \\
 P(w_{t+1} | w_{1:t}, \mathbf{v}) &= \text{softmax}(\mathbf{W}_o \cdot \mathbf{h})
 \end{aligned} \tag{1}$$

where  $\mathbf{E}$  is the embedding matrix (initialized from ConceptNet Numberbatch or DistilBERT and frozen), and  $\mathbf{W}_v$ ,  $\mathbf{W}_h$ ,  $\mathbf{W}_o$  are learned projection matrices.

#### 3.5.2 Hierarchical Cross-Modal Transformer Decoder

The Transformer decoder (Figure 4) introduces a hierarchical visual encoding scheme that captures

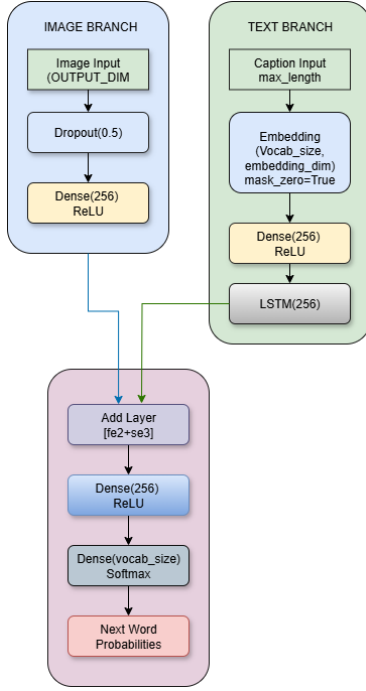


Figure 3: CNN-LSTM architecture with visual and textual branches.

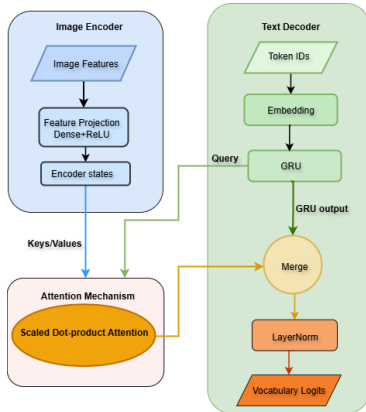


Figure 4: Hierarchical cross-modal Transformer architecture.

multi-scale spatial information, followed by cross-modal attention for visually grounded text generation.

**Multi-scale visual encoding.** The vector of image characteristics  $\mathbf{v} \in \mathbb{R}^{d_v}$  is projected in three complementary representations:

$$\begin{aligned} \mathbf{f}_{\text{global}} &= \text{ReLU}(\mathbf{W}_g \mathbf{v} + \mathbf{b}_g) \in \mathbb{R}^{d_{\text{model}}} \\ \mathbf{F}_{\text{regional}} &= \text{Dense}_{d_{\text{model}}} \left( \text{Reshape}_{4 \times d_{\text{model}}/4} (\text{ReLU}(\mathbf{W}_r \mathbf{v} + \mathbf{b}_r)) \right) \\ &\in \mathbb{R}^{4 \times d_{\text{model}}} \\ \mathbf{F}_{\text{local}} &= \text{Dense}_{d_{\text{model}}} \left( \text{Reshape}_{12 \times d_{\text{model}}/12} (\text{ReLU}(\mathbf{W}_l \mathbf{v} + \mathbf{b}_l)) \right) \\ &\in \mathbb{R}^{12 \times d_{\text{model}}} \end{aligned} \quad (2)$$

These are concatenated and normalized to form the visual context:

$$\mathbf{F}_v = \text{LayerNorm}([\mathbf{f}_{\text{global}}; \mathbf{F}_{\text{regional}}; \mathbf{F}_{\text{local}}]) \in \mathbb{R}^{17 \times d_{\text{model}}} \quad (3)$$

**Cross-modal decoding.** Input tokens are embedded with frozen pre-trained matrix  $\mathbf{E}$  and sinusoidal positional encodings:  $\mathbf{H}^{(0)} = \mathbf{E}[\mathbf{w}_{1:T}] + \text{PE}(\mathbf{w}_{1:T})$ . Each of the  $L$  decoder layers applies causal self-attention over caption tokens, followed by cross-attention over the visual context:

$$\begin{aligned} \mathbf{A}_{\text{self}} &= \text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{M}_{\text{causal}}) \\ \mathbf{A}_{\text{cross}} &= \text{MultiHead}(\mathbf{H}, \mathbf{F}_v, \mathbf{F}_v) \end{aligned} \quad (4)$$

where  $\mathbf{H}$  is the output of the self-attention sublayer and  $\mathbf{M}_{\text{causal}}$  enforces the autoregressive property.

**Output generation.** Global visual semantics are fused with textual representations:

$$\begin{aligned} \mathbf{c}_{\text{visual}} &= \text{GlobalAvgPool}(\mathbf{F}_v) \\ \hat{\mathbf{y}}_t &= \text{Dense}_V \left( \text{LayerNorm}([\mathbf{h}_t^{(L)}; \mathbf{c}_{\text{visual}}]) \right) \end{aligned} \quad (5)$$

### 3.6 Training

All models use  $d_{\text{model}} = d_{\text{emb}} = 300$ ,  $L = 2$  Transformer layers with  $h = 6$  attention heads, and are trained with masked sparse categorical cross-entropy loss:

$$\mathcal{L} = - \frac{1}{\sum_{t=1}^T m_t} \sum_{t=1}^T m_t \log P(w_t | \mathbf{w}_{<t}, \mathbf{F}_v; \theta) \quad (6)$$

where  $m_t \in \{0, 1\}$  masks padding positions. Batch size is 32, maximum sequence length 192 tokens. Captions are bounded by startseq and endseq tokens. At inference, generation proceeds autoregressively until the stop token or maximum length is reached. The complete pipeline is formalized in Algorithm 1.

## 4 Experimental Setup

### 4.1 Datasets

xBD (Gupta et al., 2019). A satellite imagery dataset for building damage assessment containing 12,738

---

**Algorithm 1** VLCE: Knowledge Graph-Enhanced Caption Refinement

---

**Require:** Disaster image dataset  $\mathcal{D} = \{I_1, \dots, I_n\}$ **Ensure:** Enhanced captions  $\mathcal{C}_{\text{Enhanced}}$ 

```
1: Stage 1: Baseline Caption Generation
2: for each image  $I_i \in \mathcal{D}$  do
3:    $O_i \leftarrow \text{YOLOv8}(I_i); \quad P_i \leftarrow$ 
   ConstructPrompt( $I_i, O_i$ )
4:    $C_i^{\text{VLM}} \leftarrow \text{VLM}(I_i, P_i)$ 
5: end for

6: Stage 2: Knowledge Preparation
7:  $\mathcal{V}_{\text{Enriched}} \leftarrow \text{Vocab}(\{C_i^{\text{VLM}}\}) \cup$ 
   RAKE+ConceptNet+WordNet
8:  $E \leftarrow \text{BuildEmbeddings}(\mathcal{V}_{\text{Enriched}})$ 

9: Stage 3: Training
10:  $\mathcal{F}_{\text{Img}} \leftarrow E^{\text{Visual}}(\mathcal{D}); \text{Train } \mathcal{N}_\theta \text{ on } (\mathcal{F}_{\text{Img}}, \mathcal{V}_{\text{Enriched}}, E)$ 

11: Stage 4: Enhanced Caption Generation
12: for each image  $I_i$  do
13:    $C_i^{\text{Enhanced}} \leftarrow \text{Decode}(I_i, \mathcal{N}_\theta)$ 
14: end for
15: return  $\mathcal{C}_{\text{Enhanced}}$ 
```

---

images. We use 6,369 post-disaster images, merging “major-damage” and “minor-damage” into a single “damaged” class to yield three labels: no-damage, damaged, and destroyed. Split: 80/20 (5,095 train / 1,274 test). RescueNet (Rahneemoufar et al., 2023). A UAV imagery dataset captured after Hurricane Michael, containing 4,494 images at  $3,000 \times 4,000$  pixel resolution across 12 fine-grained damage categories. Split: 80/20.

## 4.2 Evaluation Metrics

We evaluate using two complementary metrics: CLIPScore (Hessel et al., 2021), which computes the cosine similarity between CLIP embeddings (openai/clip-vit-base-patch32) of the image and generated caption. For each test image, we compare the CLIPScore of the VLCE-generated caption against the baseline VLM caption and note the percentage of images where VLCE scores higher. Another one is InfoMetIC (Hu et al., 2023), a composite metric:  $\alpha \cdot \text{Informativeness} + \beta \cdot \text{Relevance} + \gamma \cdot \text{Precision}$ , where Informativeness is computed as the average negative log-probability of caption words conditioned on the image using a vision-language model. InfoMetIC penalizes generic descriptions and rewards information-rich captions.

## 4.3 Configurations

We associate each decoder architecture with a designated baseline VLM: CNN-LSTM with LLaVA

and Transformer with QwenVL. Each pairing is assessed with and without knowledge graph enhancement across both datasets (RescueNet/ViT-UAV and xBD/ResNet50-EuroSAT).

## 5 Results

### 5.1 Main Results

Table 1 presents the percentage of test images where VLCE captions score higher than the corresponding baseline VLM caption. Values above 50% indicate VLCE outperforms the baseline for the majority of images.

### 5.2 Key Findings

**Knowledge graph integration is critical for UAV imagery.** On RescueNet, removal of knowledge graph (KG) enrichment causes near-complete performance collapse for both architectures (Figure 8). The CNN-LSTM without KG achieves only 0.56% CLIPScore preference and 1.22% InfoMetIC preference. The Transformer without KG drops to 0.22% and 0.08%, respectively. The 12-class damage taxonomy of RescueNet requires exact terminology that the model cannot learn from limited training captions alone.

**The Transformer with KG achieves the strongest results.** The best configuration is the Transformer decoder with ViT-UAV encoder and KG enrichment on RescueNet, achieving 73.64% CLIPScore and 95.33% InfoMetIC preference over QwenVL (Figure 5, Figure 6). The gap between metrics indicates VLCE captions are substantially more *informative* than the QwenVL baseline, even in cases where visual alignment scores are competitive.

**Satellite imagery is more forgiving.** On xBD, CNN-LSTM without KG still achieves 55.34% CLIPScore and 66.41% InfoMetIC preference. The lower label complexity (3 vs. 12 classes) and the ResNet50-EuroSAT encoder’s satellite-specific pre-training partially compensate for the absence of knowledge-enriched vocabulary.

**Architecture-specific patterns.** The CNN-LSTM shows binary behavior on RescueNet: it works with KG or fails without. The Transformer exhibits more graceful degradation on xBD without KG. It achieves a 22.14% CLIPScore, suggesting cross-attention can partially compensate for missing vocabulary knowledge on simpler scenes.

Table 1: Percentage of test images where VLCE captions score higher than the baseline VLM caption. CNN-LSTM models are compared against LLaVA; Transformer models against QwenVL. Bold indicates VLCE outperforms the baseline on the majority of images.

Decoder	Baseline	Dataset	KG	CLIPScore (%)		InfoMetIC (%)	
				VLCE	Baseline	VLCE	Baseline
CNN-LSTM	LLaVA	RescueNet	Yes	<b>52.95</b>	47.05	<b>54.51</b>	45.49
		RescueNet	No	0.56	99.44	1.22	98.78
		xBD	Yes	<b>51.10</b>	48.90	<b>66.56</b>	33.44
		xBD	No	<b>55.34</b>	44.66	<b>66.41</b>	33.59
Transformer	QwenVL	RescueNet	Yes	<b>73.64</b>	26.36	<b>95.33</b>	4.67
		RescueNet	No	0.22	99.78	0.08	99.92
		xBD	Yes	<b>60.60</b>	39.40	<b>69.86</b>	30.14
		xBD	No	22.14	77.86	18.76	81.24

Table 2: Relevant object counts across model configurations.

Exp.	Configuration	VLCE	LLaVA	QwenVL	Best
<b>ViT-UAV / RescueNet</b>					
1	With Knowledge Graph	<b>272</b>	185	201	VLCE
2	Without Knowledge Graph	<b>272</b>	178	195	VLCE
<b>ResNet-EuroSAT / xBD</b>					
3	With Knowledge Graph	<b>640</b>	445	463	VLCE
4	Without Knowledge Graph	<b>640</b>	431	448	VLCE
<b>Summary</b>					
Avg. improvement over LLaVA		<b>+45.2%</b>	–	–	–
Avg. improvement over QwenVL		<b>+38.7%</b>	–	–	–
Experiments won		<b>4/4</b>	0/4	0/4	–

### 5.3 Quantitative Analysis

Figures 5–8 aggregate the quantitative findings. Two observations are noteworthy. The KG ablation (Figure 7) demonstrates that the Transformer architecture derives much bigger advantages from KG on RescueNet (+73.4 pp CLIPScore, +95.3 pp InfoMetIC) compared to xBD (+38.5 pp, +51.1 pp), indicating the heightened language requirements of UAV images. Secondly, the CNN-LSTM on xBD without knowledge graphs is the sole configuration in which the removal of knowledge graphs marginally enhances the CLIPScore by 4.2 percentage points, indicating that for satellite imagery with less complex damage categories, direct visual-linguistic mapping can be competitive. Appendix A.1 contains the distributions of per-image scores.

### 5.4 Noun-Based Object Detection Analysis

Table 2 reports the total count of unique relevant nouns detected across all test captions to assess whether VLCE captures more disaster-relevant objects.

VLCE achieves full coverage of the reference object set (272 UAV objects, 640 satellite objects) compared to 55–65% by baseline VLMs, confirming the knowledge-enriched vocabulary enables broader and more precise object references.<sup>2</sup>

## 6 Discussion

**Why knowledge graphs matter more for UAV imagery.** The significant performance gap between with-KG and without-KG configurations on RescueNet vs. xBD shows the main differences between the two imaging methods. UAV imagery captures scenes from oblique angles at high spatial resolution. It reveals fine-grained damage details such as collapsed walls, scattered debris, and standing water that require precise vocabulary to describe. Satellite imagery, observed from a direct overhead perspective at lower resolution, shows more uniform visual patterns where more abstract descriptions suffice. RescueNet’s 12-class damage taxonomy also necessitates specific terminology, which can only be provided by the knowledge-enriched vocabulary.

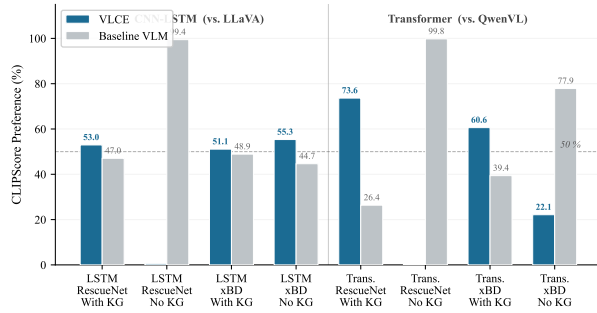


Figure 5: CLIPScore preference across all configurations. The dashed line marks 50% (parity). Without KG, VLCE preference collapses on RescueNet.

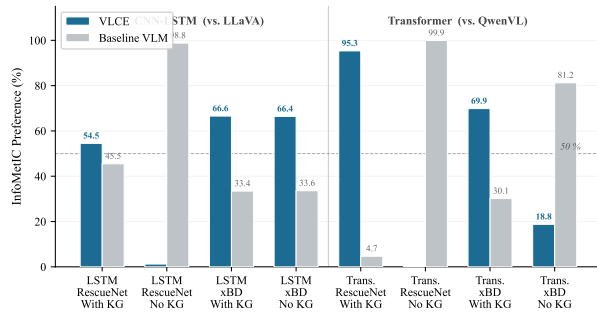


Figure 6: InfoMetIC preference across all configurations. The Transformer with KG on RescueNet achieves 95.33% preference over QwenVL.

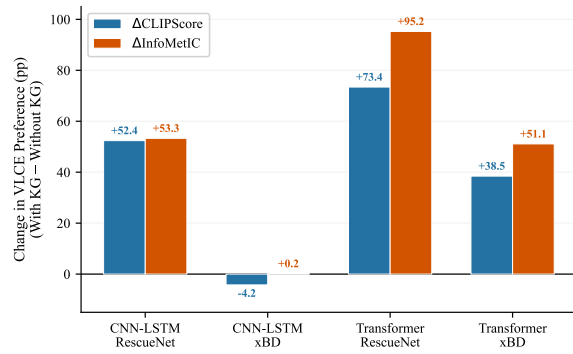


Figure 7: Effect of KG integration measured as the change in VLCE preference (pp). The Transformer on RescueNet gains +73.4 pp CLIPScore and +95.3 pp InfoMetIC.

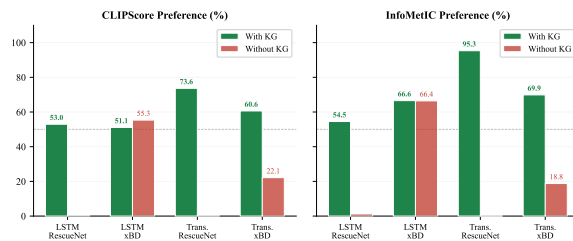


Figure 8: With vs. without KG. Green bars consistently exceed 50%, while red bars collapse to near-zero on RescueNet.

## Complementarity of CLIPScore and InfoMetIC.

Configurations with a high InfoMetIC but a moderate CLIPScore, such as the Transformer with KG on RescueNet (95.33% vs. 73.64%), show that using a knowledge graph helps the model include important details that aren't directly visible in the image. CLIPScore, which assesses image-text alignment, may underestimate descriptions that include domain knowledge regarding disaster processes, whereas InfoMetIC values such informational depth.

**Practical implications.** For operational disaster response, captions must go beyond visible elements. InfoMetIC-favored captions, which include domain-appropriate terminology and contextual reasoning, are more actionable, as demonstrated by the Transformer+KG configuration's strong performance. This suggests knowledge graphs bridge the gap between visual observation and situational understanding.

## 7 Conclusion

We introduced VLCE, a framework for generating knowledge-enriched captions for post-disaster imagery. By integrating external semantic knowledge from ConceptNet and WordNet into the caption generation pipeline, VLCE addresses the vocabulary and domain knowledge limitations of general-purpose VLMs when applied to disaster assessment. Our experiments on the xBD and RescueNet benchmarks demonstrate that knowledge graph enrichment is critical for UAV imagery captioning, where it transforms near-zero performance into strong preference over baseline VLMs. The Transformer decoder with knowledge graph integration achieves the best overall results, with 95.33% InfoMetIC preference on RescueNet. Qualitative analysis substantiates that knowledge-enhanced captions preserve factual accuracy and contextually relevant terminology, whereas captions produced without knowledge graph assistance display hallucinations, redundancy, and semantic disparity.

**Acknowledgement:** We used ChatGPT 5 to assist in refining the language, clarity, and formatting of the  $\text{\LaTeX}$  manuscript. All technical content, analysis, and conclusions are the authors' own.

## References

O. Abbas and J. Dang. 2023. Using image captioning for automatic post-disaster damage detection. *Intelli-*

- gence, Informatics and Infrastructure*, 4(2):66–74.
- I. Alisjahbana and 1 others. 2024. [DeepDamageNet: A two-step deep-learning model for multi-disaster building damage segmentation](#). *Preprint*, arXiv:2405.04800.
- J. Bai and 1 others. 2023. [Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond](#). *Preprint*, arXiv:2308.12966.
- L. Chen and K. Li. 2025. Multi-modal graph aggregation transformer for image captioning. *Neural Networks*, 181:106813.
- P.-J. Chun, T. Yamane, and Y. Maemura. 2022. A deep learning-based image captioning method for bridge damage. *Computer-Aided Civil and Infrastructure Engineering*, 37(11):1387–1401.
- Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587.
- S. Elbedwehy and 1 others. 2023. Enhanced image captioning using features concatenation and efficient pre-trained word embedding. *Computer Systems Science and Engineering*, 46:3637–3652.
- Ritwik Gupta and 1 others. 2019. [xBD: A dataset for assessing building damage from satellite imagery](#). *Preprint*, arXiv:1911.09296.
- J. Hessel and 1 others. 2021. [CLIPScore: A reference-free evaluation metric for image captioning](#). *Preprint*, arXiv:2104.08718.
- A. Hu and 1 others. 2023. [InfoMetIC: An informative metric for reference-free image caption evaluation](#). *Preprint*, arXiv:2305.06002.
- S. Kota and 1 others. 2022. Multimodal combination of text and image tweets for disaster response assessment. In *D2R2*.
- H. Liu and 1 others. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916.
- Rebecca Mason and Eugene Charniak. 2014. Domain-specific image captioning. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 28–36. Association for Computational Linguistics.
- M. Rahnemoonfar, T. Chowdhury, and R. Murphy. 2023. RescueNet image classification dataset. Figshare Dataset.
- A. F. Rasheed and M. Zarkoosh. 2025. Optimized YOLOv8 for multi-scale object detection. *Journal of Real-Time Image Processing*, 22(1):6.
- C. Schön, S. Siebert, and F. Stolzenburg. 2019. Using ConceptNet to teach common sense to an automated theorem prover. *Electronic Proceedings in Theoretical Computer Science*, 311:19–24.
- M. Talib, A. H. Y. Al-Noori, and J. Suad. 2024. YOLOv8-CAB: Improved YOLOv8 for real-time object detection. *Karbala International Journal of Modern Science*, 10(1):5.
- J. Tang, P. Li, and M. Jiang. 2022. Image caption generation based on knowledge graph guidance and self-attention. *Preprint*.
- M. Toshevska, F. Stojanovska, and J. Kalajdjieski. 2020. [Comparative analysis of word embeddings for capturing word similarities](#). *Preprint*, arXiv:2005.03812.
- G. Wang and 1 others. 2023. UAV-YOLOv8: A small-object-detection model for UAV aerial photography. *Sensors*, 23(16):7190.
- X. Wang and 1 others. 2024. Knowledge enhancement strategies for remote sensing image captioning using CLIP and LLMs. In *Proceedings of MIDA*, pages 313–318.
- X. Zhai and 1 others. 2023. YOLO-Drone: An optimized YOLOv8 for tiny UAV object detection. *Electronics*, 12(17):3664.
- Wentian Zhao and Xinxiao Wu. 2024. [Boosting entity-aware image captioning with multi-modal knowledge graph](#). *IEEE Transactions on Multimedia*, 26:2659–2670.
- Y. Zhou, Y. Sun, and V. Honavar. 2019. Improving image captioning by leveraging knowledge graphs. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 539–548.
- Z. Zou and 1 others. 2021. Disaster image classification by fusing multimodal social media data. *ISPRS International Journal of Geo-Information*, 10(10):636.

# A Appendix

## A.1 Per-Image Score Distributions

This appendix provides per-image CLIPScore and InfoMetIC score distributions for all configurations. These histograms and bar charts complement the summary statistics in Table 1.

### A.1.1 CLIPScore Distributions

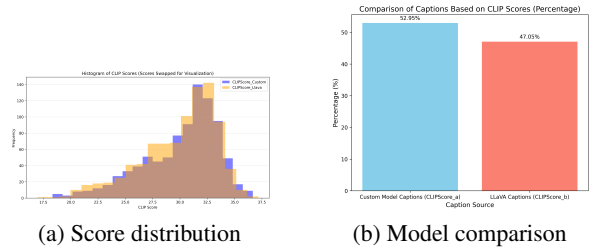


Figure 9: CLIPScore: RescueNet + LLaVA with KG.

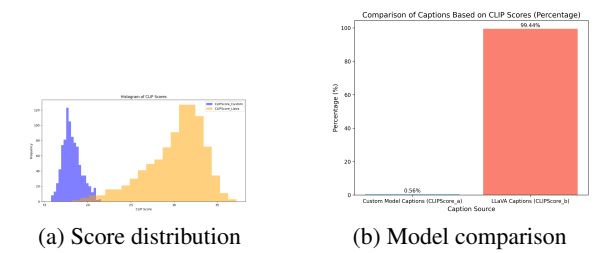


Figure 10: CLIPScore: RescueNet + LLaVA without KG.

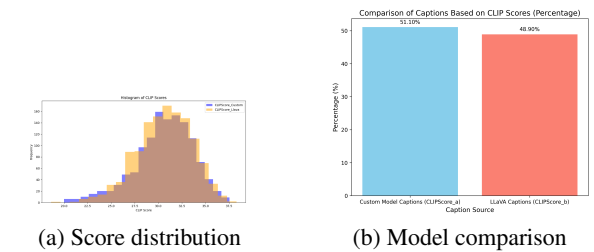


Figure 11: CLIPScore: xBD + LLaVA with KG.

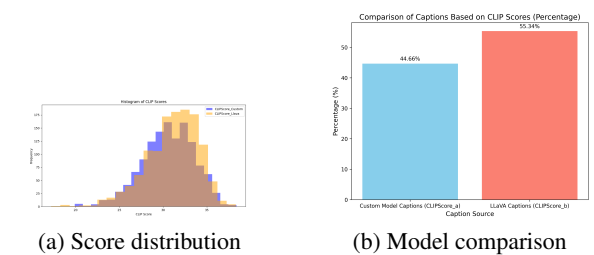


Figure 12: CLIPScore: xBD + LLaVA without KG.

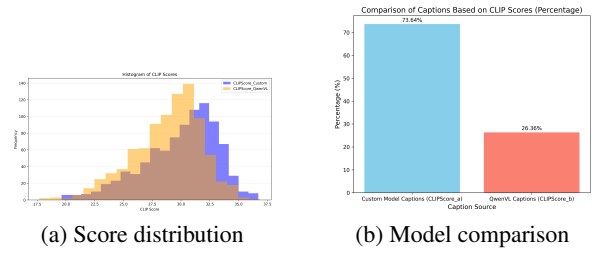


Figure 13: CLIPScore: RescueNet + QwenVL with KG.

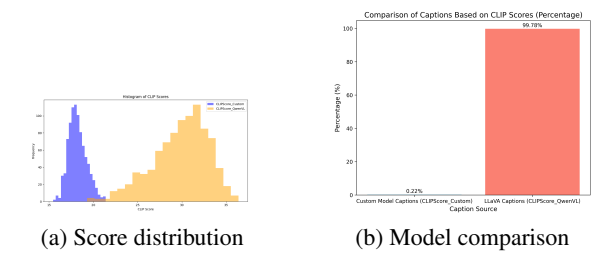


Figure 14: CLIPScore: RescueNet + QwenVL without KG.

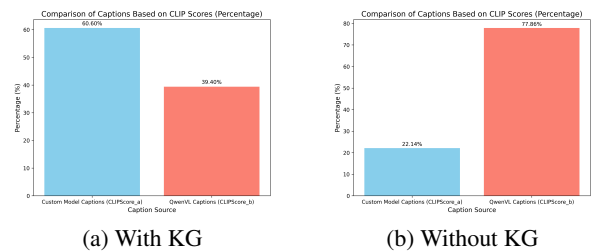


Figure 15: CLIPScore: xBD + QwenVL, with vs. without KG.

### A.1.2 InfoMetIC Distributions

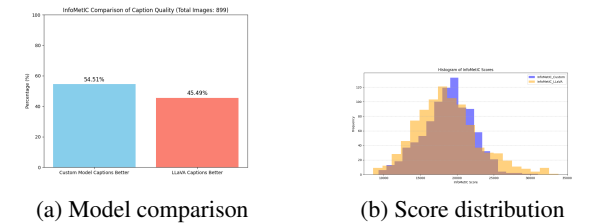


Figure 16: InfoMetIC: RescueNet + LLaVA with KG.

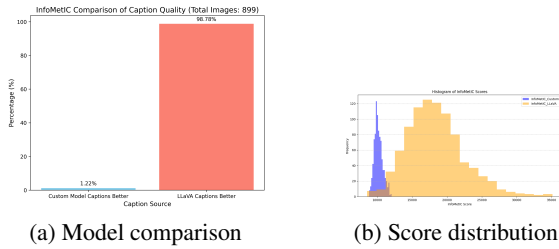


Figure 17: InfoMetIC: RescueNet + LLaVA **without** KG.

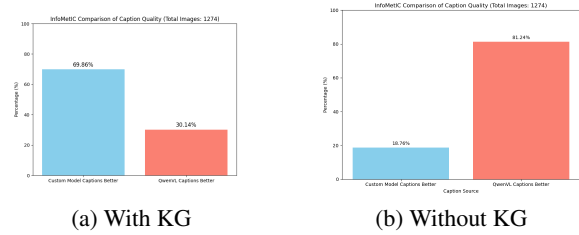


Figure 22: InfoMetIC: xBD + QwenVL, with vs. without KG.

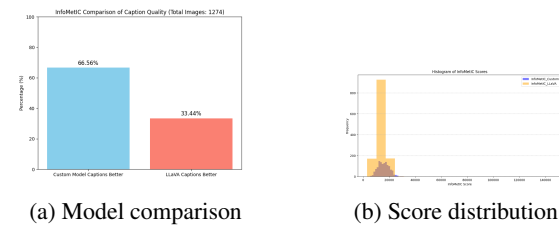


Figure 18: InfoMetIC: xBD + LLaVA **with** KG.

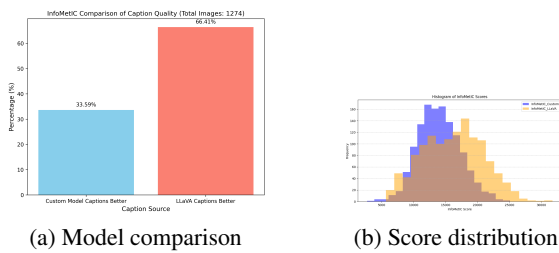


Figure 19: InfoMetIC: xBD + LLaVA **without** KG.

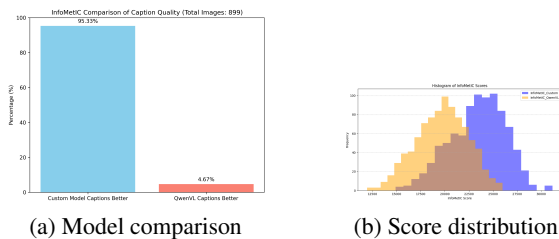


Figure 20: InfoMetIC: RescueNet + QwenVL **with** KG.

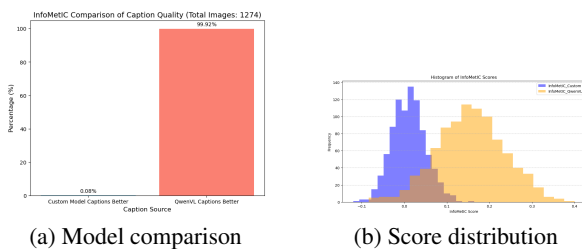


Figure 21: InfoMetIC: RescueNet + QwenVL **without** KG.

## A.2 Qualitative Case Studies

We examine representative examples across dataset-model combinations to illustrate how KG integration affects caption quality.

### A.2.1 RescueNet + LLaVA

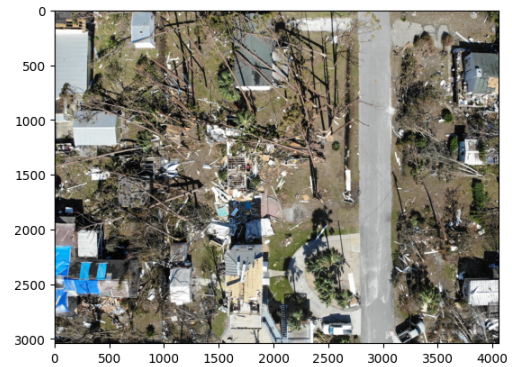


Figure 23: RescueNet with KG-enhanced LLaVA output.

#### KG-Enhanced Caption [23]

**Caption:** the image depicts the aftermath of hurricane michael which caused significant damage to the infrastructure and environment in the affected area **the most noticeable feature is the flooded area which has submerged the debris from damaged structures can be seen scattered throughout the scene** with some debris piled up in the water the presence of multiple vehicles including cars and trucks indicates that the area is being cleared and assessed for recovery efforts the overall scene conveys sense of destruction and recovery efforts needed to restore the affected area to its previous state.

The KG-enhanced caption coherently identifies flooding as the primary damage type, notes submerged debris, and infers recovery operations from visible vehicles.

#### Baseline Caption (No KG) [24]

**Caption:** **the the** image shows an aerial view of a residential severely impacted by a natural disaster fallen trees lie across rooftops, **stone** and streets directly visible structural damage can be seen while the surrounding area is littered with broken branches and debris the stark contrast between intact rooftops and **widespreadvegetation** damage impacted **by a natural disaster fallen trees lie rooftops, stone**

Without KG: word duplication (“the the”), nonsensical terms (“stone”), concatenated words (“widespreadvegetation”), and verbatim phrase repetition.

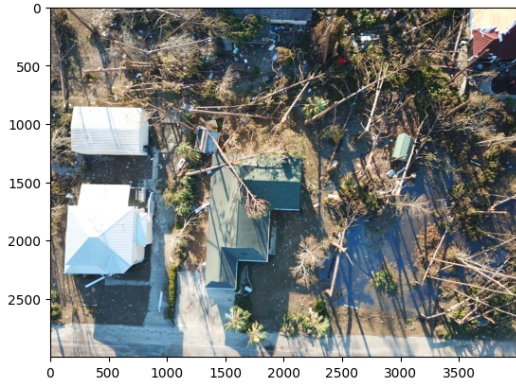


Figure 24: RescueNet without KG, LLaVA output.

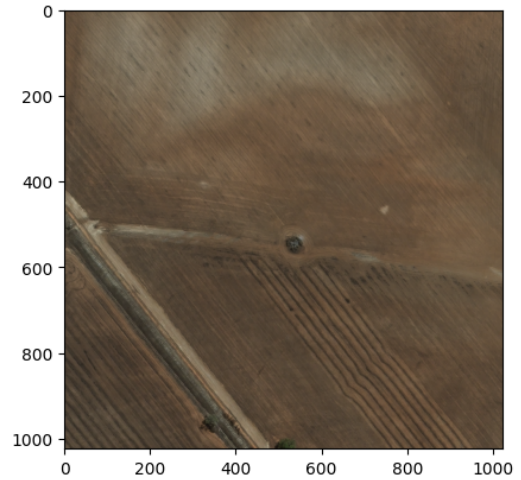


Figure 26: xBD without KG, LLaVA output.

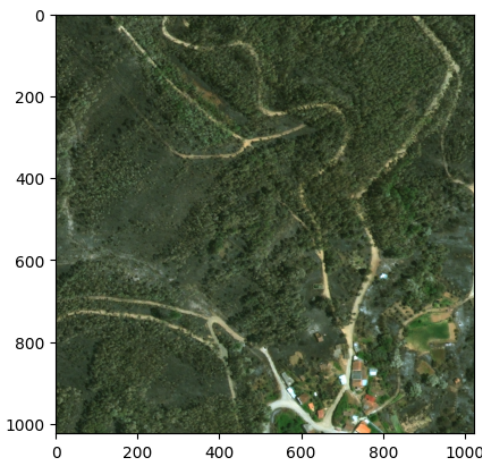


Figure 25: xBD with KG-enhanced LLaVA output.

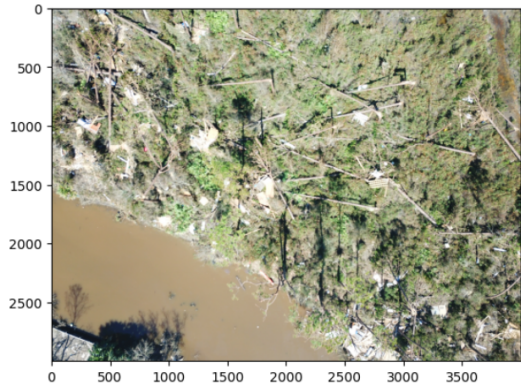


Figure 27: RescueNet with KG-enhanced QwenVL output.

## A.2.2 xBD + LLaVA

### KG-Enhanced Caption [25]

**Caption:** the image shows satellite view of forest with clear path through the trees the path is surrounded by trees and there are several trees in the area the pathway appears to be dirt road and it is surrounded by trees the image also shows few buildings in the area which seem to have escaped the aftermath of the disaster the image captures the aftermath of disaster with undamaged areas or buildings still standing amidst the destruction

### Baseline Caption (No KG) [26]

**Caption:** the image shows satellite view of field with large crack in the middle likely caused by an earthquake the crack extends from the left side of the image to the right dividing the field into two distinct sections additionally there are two people in the field one near the center and the other towards the right side the presence of these holes suggests that the area has been affected by the disaster and the animals deaths may have been temporarily closed due to the damage

The baseline fabricates elements (people in the field, animal deaths) and produces semantically incoherent phrases—a dangerous failure mode for disaster assessment.

## A.3 RescueNet + QwenVL

### KG-Enhanced Caption [27]

**Caption:** the satellite image depicts an area that has been significantly impacted by hurricane the image shows dense concentration of trees that appear to be damaged or fallen indicating the powerful force of the storm the trees are scattered throughout the area with some lying on the ground and others leaning precariously the extent of the destruction is widespread affecting both the natural landscape and built environments the fallen trees and debris suggest that the hurricane had strong impact on the vegetation potentially causing widespread damage to the local ecosystem the presence of debris and fallen trees can obstruct roads and paths making it difficult for emergency services and residents

### Baseline Caption (No KG) [28]

**Caption:** the satellite photo depicts an area that has been significantly impacted by a food area and causing harm to five people of the destruction is evident from the scattered debris and damaged structures here is detailed description of the visible damage and its implications damaged structures there are two large piles of debris scattered across the area these piles consist of various materials such as trees, wood metal and possibly concrete indicating that multiple buildings have been severely damaged or destroyed the debris

The baseline generates semantic inaccuracies (“food area and causing harm to five people”), fabricated casualty figures, and fragmented sentences.



Figure 28: RescueNet without KG, QwenVL output.

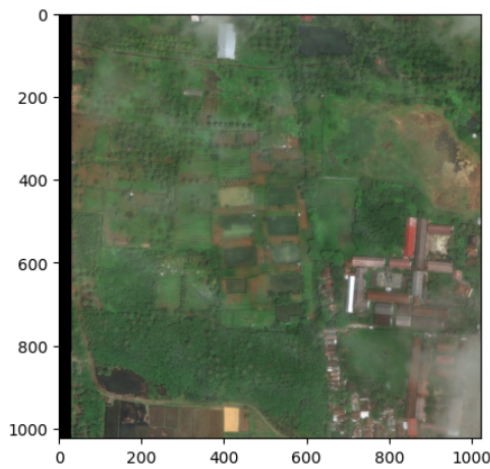


Figure 29: xBD with KG-enhanced QwenVL output.

#### A.4 xBD + QwenVL

##### KG-Enhanced Caption [29]

**Caption:** The image shows **satellite view of town with mix of undamaged and damaged buildings there are undamaged buildings and damaged buildings visible in the scene** the presence of both undamaged and damaged structures highlights the impact of the disaster on the affected area

##### Baseline Caption (No KG) [30]

**Caption:** the satellite image depicts an area that has been significantly impacted by tornado the visible damage is extensive with numerous buildings and structures **showing signs of destruction** **showing signs of destruction** here is detailed description of the visible damage and its implications damage overview damaged areas buildings these areas show visible signs of damage such as showing signs of destruction **collapsed trees broken** and debris scattered around the damage is concentrated in specific areas indicating that the tornado had localized impact **undamaged areas buildings** these areas appear to be relatively intact with no visible signs of damage from the tornado

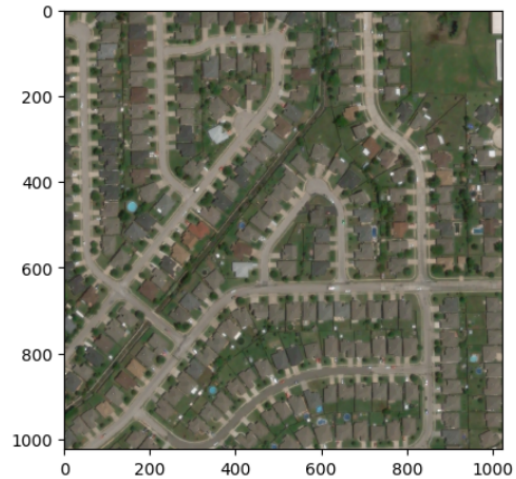


Figure 30: xBD without KG, QwenVL output.

#### A.5 Summary of Failure Modes

Across all configurations, removing KG consistently produces four categories of errors: (1) **hallucinations**- fabricated casualty counts and fictitious elements; (2) **redundancy**- phrases repeated three or more times; (3) **formatting errors**- concatenated words and missing spacing; and (4) **semantic incoherence**- fragmented sentences. KG-enhanced models eliminate these failure modes across all architectures, VLMs, and datasets.

# Beyond Visual Similarity: Rule-Guided Multimodal Clustering with explicit domain rules

Kishor Datta Gupta<sup>1</sup>, Mohd Ariful Haque<sup>1</sup>, Marufa Kamal<sup>2</sup>, Ahmed Rafi Hasan<sup>3</sup>,  
Md. Mahfuzur Rahman<sup>1</sup>, Roy George<sup>1</sup>

<sup>1</sup>Clark Atlanta University, USA,

<sup>2</sup>BRAC University, Bangladesh,

<sup>3</sup> United International University, Bangladesh

Correspondence: [mdmahfuzur.rahman@students.cau.edu](mailto:mdmahfuzur.rahman@students.cau.edu)

## Abstract

Traditional clustering techniques often rely solely on similarity in the input data, limiting their ability to capture structural or semantic constraints that are critical in many domains. We introduce the Domain-Aware Rule-Triggered Variational Autoencoder (DART-VAE), a rule-guided multimodal clustering framework that incorporates domain-specific constraints directly into the representation learning process. DART-VAE extends the VAE architecture by embedding explicit rules, semantic representations, and data-driven features into a unified latent space, while enforcing constraint compliance through rule-consistency and violation penalties in the loss function. Unlike conventional clustering methods that rely only on visual similarity or apply rules as post-hoc filters, DART-VAE treats rules as first-class learning signals. The rules are generated by LLMs, structured into knowledge graphs, and enforced through a loss function combining reconstruction, KL divergence, consistency, and violation penalties. Experiments on aircraft and automotive datasets demonstrate that rule-guided clustering produces more operationally meaningful and interpretable clusters—for example, isolating UAVs, unifying stealth aircraft, or separating SUVs from sedans—while improving traditional clustering metrics. However, the framework faces challenges: LLM-generated rules may hallucinate or conflict, excessive rules risk overfitting, and scaling to complex domains increases computational and consistency difficulties. By combining rule encodings with learned representations, DART-VAE achieves more meaningful and consistent clustering outcomes than purely data-driven models, highlighting the utility of constraint-guided multimodal clustering for complex, knowledge-intensive settings.

## 1 Introduction

Many visual clustering methodologies presume that visual similarity reflects functional similar-

ity; however, appearance and function can diverge in specialized domains. General image clustering methods perform well in natural image domains (Van Gansbeke et al., 2020), yet frequently falter in specialized datasets where operational semantics precede visual appearance. Public benchmark models, while achieving impressive results on large-scale datasets, often fail to transfer effectively when fine-tuned for domain-specific tasks. Vision Transformers (ViTs) have shown promise in this space due to their strong representational capacity, but their success typically relies on very large volumes of training data. Attempting to fine-tune ViTs with limited specialized datasets often results in overfitting (Liu et al., 2021; Zhao et al., 2025), whereas relying solely on public datasets without proper adaptation leads to underfitting and poor generalization in operational settings. Multimodal techniques (Radford et al., 2021; Jia et al., 2021) attempt to bridge this gap by inferring relationships through large-scale data-driven optimization; however, without embedding domain-specific constraints, they risk overlooking the expert knowledge essential for high-stakes applications.

As an illustrative case, stealth bombers and fighters often adopt angular geometries to minimize radar cross-section, yet aircraft with equivalent functions may present markedly distinct visual profiles due to differing manufacturer philosophies and generational design shifts. Similarly, crossover SUVs may appear visually similar, but subtle distinctions in structure and purpose reflect their alignment with separate market segments. These examples highlight that visual similarity alone is insufficient to capture functional or operational equivalence.

Our primary contributions:

- A multimodal VAE architecture encodes visual features, semantic knowledge graphs, and explicit domain rules (Generated by LLM) via specialized pathways, ensuring that learned representations comply with operational constraints.

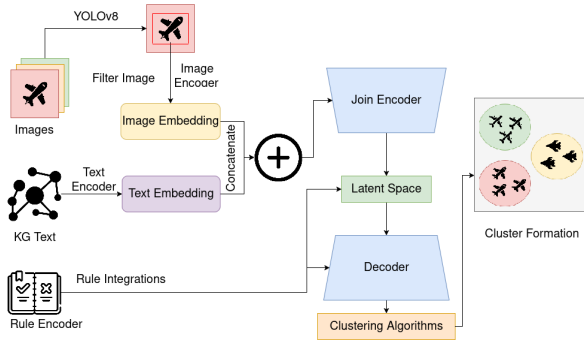


Figure 1: DART-VAE: Domain-Aware Rule-Guided Training for Variational Autoencoders. Our framework integrates visual features, semantic knowledge from structured ontologies, and explicit domain rules (Generated by LLM) through specialized encoders, ensuring learned representations inherently respect operational constraints.

- A multifaceted objective function that equilibrates reconstruction fidelity with rule enforcement through the integration of consistency and violation losses, thereby maintaining generative properties while conforming to expert-defined relationships.

This study formulates a systematic framework for integrating expert knowledge into deep clustering, relevant to domains such as medical imaging, industrial inspection, and scientific analysis, where domain constraints are paramount.

## 2 Related Works

Clustering techniques for visual data have achieved notable success in grouping images based on surface-level similarities. Methods such as k-means, fuzzy c-means, Deep Embedded Clustering (DEC) (Xie et al., 2016), and Variational Deep Embedding (VaDE) (Jiang et al., 2016) optimize purely data-driven objectives but often fail in capturing fine-grained, domain-specific distinctions. In specialized datasets (e.g., vehicles), these approaches neglect semantic and structural knowledge required for meaningful subclass separation. While joint feature learning frameworks such as DAC (Chang et al., 2017) and its refinements with local structure preservation (Guo et al., 2017) and augmentation (Guo et al., 2018) improve robustness, they remain limited to visual similarity. More recent pipelines, including CPP (Chu et al., 2024), leverage pre-trained features for scalability yet still lack explicit integration of domain constraints. Constraint-based clustering has introduced must-link and cannot-link supervision (Ge et al., 2007), and generative models have been applied to encode expert priors (Andreeva et al., 2020). Multimodal approaches enhance interpretability by aligning

visual and semantic features (Chen et al., 2021), and fine-grained methods have advanced through diffusion-based (Yang et al., 2024) and bipartite factorization (Peng et al., 2024). Despite these efforts, existing methods rarely embed domain rules as first-class constraints during representation learning, leaving a gap for frameworks that unify visual, semantic, and rule-based guidance.

## 3 Methodology

### 3.1 Problem Formulation and Motivation

The primary challenge in specialized domain clustering is the disparity between visual appearance and functional purpose. Conventional clustering techniques depend solely on pixel-level or derived visual features, operating under the premise that visually analogous objects possess functional similarities. This assumption fails profoundly in areas where form adheres to highly specialized function rather than aesthetic resemblance.

#### Human vs. Machine Perception Disparity:

Humans possess inherent domain knowledge enabling them to instantly differentiate between a combat fighter and a transport aircraft, despite their similar visual characteristics. Machine learning models, however, lack contextual comprehension and rely solely on superficial visual patterns. For example, the F-16 Fighting Falcon and C-130 Hercules may seem analogous in aerial images due to their monoplane design; however, they fulfill distinctly different operational roles—one as an air superiority fighter and the other as a tactical transport aircraft.

**Operational Reality:** In military aviation, aircraft with nearly identical visual signatures can serve very different functions. The F-22 Raptor and F-117 Nighthawk both possess angular, stealth-optimized designs for radar evasion; however, the F-22 serves as an air superiority fighter with supercruise capability, whereas the F-117 operated as a precision strike bomber. Conversely, functionally analogous aircraft may exhibit significant visual diversity owing to varying design epochs, manufacturers, and technological methodologies.

**Clustering Inadequacy:** Conventional clustering algorithms consistently categorize the MQ-9 Reaper (UAV), F-18 Hornet (fighter), and KC-135 Stratotanker (refueling aircraft) solely based on visual resemblance, resulting in operationally irrelevant clusters that contravene essential military doctrine principles.

Example	Without Rules	With Rules	Improvement
<b>Aircraft Domain</b>			
MQ-9 Reaper (UAV)	Mixed F-18 fighter	Isolated UAV cluster	✓ UAV separation
F-22, Su-57	Scatter multiple clusters	Unified stealth cluster	✓ Technology consistency
C-130, C-2	Mixed A-10 combat	Pure transport cluster	✓ Mission separation
<b>Vehicle Domain</b>			
BMW M3 (Performance)	Mixed Toyota Camry	Performance cluster	✓ Performance separation
Ferrari 488 (Luxury)	Grouped Honda Civic	Luxury sports cluster	✓ Market segmentation
Range Rover (SUV)	Mixed sedan vehicles	SUV cluster	✓ Body style coherence

Table 1: Rule-Guided Clustering Improvement Examples

Our DART-VAE framework addresses this limitation by embedding domain-specific physical rules (which generated by LLM and Domain specific Books) directly into the representation learning process, ensuring that learned embeddings respect both visual coherence and operational semantics. We define this as the acquisition of a latent representation  $z \in \mathbb{R}^d$  that organizes data points based on visual similarity and domain constraints  $R = \{r_j\}_{j=1}^M$ .

### 3.2 Overall Architecture

The DART-VAE framework employs a three-stage pipeline that methodically converts raw multimodal data into constraint-aware latent representations appropriate for domain-informed clustering.

#### Stage 1: Multimodal Feature Extraction

Raw images are subjected to object detection via YOLOv8 to delineate regions of interest (ROI) with adaptive padding, thereby removing background noise that results in erroneous groupings. Concurrently, structured domain knowledge from JSON-formatted knowledge graphs (aircraft) or CSV metadata (vehicles) is processed using SentenceBERT to produce semantic embeddings. These Knowledge graphs and metadata are acquire by LLM fine-tuned by domain-specific contents). Binary rule features are extracted and encoded via specialized MLPs.

#### Stage 2: Constraint-Guided Representation Learning

The DART-VAE encoder processes concatenated multimodal features (visual + semantic + rules) through a joint encoder network that learns the posterior distribution  $q(z|x, t, r)$ . The formation of latent space is directed by a multi-faceted loss function that equilibrates reconstruction accuracy, KL regularization, rule adherence, and penalty for violations.

**Stage 3: Rule-Validated Clustering** Acquired latent representations undergo hard (K-means) and soft (Fuzzy C-means) clustering, followed by rule-

guided refinement, in which constraint violations lead to reassignment to the nearest compliant cluster, based on both latent distance and rule adherence.

### 3.3 Domain-Specific Physical Rules

The core innovation of DART-VAE lies in the explicit formalization of domain knowledge as enforceable constraints. We establish unique rule sets for each domain that encapsulate essential operational principles.

#### Algorithm 1

DART-VAE: Domain-Aware Rule-Triggered Clustering

**Require:** Multimodal dataset  $\mathcal{D} = \{(x_i, t_i, r_i)\}_{i=1}^N$ , Domain rules  $\mathcal{R}$

**Ensure:** Rule-compliant clusters  $\mathcal{C} = \{C_k\}_{k=1}^K$

1: **Stage 1: Multimodal Feature Extraction**

2:  $X_{\text{roi}} \leftarrow \text{ObjectDetection}(\{x_i\})$   $\triangleright$  ROI extraction

3:  $F_v \leftarrow \text{VisualEncoder}(X_{\text{roi}})$   $\triangleright$  Visual features

4:  $F_t \leftarrow \text{SemanticEncoder}(\{t_i\})$   $\triangleright$  Knowledge features

5:  $F_r \leftarrow \text{RuleEncoder}(\{r_i\}, \mathcal{R})$   $\triangleright$  Rule features

6:  $F_{\text{joint}} \leftarrow \text{Concatenate}(F_v, F_t, F_r)$

7: **Stage 2: Constraint-Guided Representation Learning**

8: **for** epoch  $e = 1$  to  $T$  **do**

9:  $\mu, \sigma^2 \leftarrow \text{Encoder}(F_{\text{joint}})$

10:  $Z \leftarrow \text{Reparameterize}(\mu, \sigma^2)$   $\triangleright$  Latent sampling

11:  $\mathcal{L} \leftarrow \mathcal{L}_{\text{recon}} + \beta \mathcal{L}_{\text{KL}} + \alpha_e (\mathcal{L}_{\text{consistency}} + \mathcal{L}_{\text{violation}})$

12: Update  $\theta$  via  $\nabla_{\theta} \mathcal{L}$   $\triangleright$  Progressive rule integration

13: **end for**

14: **Stage 3: Rule-Validated Clustering**

15:  $C_{\text{init}} \leftarrow \text{Clustering}(Z)$   $\triangleright$  K-means or Fuzzy C-means

16: **for** each constraint  $r \in \mathcal{R}$  **do**

17: **for** each cluster  $C_k \in C_{\text{init}}$  **do**

18: **if** ViolatesRule( $C_k, r$ ) **then**

19: Reassign violating samples to nearest compliant clusters

20: **end if**

21: **end for**

22: **end for**

23: **return**  $\mathcal{C}$

#### 3.3.1 Aircraft Domain Rules

According to military aviation doctrine and aerospace engineering principles, a fine-tuned LLM generated four essential constraints:

##### Rule 1: Stealth Technology Consistency

Stealth aircraft represent a highly specialized tech-

nological category requiring sophisticated systems integration. Aircraft with stealth capabilities must exhibit technological reliability via advanced avionics systems and possess either air superiority (fighter classification) or supersonic cruise capability.

*Formal Definition:*  $\forall a \in \text{Aircraft} : \text{is\_stealth}(a) \rightarrow (\text{has\_advanced\_avionics}(a) \wedge (\text{is\_fighter}(a) \vee \text{has\_supercruise}(a)))$

*Physical Rationale:* Stealth technology necessitates advanced radar systems, electronic warfare capabilities, and intricate flight controls. The F-22 Raptor integrates stealth capabilities with supercruise, whereas the F-117 Nighthawk depended on sophisticated avionics for precision strike operations.

**Rule 2: UAV Operational Separation** Unmanned and manned aircraft function under distinct doctrines, certification criteria, and operational protocols. They must uphold distinct clustering boundaries to accurately represent operational reality.

*Formal Definition:*  $\forall a_i, a_j \in C_k : \text{is\_uav}(a_i) \leftrightarrow \text{is\_uav}(a_j)$

*Physical Rationale:* UAVs such as the MQ-9 Reaper function with different risk profiles, endurance capacities, and mission specifications in contrast to manned aircraft like the F-16.

**Rule 3: Mission-Type Doctrinal Enforcement** Military aircraft are designed and optimized for specific mission profiles. Combat platforms must not be grouped with transport or logistics aircraft due to inherent disparities in operational requirements, threat environments, and deployment patterns.

*Formal Definition:*  $\forall a_i, a_j \in C_k : \text{mission\_type}(a_i) = \text{combat} \rightarrow \text{mission\_type}(a_j) \neq \text{transport}$

*Physical Rationale:* Combat aircraft like the A-10 Thunderbolt II are armored for survivability in hostile environments, while transport aircraft like the C-130 prioritize cargo capacity and operational versatility.

**Rule 4: Physical Attribute Coherence** Aircraft within the same operational cluster must demonstrate comparable fundamental physical attributes, including propulsion systems and performance envelopes, indicative of analogous operational requirements.

*Formal Definition:*  $\forall a_i, a_j \in C_k : \text{engine\_type}(a_i) = \text{engine\_type}(a_j) \wedge \text{speed\_class}(a_i) = \text{speed\_class}(a_j)$

*Physical Rationale:* Turbofan-powered aircraft function within individual performance parameters compared to turboprop aircraft, influencing range,

altitude capabilities, and mission appropriateness.

### 3.3.2 Automotive Domain Rules

For vehicles, OPENAI GPT3.5 outlines four rules that encapsulate market segmentation and engineering principles for vehicles.

**Rule 1: Body Style Coherence** Vehicles with fundamentally diverse body structures cater to specific market niches and usage patterns. SUVs, sedans, and convertibles cater to their own market requirements and should maintain cluster differentiation.

**Rule 2: Performance Tier Consistency** Economy cars and high-performance cars have different engineering aims and target various market segments. Performance cars put handling and power-to-weight ratios first, whereas economy cars maximize cost and fuel efficiency.

**Rule 3: Dimensional Proportionality** Vehicles with notably distinct physical proportions (height-to-length ratios, ground clearance) fulfill diverse practical functions and should remain distinct in clustering.

**Rule 4: Luxury Market Segmentation** Luxury and standard market vehicles represent distinct value propositions with different feature sets, pricing strategies, and brand positioning.

**Semantic Knowledge Encoding:** Sentence-BERT (all-mpnet-base-v2) processes structured domain expertise. Aircraft use JSON-formatted knowledge graph triples with technical specifications, operational roles, and performance attributes contained in 384D vectors and compressed to 256D via 2-layer MLP. Sentence-BERT to 768D embeddings procedure CSV brand hierarchies, technical specifications, body styles, and market categories for vehicles.

**Rule Feature Engineering:** Domain constraints are implemented as explicit feature vectors via specialized MLPs. Aircraft utilize 10 binary attributes (is\_stealth, is\_uav, has\_crew, has\_supercruise, has\_advanced\_avionics, mission\_type indications) processed using MLP to 16-dimensional space. Vehicles utilize 18 derived attributes from the four automotive rules (body\_style\_category, performance\_tier, size\_ratios, luxury\_indicators) compressed into a 32-dimensional space.

### 3.4 DART-VAE Architecture Details

**Joint Encoding:** The multimodal feature fusion creates domain-specific joint representations. Aircraft:  $f_{\text{joint}} \in \mathbb{R}^{100,624}$  (visual: 100,352D + seman-

tic: 256D + rules: 16D). Vehicles:  $f_{\text{joint}} \in \mathbb{R}^{100,388}$  (visual: 100,352D + semantic: 768D + rules: 32D).

**Encoder Network:** A progressive compression architecture maps joint features to latent parameters:  $h_1 = \text{ReLU}(\text{Linear}(f_{\text{joint}}, 512))$ ,  $h_2 = \text{ReLU}(\text{Linear}(h_1, 256))$ ,  $\mu, \log \sigma^2 = \text{Linear}(h_2, 64), \text{Linear}(h_2, 64)$ .

**Latent Sampling:** The reparameterization trick enables gradient-based optimization:  $z = \mu + \sigma \odot \epsilon$ , where  $\epsilon \sim \mathcal{N}(0, I_{64})$ .

**Decoder Network:** Reconstructs multimodal features to ensure representation fidelity:  $h_3 = \text{ReLU}(\text{Linear}(z, 256))$ ,  $h_4 = \text{ReLU}(\text{Linear}(h_3, 512))$ ,  $f_{\text{reconstructed}} = \text{Linear}(h_4, \text{dim}(f_{\text{joint}}))$ .

### 3.5 Multi-Component Loss Function

Our training objective combines traditional VAE losses with rule-specific penalties that address the fundamental challenge observed in our aircraft and automotive clustering experiments:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \beta \mathcal{L}_{\text{KL}} + \alpha (\mathcal{L}_{\text{consistency}} + \mathcal{L}_{\text{violation}}) \quad (1)$$

**Reconstruction Loss ( $\mathcal{L}_{\text{recon}}$ ):** Standard VAE reconstruction ensures that our multimodal features; visual ROI features, knowledge graph embeddings, and rule encodings, can be faithfully recovered from the latent space:

$$\mathcal{L}_{\text{recon}} = \text{MSE}(f_{\text{joint}}, \hat{f}_{\text{joint}}) \quad (2)$$

**KL Divergence Loss ( $\mathcal{L}_{\text{KL}}$ ):** The standard VAE regularization prevents latent space collapse:

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2} \sum_{i=1}^d (1 + \log \sigma_i^2 - \mu_i^2 - \sigma_i^2) \quad (3)$$

**Rule Consistency Loss ( $\mathcal{L}_{\text{consistency}}$ ):** This component indicates a key finding from our experiments: aircraft with analogous operational profiles should exhibit comparable latent representations, despite visual differences.

$$\mathcal{L}_{\text{consistency}} = \sum_{i,j} \text{MSE}(\text{sim}(z_i, z_j), \text{sim}(r_i, r_j)) \quad (4)$$

The function  $\text{sim}(r_i, r_j)$  estimates cosine similarity between 16-dimensional rule features generated by the rule encoder, rather than raw binary inputs. This groups the MQ-9 and TB2 UAVs despite their visual profiles and aligns BMW and Mercedes

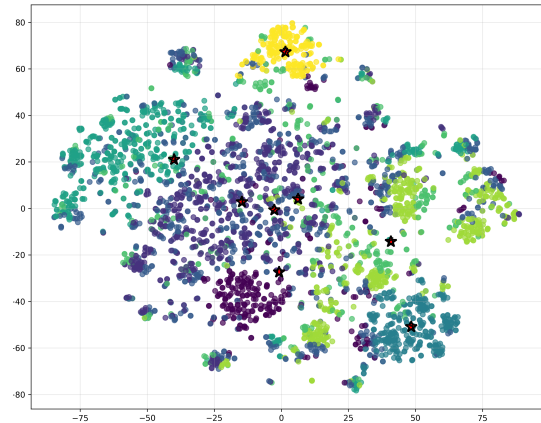
premium automobiles in latent space despite brand distinctions.

**Rule Violation Loss ( $\mathcal{L}_{\text{violation}}$ ):** Direct constraint enforcement emerges from our domain analysis:

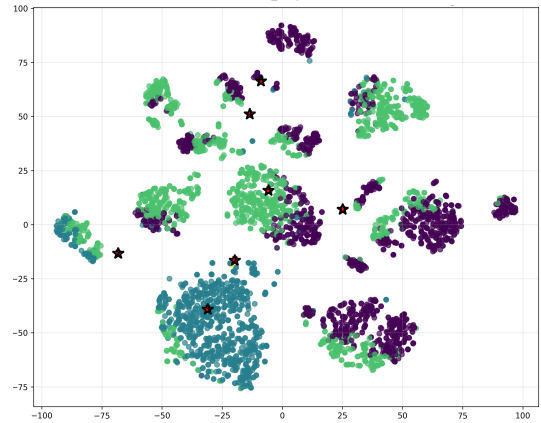
$$\mathcal{L}_{\text{violation}} = \text{MSE}(\sigma(v_{\text{pred}}), v_{\text{target}}) \quad (5)$$

where  $v_{\text{pred}} \in \mathbb{R}^{N \times 4}$  are the raw violation predictions from the rule predictor network,  $\sigma$  is the sigmoid activation, and  $v_{\text{target}} \in \{0, 1\}^{N \times 4}$  are the binary violation targets computed from the logical rules.

**Rule Weight Configuration:** Reconstruction fidelity and domain constraint enforcement are balanced using a 0.15 rule weight in airplane (40 epochs) and automobile (30 epochs) domain training.



(a) Without physical rules



(b) With 4 physical rules

Figure 2: Car Hard Clustering t-sne visualization

## 4 Experimental Results and Analysis

We assess DART-VAE in the aircraft and automotive domains through extensive experimental setups to illustrate the incremental advantages of including domain constraints.

## 4.1 Experiments

All experiments employ appropriate statistical validation and adhere to the implementation specifics outlined in Section 4.1 and appendix. We applied a two-stage filtering pipeline (Mask R-CNN + YOLOv8, confidence  $\geq 0.97$ , coverage  $\geq 60\%$ ) to curate high-quality images, selecting a representative subset (800 aircraft images across 76 classes) for efficient evaluation. Both datasets are enriched with structured metadata (JSON/CSV) capturing technical specifications and attributes for downstream clustering and analysis.

## 4.2 Aircraft Domain Clustering Results

### 4.2.1 Quantitative Performance Analysis

Table 2 breaks down comprehensive aircraft clustering outcomes across various rule configurations and techniques. The methodical advancement from baseline to rule-based grouping reveals distinct performance variation contingent upon the complexity of constraints. **Hard Clustering Performance:** The 2-rule hard configuration achieves remarkable quantitative measures, evidenced by a Silhouette Score of 0.7109 and a robust Calinski-Harabasz Score of 16,325.64, reflecting a 405% enhancement over the baseline hard clustering score of 0.1406. However, the 4-rule hard configuration exhibits diminished traditional metrics (Silhouette: 0.3325) while achieving improved Davies-Bouldin performance (0.9147), which means greater cluster compactness despite lowered separation scores. **Soft Clustering Performance:** Fuzzy C-means with 2-rule configuration achieves remarkable performance with Fuzzy Partition Coefficient of 0.9765 and minimal Fuzzy Partition Entropy of 0.0474, demonstrating optimal cluster separation. Table 2 shows the fuzzy-specific metrics where the 2-rule configuration excels with high certainty in cluster assignments and Average Membership Strength of 0.9850. The 4-rule soft configuration achieves Fuzzy Partition Coefficient of 0.4736 and Fuzzy Partition Entropy of 1.1456, with Average Membership Strength of 0.6184, trading traditional fuzzy metrics for comprehensive constraint coverage.

### 4.2.2 Rule Violation Analysis

**Stealth Consistency Constraints:** Violations of stealth technology persist at a consistent rate of 85 across all configurations, signifying structural issues within the stealth aircraft category that surpass clustering algorithms. This ongoing violation

pattern indicates intrinsic data complexity, wherein stealth qualities do not fully correspond with other operational features. **UAV Operational Separation:** Hard clustering with rule refinement provides perfect UAV separation (0 violations) by post-processing optimization, while soft clustering keeps 90-214 violations but accepts boundary scenarios when UAV characteristics overlap with manned aircraft traits. With increasing rule complexity, constraint management improves from 214 UAV violations in the 2-rule soft setup to 90 in the 4-rule version. **Mission and Semantic Coherence:** The four-rule configurations impose constraints related to mission type and semantic coherence, resulting in 401 to 533 mission violations and 0 to 103 semantic violations. While these increase overall constraint violations, they provide fine-grained operational classification essential for military applications.

### 4.2.3 Visual Clustering Analysis

Figure 3 illustrates t-SNE visualizations that depict the evolution of clustering across various rule configurations. The 2-rule guided clustering (Figure 3b and 3e) demonstrates effective cluster separation, resulting in well-defined operational groupings. The configurations with four rules (Figure 3c and 3f) demonstrate more intricate boundaries that indicate a higher level of constraint complexity, whereas Figure 3a presents the baseline performance in the absence of rule guidance.

**Cluster Coherence:** Rule-guided clustering efficiently creates operational coherence, as stealth platforms (F-22, Su-57, F-117) cluster according to technological uniformity, UAVs maintain clear operational distinctions from manned aircraft, and transport aircraft (C-130, C-2) are differentiated from combat platforms (A-10). This indicates a significant shift from visual similarity to functional effectiveness.

## 4.3 Automotive Domain Clustering Results

### 4.3.1 Quantitative Performance Analysis

Table 3 illustrates the performance of hard clustering in the automotive sector at various levels of rule integration. Rule-based hard clustering demonstrates significant enhancements: The Silhouette Score rises from 0.0543 (baseline) to 0.1393 (156% enhancement), while the Calinski-Harabasz Score escalates from 173.82 to 12,234.91 (6,941% enhancement), signifying significantly enhanced cluster separation and compactness.

Table 2: Aircraft clustering performance comparison (K-means vs. Fuzzy C-means) across rule configurations

Configuration	Hard Clustering (K-means)				Rule Violations (Hard)			
	SS	DB	CH		Stealth	UAV	Mission	Semantic
Baseline	0.1406	1.5167	174.37	–	–	–	–	–
2-Rule	0.7109	1.0666	16325.64	–	85	0	–	–
4-Rule	0.3325	0.9147	794.43	–	85	0	401	103
Configuration	Soft Clustering (Fuzzy C-means)				Rule Violations (Soft)			
	FPC	FPE	MS	FS	Stealth	UAV	Mission	Semantic
Baseline	0.5401	0.9775	0.396	–	–	–	–	–
2-Rule	0.9765	0.0474	0.9850	–	85	214	–	–
4-Rule	0.4736	1.1456	0.6184	0.1960	85	90	533	0

SS: Silhouette Score, DB: Davies-Bouldin Score, CH: Calinski-Harabasz Score,

FPC: Fuzzy Partition Coefficient, FPE: Fuzzy Partition Entropy, MS: Average Membership Strength, FS: Fuzzy Silhouette Index

Table 3: Automotive clustering performance comparison (K-means vs. Fuzzy C-means)

	FPC	FPE	MS	FS
Baseline (Soft)	0.125	2.07	0.125	0.167
Rule-Guided (Soft)	0.125	2.08	0.125	0.189
	SS	DB	CH	
Baseline (Hard)	0.054	3.27	173.8	–
Rule-Guided (Hard)	0.139	0.92	12234.9	–
Rule Violations (Soft vs. Hard, Rule-Guided)				
Metric	Soft	Hard		
Body	0.19	3.01		
Performance	3.60	4.83		
Size	2.00	3.92		
Luxury	2.72	4.53		

Total violations: Soft = 8.51; Hard = 16.29

### 4.3.2 Rule Violation Analysis

**Body Style Coherence:** The use of rule-guided clustering results in differing degrees of Body Style/Segment violations according on the clustering algorithm utilized. Tables 3 illustrate that fuzzy C-means clustering exhibits superior constraint management with merely 0.19 violations, whereas K-means clustering results in 3.01 violations. This notable disparity demonstrates fuzzy clustering’s efficacy in managing the diverse classifications of body shapes in the automotive sector, including SUV, sedan, and hatchback, where automobiles may display ambiguous traits.

**Performance and Engineering Constraints:** Performance/Drivetrain Consistency is the hardest criteria for both clustering methods, with soft clustering (fuzzy C-means) obtaining 3.60 violations and hard clustering (K-means) 4.83. High violation rates reflect the car industry’s sophisticated performance specs that don’t match visual similarities—high-performance variations of ordinary models sometimes look identical despite having very different powertrains. Size Proportion limitations yield mild violations, with fuzzy C-means achieving 2.00 and K-means 3.92. Luxury/Performance Feature consistency provides sufficient con-

straint adherence, with fuzzy C-means committing 2.72 violations and K-means 4.53.

**Algorithm Comparison:** The experimental results indicate complementary benefits among clustering methodologies. K-means demonstrates superior mathematical clustering effectiveness, as indicated by a higher Silhouette Score (0.1393 compared to 0.0100) and better cluster separation metrics, making it suitable for applications that prioritize geometric cluster quality. Fuzzy C-means shows improved adherence to domain constraints, with 48% fewer rule violations (8.51 versus 16.29), making it more suitable for applications that require semantic coherence over mathematical optimization. Figure 2 illustrates that rule-guided K-means clustering (Figure 2b) achieves remarkable separation with 8 distinct clusters, whereas baseline clustering (Figure 2a) exhibits considerable mixing.

### 4.4 Qualitative Clustering Improvements

Table 1 demonstrates specific clustering improvements through rule integration. Critical operational separations achieved include: **Military Applications:** MQ-9 Reaper UAVs isolated from F-18 fighters, stealth aircraft (F-22, Su-57) unified by technological consistency, and transport aircraft (C-130, C-2) separated from combat platforms. **Automotive Applications:** BMW M3 performance vehicles separated from economy cars, Ferrari luxury sports cars grouped appropriately, and Range Rover SUVs clustered by body style coherence. These improvements represent fundamental advances from appearance-based to function-based clustering, critical for domain expert applications where operational semantics transcend visual similarity.

## 5 Threats to Validity

Several threats to validity remain. First, the rules themselves are generated by LLMs and subse-

quently formatted as structured knowledge graphs. This process introduces the risk of hallucinations, where the LLM may produce inaccurate or spurious rules. Such negative rules can inadvertently bias the clustering process, leading to distortions rather than improvements in cluster quality. Second, the reliance on LLM-generated rules makes the framework sensitive to overfitting. When too many rules are imposed simultaneously, the latent space may become overly constrained, forcing the model to adhere to rigid relationships at the expense of generalizability. This effect was particularly evident when scaling from two-rule to four-rule configurations, where clustering metrics showed diminishing returns despite improved constraint enforcement. Third, scaling to complex domains presents a fundamental challenge. As the number of rules, modalities, and semantic categories increases, the computational overhead and difficulty of maintaining consistent enforcement escalate. In such cases, rule conflicts and inconsistencies may proliferate, complicating both training and interpretability. Without careful curation and validation of rule sets, domain expansion risks undermining the stability and reliability of the method. Finally, the evaluation relies on datasets where rule definitions are relatively well aligned with domain expertise (e.g., aircraft doctrine, automotive market segmentation). For less formalized or more ambiguous domains, the suitability of LLM-derived rules remains uncertain, limiting the external validity of the approach.

## 6 Conclusion

Proposed multimodal clustering framework that elevates LLM-generated rules and knowledge-graph constraints to first-class learning signals. By embedding rules directly into representation learning, the method balances visual coherence and operational semantics, producing clusters that are both interpretable and quantitatively robust. Results in aircraft and automotive domains show that rule-guided clustering achieves clearer functional separation than purely visual baselines. However, reliance on LLM-generated rules introduces risks of hallucination and inconsistency, and applying too many constraints can lead to overfitting. Scaling to complex domains with large rule sets remains challenging. Despite these limitations, DART-VAE demonstrates the potential of rule-informed clustering as a principled step toward interpretable and domain-aligned AI.

## References

- a2015003713. [Military aircraft detection dataset](#). Kaggle dataset. Accessed: 2025-01-17.
- Olga Andreeva, Wei Li, Wei Ding, Marieke Kuijjer, John Quackenbush, and Ping Chen. 2020. Catalysis clustering with gan by incorporating domain knowledge. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1344–1352.
- Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. 2017. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5879–5887.
- Brian Chen, Andrew Rouditchenko, Kevin Duarte, Hilde Kuehne, Samuel Thomas, Angie Boggust, Rameswar Panda, Brian Kingsbury, Rogerio Feris, David Harwath, and 1 others. 2021. Multimodal clustering networks for self-supervised learning from unlabeled videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8012–8021.
- Tianzhe Chu Chu, Shengbang Tong, Tianjiao Ding, Xili Dai, Benjamin Haeffele, Rene Vidal, and Yi Ma. 2024. Image clustering via the principle of rate reduction in the age of pretrained models. International Conference on Learning Representations (ICLR).
- Rong Ge, Martin Ester, Wen Jin, and Ian Davidson. 2007. Constraint-driven clustering. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 320–329.
- Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. 2017. Improved deep embedded clustering with local structure preservation. In *Ijcai*, volume 17, pages 1753–1759.
- Xifeng Guo, En Zhu, Xinwang Liu, and Jianping Yin. 2018. Deep embedded clustering with data augmentation. In *Asian conference on machine learning*, pages 550–565. PMLR.
- Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.
- Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. 2016. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*.
- Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. [Ultralytics yolov8](#).
- Zhenda Liu, Han Chen, Yujie Feng, Songtao Liu, Jizhou He, Jie Zhou, and James Zou. 2021. [Efficient training of visual transformers with small datasets](#). In

*Advances in Neural Information Processing Systems (NeurIPS)*.

Chong Peng, Pengfei Zhang, Yongyong Chen, Zhao Kang, Chenglizhao Chen, and Qiang Cheng. 2024. Fine-grained bipartite concept factorization for clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26264–26274.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmlR.

Unit293. 2024. [Car-models-3887: 3D meshes of 3887 car types](#). Hugging Face dataset. Accessed: 2024-09-17.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. 2020. Scan: Learning to classify images without labels. In *Computer Vision – ECCV 2020*, pages 268–285, Cham. Springer International Publishing.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.

Ruohong Yang, Peng Hu, Xi Peng, Xiting Liu, and Yunfan Li. 2024. Dific: Your diffusion model holds the secret to fine-grained clustering. *arXiv preprint arXiv:2412.18838*.

Chen Zhao, Ziqian Chen, Li Zhang, and Yan Wang. 2025. [The missing piece in vit fine-tuning for image aesthetic assessment](#). *arXiv preprint arXiv:2504.02522*.

## 7 Appendix

### 7.1 Dataset Details

**Military Aircraft:** We conducted a two-stage data cleansing process on 8,311 raw photos from the military aircraft dataset ([a2015003713](#)). Preliminary filtering employing Mask R-CNN, succeeded by YOLOv8 (Jocher et al., 2023) with dual parameters (confidence  $\geq 0.97$  and aircraft coverage  $\geq 60\%$ ), produced 3,103 high-quality images encompassing 77 aircraft types. To ensure computational efficiency and uniform assessment across rule configurations, we chose a representative subset of 800 photos (after YOLOv8 filtering, preserving 728 images with aircraft detections) encompassing 76 aircraft classes for our clustering tests. Every aircraft is marked with operating characteristics and connected to a JSON knowledge graph that

includes technical specifications as triples. **Automotive Dataset:** The vehicle dataset (Unit293, 2024) was subjected to quality control comparable to aircraft photos, using YOLOv8 to exclude non-vehicle images, notably interiors. A hierarchical brand/model/year directory structure and CSV metadata with technical specifications, body styles, segments, and dimensional attributes from manufacturer databases comprise the cleaned dataset.

### 7.2 Implementation Details

All tests utilize the PyTorch framework using NVIDIA RTX 6000 Ada Generation GPUs, each equipped with 49GB of VRAM. The visual encoder utilizes a custom CNN architecture with domain-specific configurations, comprising three convolutional layers for aircraft and four for cars. Text encoding using Sentence-BERT (all-mpnet-base-v2). In the preprocessing phase, YOLOv8 does object detection with a confidence threshold of 0.25 and adaptive padding for region of interest extraction. Optimization employs AdamW with a weight decay of  $\lambda = 10^{-4}$ . Training setups are specialized to their domains: airplane experiments utilize 40 epochs across all rule configurations, whereas car experiments employ 30 epochs with a 4-rule configuration.

### 7.3 Multimodal Feature Extraction Pipeline

**Visual Feature Processing:** We utilize YOLOv8 for object detection and ROI extraction, implementing adaptive padding to encompass entire aircraft and vehicle structures while minimizing background noise. Our experimental validation demonstrates that background contamination substantially affects clustering quality, rendering ROI extraction an essential preprocessing step.

*Aircraft Domain:* Three convolutional layers  $\text{Conv2d}(3,32) \rightarrow \text{Conv2d}(32,64) \rightarrow \text{Conv2d}(64,128)$ , flattening yields 100,352D feature vectors ( $128 \times 28 \times 28$ ), 20% adaptive padding applied during ROI extraction.

*Automotive Domain:* Four convolutional layers with adaptive average pooling, culminating in a 256-dimensional visual feature representation, with a confidence threshold of 0.25 for vehicle detection.

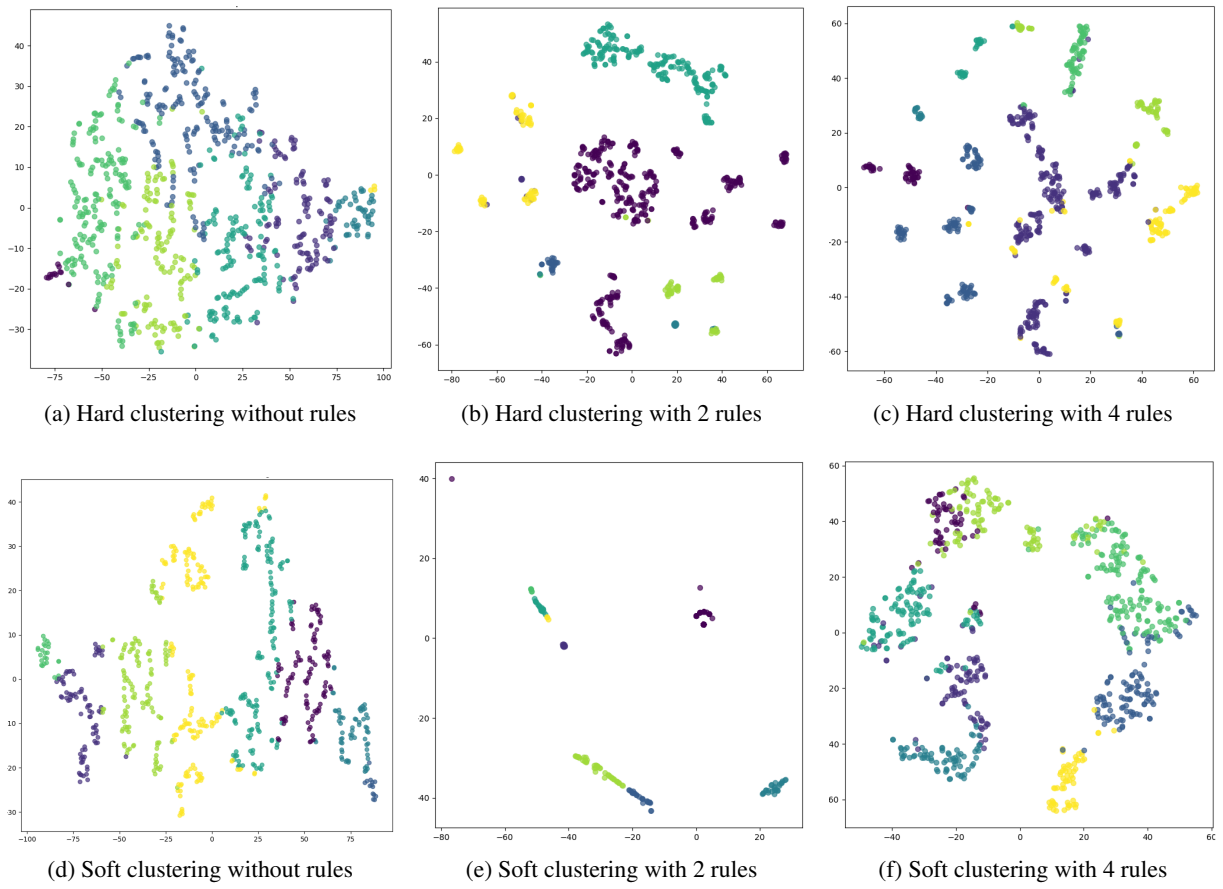


Figure 3: Aircraft clustering performance demonstrated through t-SNE visualizations. Top row shows hard clustering (K-means), bottom row shows soft clustering (Fuzzy C-means), progressing from no rules (left) to 2 rules (center) to 4 rules (right).

# ChartDiff: A Large-Scale Benchmark for Comprehending Pairs of Charts

Rongtian Ye

Department of Computer Science, Aalto University

rongtian.ye@aalto.fi

<https://ckchaos.github.io/ChartDiff>

## Abstract

Charts are central to analytical reasoning, yet existing benchmarks for chart understanding focus almost exclusively on single-chart interpretation rather than comparative reasoning across multiple charts. To address this gap, we introduce **ChartDiff**, the first large-scale benchmark for cross-chart comparative summarization. ChartDiff consists of 8,541 chart pairs spanning diverse data sources, chart types, and visual styles, each annotated with LLM-generated and human-verified summaries describing differences in trends, fluctuations, and anomalies. Using ChartDiff, we evaluate general-purpose, chart-specialized, and pipeline-based models. Our results show that frontier general-purpose models achieve the highest GPT-based quality, while specialized and pipeline-based methods obtain higher ROUGE scores but lower human-aligned evaluation, revealing a clear mismatch between lexical overlap and actual summary quality. We further find that multi-series charts remain challenging across model families, whereas strong end-to-end models are relatively robust to differences in plotting libraries. Overall, our findings demonstrate that comparative chart reasoning remains a significant challenge for current vision-language models and position ChartDiff as a new benchmark for advancing research on multi-chart understanding.

## 1 Introduction

Charts play a central role in analytical reasoning, communication, and decision-making. From scientific publications to business dashboards, chart presentations allow humans to quickly extract patterns, compare trends, and evaluate alternatives. As vision-language models (VLMs) (Radford et al., 2021; Li et al., 2022; Liu et al., 2023c; Bai et al., 2023) continue to advance, enabling them to understand and explain charts (Han et al., 2023; Masry et al., 2023; Zhang et al., 2024) has become an increasingly active research area. Recent

progress has produced a variety of benchmarks and methods targeting tasks such as chart question answering (Masry et al., 2022; Wang et al., 2024), chart summarization (Kantharaj et al., 2022; Meng et al., 2024), structured information extraction (Liu et al., 2023b,a). Despite these advances, existing work (Masry et al., 2022, 2023; Han et al., 2023; Zhang et al., 2024; Masry et al., 2025a; Meng et al., 2024; Zhao et al., 2025; Xu et al., 2025) overwhelmingly focuses on single-chart understanding, treating each chart as an isolated unit.

However, many real-world analytical tasks are inherently comparative. Analysts routinely juxtapose multiple charts to evaluate differences across groups, time periods, experimental conditions, or modeling assumptions. Detecting how two charts differ—whether in their underlying data, visual encodings, statistical relationships, or narrative intent—is central to tasks such as anomaly detection, model comparison, A/B testing, monitoring system performance, or verifying the reproducibility of results. Yet, to date, though some works (Zhu et al., 2025; Iyengar et al., 2025; Kazemi et al., 2024) have been proposed in multi-chart setting, the ability of VLMs to perform such comparative reasoning remains largely unexplored.

To address this gap, we introduce a new large-scale benchmark **ChartDiff** (Dataset illustration in Figure 1) designed specifically for cross-chart difference comprehension. ChartDiff comprises 8,541 pairs of charts, covering a wide range of data sources, visualization types, and styles. For each pair, we provide LLM-generated and human-verified annotations describing differences in trends, fluctuations, and anomalies. These annotations form a rigorous testbed for evaluating whether VLMs can move beyond single-image interpretation and engage in comparative visual reasoning.

Using this benchmark, we conduct the first systematic evaluation of state-of-the-art VLMs on

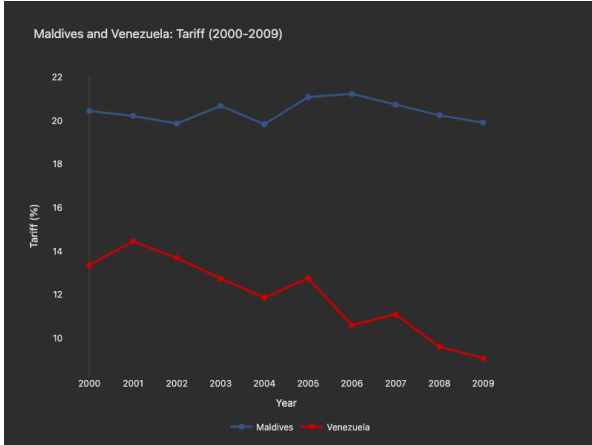


Chart A (Left)

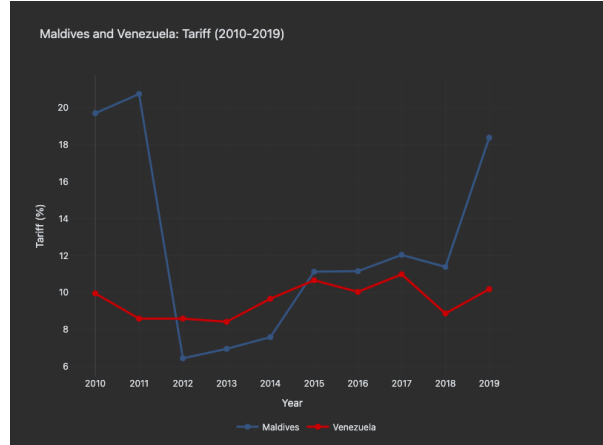


Chart B (Right)

**Comparison Summary:** From 2000 to 2009, Maldives maintained a high and fairly stable tariff around 20–21%, while Venezuela showed a clear downward trend from 13.37% to 9.09% with only minor fluctuations. In 2010–2019, Maldives changed much more dramatically: after staying near 20% in 2010–2011, its tariff collapsed to 6.45% in 2012, then gradually recovered through the decade before rebounding sharply to 18.39% in 2019. Venezuela, by contrast, remained relatively steady in the later period, mostly fluctuating within a narrower 8.4–11.0% range after its earlier decline. Overall, the key shift is that Maldives moved from stability to major volatility and structural change, whereas Venezuela moved from a steady decline to a comparatively stable low-tariff pattern.

Figure 1: **ChartDiff Dataset Illustration.** The task requires comparing two charts and generating a concise description of their differences. More examples can be found in Appendix A.

the task of cross-chart comparative summarization. Our experiments show that, while these models can achieve comparable performance on classic metrics (Lin, 2004), their performance varies significantly on modern metrics (Fu et al., 2024). Some models struggle with poor reasoning abilities.

Our contributions are in two aspects:

- We introduce the first benchmark ChartDiff on cross-chart comparative summarization, consisting of 8,541 annotated chart pairs with high diversity.
- We evaluate various modern VLMs on cross-chart comparative summarization to illustrate the great progress of new VLMs and provide new insights into the challenges of chart understanding.

We hope this work catalyzes future research on multi-chart reasoning, an ability that is essential for supporting real-world visual analytics workflows yet remains underrepresented in current model capabilities and benchmarks.

## 2 Related Works

### 2.1 Vision-Language Models for Chart Understanding

VLMs have made rapid progress in chart understanding, and existing approaches can be broadly

divided into general-purpose multimodal models and chart-specialized models. General-purpose models such as GPT systems (OpenAI et al., 2024; Singh et al., 2025), Gemini series (Comanici et al., 2025), Qwen3.5 (Qwen Team, 2026), and InternVL (Chen et al., 2024) exhibit strong visual reasoning abilities and can handle a wide range of chart-related tasks without task-specific training. In contrast, chart-specialized models, including ChartLlama (Han et al., 2023), UniChart (Masry et al., 2023), ChartAssistant (Meng et al., 2024), and ChartGemma (Masry et al., 2025b) are typically trained via visual instruction tuning (Liu et al., 2023c) or multitask learning to better capture chart structures and semantics. Some works further extend encoder-decoder frameworks, such as Matcha (Liu et al., 2023b) built upon Pix2Struct (Lee et al., 2023), by incorporating chart data extraction and mathematical reasoning capabilities. While these models achieve strong performance on tasks like chart question answering and summarization, they are often limited by task-specific tuning and predefined pipelines, which restrict their ability to generalize to diverse chart types and complex reasoning scenarios.

Recent work focuses on improving reasoning for chart understanding. TinyChart (Zhang et al., 2024) adopts Program-of-Thought prompting (Chen et al.,

2023), while ChartCoder (Zhao et al., 2025) and ChartReasoner (Jia et al., 2025) leverage chart-to-code generation for multi-step inference. ChartR1 (Chen et al., 2026) and Chart-RL (Zhang et al., 2026) further incorporates reinforcement learning to enhance reasoning quality. However, these approaches depend on intermediate representations such as code or templates, making them sensitive to upstream errors and limiting robustness, as highlighted by works like ChartMimic (Yang et al., 2025) and ChartMoE (Xu et al., 2025).

## 2.2 Benchmarks for Chart Understanding

A wide range of benchmarks have been proposed to evaluate chart understanding capabilities of VLMs. Early datasets such as FigureQA (Kahou et al., 2017), DVQA (Kafle et al., 2018), and PlotQA (Methani et al., 2020) primarily rely on synthetic charts and template-based questions, which constrain both visual diversity and reasoning complexity. Subsequent efforts, including ChartQA (Masry et al., 2022) and Chart-to-Text (Kantharaj et al., 2022), shift toward real-world data and more natural language queries, enabling evaluation of higher-level tasks such as question answering and summarization. More recent benchmarks, such as ChartX (Xia et al., 2024) and ChartBench (Xu et al., 2023), further expand chart diversity and task coverage. Despite these advances, most existing benchmarks are designed around single-chart settings, where models process one visualization at a time without considering relationships across multiple charts.

To address increasing model capabilities, several works introduce more challenging tasks and broader evaluation protocols. Benchmarks like CharXiv (Wang et al., 2024) and SciGraphQA (Li and Tajbakhsh, 2023) emphasize complex reasoning, multi-turn interactions, or domain-specific knowledge, while ChartQAPro (Masry et al., 2025a) and MMMU (Yue et al., 2024) explore more diverse question types and multimodal reasoning scenarios. However, these benchmarks still predominantly evaluate understanding within a single chart, focusing on tasks such as data extraction, description, or localized reasoning. As a result, they fail to capture a critical real-world requirement: comparing and synthesizing information across multiple visualizations, which often involves aligning semantics, identifying differences, and aggregating trends.

Only a limited number of benchmarks begin to

explore multi-chart or cross-chart reasoning. For example, MultiChartQA (Zhu et al., 2025) introduces multi-hop reasoning across related charts, while INTERCHART (Iyengar et al., 2025) investigates cross-chart understanding under controlled settings. ReMI (Kazemi et al., 2024) also includes a small subset of multi-chart scenarios. Nevertheless, these datasets are relatively small in scale or limited in task design, leaving substantial gaps in evaluating comprehensive cross-chart reasoning. To bridge this gap, we propose ChartDiff, a large-scale benchmark for cross-chart comparative summarization, consisting of 8,541 chart pairs. Notably, ChartAB (Bansal et al., 2025) presents a dataset of comparable scale, but focuses on a diagnostic framework centered on grounding, alignment, and robustness, emphasizing fine-grained pairwise difference identification rather than holistic comparative summarization. In contrast, ChartDiff explicitly targets comparative reasoning by requiring models to identify differences in trends, fluctuations, and anomalies across charts and to generate coherent summaries, offering an additional perspective for evaluating chart understanding in multi-chart scenarios.

## 3 Dataset Construction

In this section, we describe the construction pipeline of ChartDiff. We begin by collecting raw data from real-world sources, followed by preprocessing to prepare the data for chart rendering. Finally, we construct chart pairs and apply an annotation pipeline to produce comparison summaries.

### 3.1 Raw Data Collection and Processing

We collect tabular time-series data from publicly available sources, including Macrotrends (Macrotrends LLC, 2026), Yahoo Finance (Yahoo Finance, 2026), and Visual Crossing (Visual Crossing Corporation, 2026). The collected data spans eight domains: economy, health, immigration, labor force, population, trade, stock markets, and weather. In total, the dataset covers approximately 200 countries or regions, 100 cities, and 100 publicly traded stocks.

We first filter out datasets with discontinuous or incomplete time series to ensure data quality. We then perform data sampling and construct pairs of datasets for comparison. To ensure that each pair reflects a meaningful and controlled difference, we constrain paired datasets to differ along only one

of the following three aspects: (1) data entity (e.g., different countries or stocks), (2) time span, or (3) data category.

After pairing, each pair consists of two CSV datasets, which are used to generate a pair of charts for subsequent comparison and annotation.

### 3.2 Chart Rendering

To generate visualizations, we utilize three widely used Python plotting libraries: Matplotlib (Hunter, 2007), Plotly (Plotly Technologies Inc., 2015), and Plotnine (Hassan Kibirige, 2026). For each library, we design multiple styling configurations to enhance visual diversity, resulting in approximately 60 distinct visualization styles in total.

ChartDiff includes six chart categories: line charts, bar charts, horizontal bar charts, multi-series line charts<sup>1</sup>, multi-series bar charts, and pie charts. These chart types cover a wide range of common visualization scenarios, including temporal trends, categorical comparisons, and proportional distributions.

To ensure high-quality visualizations, all generated charts undergo manual inspection. We verify the absence of common visualization issues, including legend occlusion, missing or improperly rendered data points, inconsistent axis scaling, and other artifacts that could hinder accurate interpretation.

### 3.3 Annotation Pipeline

To produce high-quality comparison summaries, we employ a multi-stage annotation pipeline leveraging large language models (LLMs). The pipeline follows an *annotate-judge-verify* paradigm for each chart pair.

First, we define a pool of LLMs  $\mathcal{A}$ . For each chart pair  $P$ , we randomly sample an annotator model  $L_1 \in \mathcal{A}$  and prompt it with a carefully designed instruction (Figure 11) to generate a candidate comparison summary  $S$ . We provide only the underlying CSV data of the charts to ensure accurate and consistent analysis.

Next, we sample a second model  $L_2 \in \mathcal{A} \setminus \{L_1\}$  to act as a judge. Given the same chart pair  $P$  and the candidate summary  $S$ ,  $L_2$  evaluates whether  $S$  is accurate and acceptable according to a predefined

<sup>1</sup>A multi-series chart is defined as a visualization where the underlying CSV data contains multiple distinct value columns. An example pair of multi-series line charts is shown in Figure 6, and an example pair of multi-series bar charts is shown in Figure 7.

Chart Type	Pairs	Percentage
Line	2257	26.4%
Bar	2153	25.2%
Horizontal Bar	679	8.0%
Line (Multi-series)	1044	12.2%
Bar (Multi-series)	1072	12.6%
Pie	1336	15.6%

Table 1: Distribution of chart pairs across different chart types in ChartDiff.

evaluation prompt (Figure 12), and decides whether to accept or reject it. For accepted summaries, the words “Dataset” and “dataset” are replaced with “Chart” and “chart”, respectively.

Finally, all accepted summaries undergo manual verification to ensure quality. During this stage, we check for factual correctness, completeness of key differences, and overall clarity, filtering out any remaining low-quality or ambiguous annotations.

### 3.4 Dataset Curation Details

For data pairing, we sample between 6 and 12 data points for each dataset across all chart types, except for pie charts, and ensure that the two datasets in each pair contain the same number of data points. For pie charts, we restrict each dataset to 3–5 categories to maintain clear and interpretable proportional comparisons.

For chart rendering, we randomly select a visualization style from a set of predefined styling configurations for each pair, and both charts in the pair share the same styling configuration.

In the annotation pipeline, we employ a pool of LLMs, including GPT-5.4 (Singh et al., 2025) and Gemini 3.1 Pro (Google Deepmind, 2026b). The acceptance rate of candidate summaries generated by GPT-5.4 is 0.93, while that of Gemini 3.1 Pro is 0.967, indicating a high level of annotation quality.

After completing the three-stage pipeline, we obtain a total of 8,541 chart pairs with high-quality comparison annotations. The distribution of samples across different chart types is summarized in Table 1.

Each chart is rendered at a resolution of  $800 \times 600$  pixels. We further split ChartDiff into training, validation, and test sets containing 6,041, 1,000, and 1,500 chart pairs, respectively.

## 4 Experiments

### 4.1 Models

We evaluate a diverse set of models spanning four categories:

**General-Purpose Closed-Source Models.** We include state-of-the-art proprietary VLMs, including GPT-5.4 (Singh et al., 2025), Gemini 3.1 Pro (Google Deepmind, 2026b), GPT-5.4-mini (Singh et al., 2025), Gemini 3.1 Flash Lite (Google Deepmind, 2026a), Claude Sonnet 4.6 (Anthropic, 2026), and GPT-4o (OpenAI et al., 2024).

**General-Purpose Open-Source Models.** We consider representative open-source models, including Qwen3.5-397B-A17B (Qwen Team, 2026), Qwen3.5-9B (Qwen Team, 2026), and Qwen2.5-VL-7B (Bai et al., 2025).

**Chart-Domain Specialized Models.** We evaluate models specifically designed for chart understanding, including ChartGemma (Masry et al., 2025b) and MatCha (Liu et al., 2023b).

**Pipeline-Based Methods.** In addition to end-to-end models, we evaluate a pipeline-based approach that first extracts structured data from charts and then performs comparison using an LLM. Specifically, we use DePlot (Liu et al., 2023a) as the chart-to-table extractor, followed by GPT-5.4 (Singh et al., 2025) or Qwen3.5-9B (Qwen Team, 2026) for comparison generation.

Finally, to establish a lower-bound baseline, we prompt GPT-5.4 (Singh et al., 2025) to randomly generated outputs (Figure 15), which serve as a reference for non-informative predictions.

### 4.2 Evaluation Metrics

We adopt two complementary evaluation metrics:

**ROUGE.** We use ROUGE (Lin, 2004) as a standard lexical-overlap metric to measure similarity between generated summaries and reference annotations.

**GPT Score.** We further employ a model-based evaluation metric, GPT Score (Fu et al., 2024), using GPT-5.4 (Singh et al., 2025) as the judge model to assess the quality of generated summaries with a predefined grading prompt (Figure 16 and Figure 17).

To validate the reliability of GPT Score, we randomly sample 300 comparison summaries generated by different models and obtain human ratings using the same evaluation criteria as the grading prompt (Figure 16 and Figure 17). We observe

a Pearson correlation coefficient of 0.91 between human scores and GPT scores, indicating strong agreement.

### 4.3 Implementation Details

For all models, we concatenate the two charts horizontally into a single combined image as the model input.

For general-purpose models, we evaluate performance in a zero-shot setting without task-specific fine-tuning. We directly prompt the models (Figure 13) to generate comparison summaries.

For pipeline-based methods, we first use a pre-trained DePlot (Liu et al., 2023a) model for chart-to-table extraction without additional fine-tuning, and then prompt the selected LLM (Figure 14) with the extracted tables to generate comparison summaries.

For chart-specific models, including ChartGemma (Masry et al., 2025b) and MatCha (Liu et al., 2023b), we fine-tune each model on the ChartDiff training set for five epochs.

### 4.4 Analysis

**Overall.** Table 2 shows that general-purpose closed-source models achieve the best overall generation quality, with GPT-5.4 (Singh et al., 2025) obtaining the highest GPT Score (4.95), followed by Gemini 3.1 Pro (Google Deepmind, 2026b) (4.86). In contrast, chart-domain specialized models and pipeline-based methods achieve the strongest ROUGE scores, with ChartGemma (Masry et al., 2025b) reaching the highest ROUGE-1 (51.49), MatCha (Liu et al., 2023b) the best ROUGE-2 (18.34) and ROUGE-L (28.75), and Deplot (Liu et al., 2023a) + GPT-5.4 also performing strongly on all ROUGE metrics. However, these models obtain much lower GPT Scores, indicating a substantial mismatch between lexical overlap and human-aligned generation quality. Among open-source models, Qwen3.5-397B-A17B (Qwen Team, 2026) is the strongest, achieving competitive ROUGE scores and a GPT Score of 4.54, though it still trails the best proprietary systems. Overall, the results suggest that while specialized and pipeline-based approaches are advantageous for reference matching, frontier general-purpose LLMs produce more natural and better-evaluated chart descriptions, highlighting the importance of complementing lexical overlap-based metrics with quality-oriented evaluation metrics.

Models	ROUGE-1	ROUGE-2	ROUGE-L	GPT Score
<i>General-Purpose Closed-Source Models</i>				
GPT-5.4 (Singh et al., 2025)	46.02	12.28	23.45	<b>4.95</b>
Gemini 3.1 Pro (Google Deepmind, 2026b)	47.21	<b>13.48</b>	<b>24.2</b>	4.86
GPT-5.4-mini (Singh et al., 2025)	43.0	10.62	21.68	4.82
Gemini 3.1 Flash Lite (Google Deepmind, 2026a)	46.37	12.83	22.82	4.63
Claude Sonnet 4.6 (Anthropic, 2026)	<b>47.54</b>	13.31	23.42	4.58
GPT-4o (OpenAI et al., 2024)	44.43	11.48	22.44	4.23
<i>General-Purpose Open-Source Models</i>				
Qwen3.5-397B-A17B (Qwen Team, 2026)	<b>47.07</b>	<b>12.68</b>	<b>22.57</b>	<b>4.54</b>
Qwen3.5-9B (Qwen Team, 2026)	44.09	10.84	21.16	3.65
Qwen2.5VL-7B (Bai et al., 2025)	41.18	9.82	20.88	3.18
<i>Chart-Domain Specialized Models</i>				
ChartGemma (Masry et al., 2025b)	<b>51.49</b>	17.81	28.53	<b>2.0</b>
MatCha (Liu et al., 2023b)	49.52	<b>18.34</b>	<b>28.75</b>	1.45
<i>Pipeline-Based Methods</i>				
Deplot (Liu et al., 2023a) + GPT-5.4	<b>50.75</b>	<b>17.25</b>	<b>28.88</b>	<b>3.58</b>
Deplot + GPT-4o	46.46	13.19	23.66	3.38
Deplot + Qwen3.5-9B	43.1	10.38	20.3	2.81
Random	25.5	2.5	12.81	1.17

Table 2: Performance comparison of all evaluated models. Boldface indicates the best result within each group.

Models	GPT Score						
	Overall	Chart Type					
		Line	Bar	Bar(H.)	Line(M.)	Bar(M.)	Pie
<i>General-Purpose Closed-Source Models</i>							
GPT-5.4 (Singh et al., 2025)	<b>4.95</b>	<b>4.97</b>	<b>4.97</b>	4.89	<b>4.9</b>	<b>4.88</b>	<b>4.99</b>
Gemini 3.1 Pro (Google Deepmind, 2026b)	4.86	4.82	4.9	<b>4.94</b>	4.65	4.85	4.98
GPT-5.4-mini (Singh et al., 2025)	4.82	4.86	4.82	4.75	4.74	4.78	4.85
Gemini 3.1 Flash Lite (Google Deepmind, 2026a)	4.63	4.65	4.66	4.67	4.33	4.47	4.87
Claude Sonnet 4.6 (Anthropic, 2026)	4.58	4.54	4.6	4.57	4.38	4.46	4.87
GPT-4o (OpenAI et al., 2024)	4.23	4.23	4.32	3.85	3.88	3.78	4.85
<i>General-Purpose Open-Source Models</i>							
Qwen3.5-397B-A17B (Qwen Team, 2026)	<b>4.54</b>	<b>4.59</b>	<b>4.58</b>	<b>4.36</b>	<b>4.41</b>	<b>4.35</b>	<b>4.72</b>
Qwen3.5-9B (Qwen Team, 2026)	3.65	3.82	3.89	3.55	3.2	3.33	3.57
Qwen2.5VL-7B (Bai et al., 2025)	3.18	3.54	3.14	2.79	2.79	2.53	3.58
<i>Chart-Domain Specialized Models</i>							
ChartGemma (Masry et al., 2025b)	<b>2.0</b>	<b>2.36</b>	<b>2.36</b>	<b>2.01</b>	<b>1.3</b>	<b>1.36</b>	<b>1.68</b>
MatCha (Liu et al., 2023b)	1.45	1.62	1.69	1.43	0.92	0.99	1.47
<i>Pipeline-Based Methods</i>							
Deplot (Liu et al., 2023a) + GPT-5.4	<b>3.58</b>	<b>3.89</b>	<b>4.63</b>	3.16	<b>2.91</b>	<b>4.65</b>	1.24
Deplot + GPT-4o	3.38	3.68	4.32	<b>3.23</b>	2.77	4.21	<b>1.27</b>
Deplot + Qwen3.5-9B	2.81	3.16	3.8	2.16	2.3	3.44	0.79
Random	1.17	1.23	1.5	1.52	1.28	1.24	0.21

Table 3: GPT Score results by chart type for all evaluated models. Bar(H.) indicates horizontal bar charts. Line(M.) indicates multi-series line charts. Bar(M.) indicates multi-series bar charts. Boldface indicates the best result within each model category.

Models	Overall	Plotting Libraries		
		Matplotlib	Plotly	Plotnine
GPT-5.4 (Singh et al., 2025)	4.95	4.94	4.97	4.93
Qwen3.5-397B-A17B (Qwen Team, 2026)	4.54	4.51	4.63	4.44
ChartGemma (Masry et al., 2025b)	2.0	2.06	1.99	1.95
Deplot (Liu et al., 2023a)+ GPT-5.4	3.58	4.08	3.12	3.89

Table 4: GPT Score of representative models across different plotting libraries.

**Chart type.** Table 3 shows clear performance differences across chart types. Pie charts are generally the easiest for general-purpose LLMs, with nearly all closed-source models achieving very high GPT Scores on this type. Simple line and bar charts are also handled well, while multi-series charts are more challenging, as performance tends to drop across most model families. This pattern is particularly evident for smaller open-source and chart-specialized models, whose scores decrease substantially on multi-series charts. In contrast, pipeline-based methods perform relatively well on bar and multi-series bar charts but fail badly on pie charts, suggesting that their effectiveness is highly dependent on chart structure, possibly because DePlot was not pretrained on pie charts. Overall, the results indicate that chart complexity is a major factor in chart-to-text generation, with multi-series charts remaining the most difficult cases.

**Plotting library.** Table 4 shows that performance is generally stable across plotting libraries for strong end-to-end models, suggesting that library-specific rendering differences have limited impact on top-performing LLMs. For example, GPT-5.4 (Singh et al., 2025) maintains nearly identical GPT Scores on Matplotlib (4.94), Plotly (4.97), and Plotnine (4.93), while Qwen3.5-397B-A17B also shows only moderate variation across libraries. Among the three libraries, Plotly appears slightly easier for general-purpose models, yielding the highest scores for both GPT-5.4 and Qwen3.5-397B-A17B (Qwen Team, 2026). In contrast, pipeline-based methods are more sensitive to library choice: Deplot (Liu et al., 2023a) + GPT-5.4 performs best on Matplotlib (4.08) and Plotnine (3.89) but drops noticeably on Plotly (3.12). Overall, the results suggest that plotting library has only a minor effect for strong end-to-end LLMs, but can substantially affect modular pipeline approaches.

## 5 Conclusion

We present ChartDiff, the first large-scale benchmark for cross-chart comparative summarization, consisting of 8,541 chart pairs with high-quality comparison annotations. Our evaluation shows that although modern vision-language models have become strong at single-chart understanding, they still face clear challenges in comparative chart reasoning, especially on structurally complex chart types such as multi-series charts. We further find a substantial mismatch between lexical-overlap met-

rics and GPT-based quality evaluation, suggesting that chart comparison should be assessed with more than overlap-based metrics alone. We hope ChartDiff will provide a useful foundation for future research on multi-chart understanding and comparative visual reasoning.

## Limitations

Our work has several limitations. ChartDiff covers only a subset of common chart types and may not generalize to more complex real-world visualizations. Although annotations are human-verified, they are partially LLM-generated and may reflect annotation bias. In addition, our main evaluation relies on GPT-Score, which, despite strong correlation with human judgment, is still an imperfect automatic evaluator. Finally, we focus on open-ended comparative summarization rather than other multi-chart reasoning tasks, leaving broader comparative understanding for future work.

## Ethics Statement

ChartDiff is constructed from publicly available data and synthetically generated charts and does not contain personal or sensitive information. The dataset is intended for research purposes only. We note that models evaluated on this benchmark may generate incorrect or misleading summaries, and therefore should not be used in high-stakes applications without human verification. We encourage responsible use of the dataset and future work on improving the reliability and safety of chart understanding systems.

## Acknowledgements

The author thanks the anonymous reviewers for their careful reading and constructive feedback.

The author is deeply grateful to his friends FY, Dingyuan, and Xinchu for their unwavering support throughout this research. Their constant encouragement, together with their equally constant insults, proved invaluable in sustaining this work.

The author also acknowledges the computational resources provided by CSC – IT Center for Science, Finland.

## References

Anthropic. 2026. *Claude sonnet 4.6*. Accessed: 2026-03-18.

- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. **Qwen2.5-vl technical report**. *ArXiv*, abs/2502.13923.
- Aniruddh Bansal, Davit Soselia, Dang Nguyen, and Tianyi Zhou. 2025. **Chartab: A benchmark for chart grounding & dense alignment**. *Preprint*, arXiv:2510.26781.
- Lei Chen, Xuanle Zhao, Zhixiong Zeng, Jing Huang, Yufeng Zhong, and Lin Ma. 2026. **Chart-r1: Chain-of-thought supervision and reinforcement for advanced chart reasoner**. *Preprint*, arXiv:2507.15509.
- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.
- Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsen, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. **Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities**. *Preprint*, arXiv:2507.06261.
- Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. **GPTScore: Evaluate as you desire**. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.
- Google Deepmind. 2026a. **Gemini 3.1 flash lite**. Accessed: 2026-03-18.
- Google Deepmind. 2026b. **Gemini 3.1 pro**. Accessed: 2026-03-18.
- Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. 2023. **Chartllama: A multimodal llm for chart understanding and generation**. *Preprint*, arXiv:2311.16483.
- Hassan Kibirige. 2026. **plotnine**. Accessed: 2026-08.
- J. D. Hunter. 2007. **Matplotlib: A 2d graphics environment**. *Computing in Science & Engineering*, 9(3):90–95.
- Anirudh Iyengar Kaniyar Narayana Iyengar, Srija Mukhopadhyay, Adnan Qidwai, Shubhankar Singh, Dan Roth, and Vivek Gupta. 2025. **INTERCHART: Benchmarking visual reasoning across decomposed and distributed chart information**. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2046–2067, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.
- Caijun Jia, Nan Xu, Jingxuan Wei, Qingli Wang, Lei Wang, Bihui Yu, and Junnan Zhu. 2025. **Chartreasoner: Code-driven modality bridging for long-chain reasoning in chart question answering**. *Preprint*, arXiv:2506.10116.
- Kushal Kafle, Scott Cohen, Brian Price, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *CVPR*.
- Samira Ebrahimi Kahou, Adam Atkinson, Vincent Michalski, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. **Figureqa: An annotated figure dataset for visual reasoning**. *ArXiv*, abs/1710.07300.
- Shankar Kantharaj, Rixie Tiffany Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. **Chart-to-text: A large-scale benchmark for chart summarization**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4005–4023, Dublin, Ireland. Association for Computational Linguistics.
- Mehran Kazemi, Nishanth Dikkala, Ankit Anand, Petar Devic, Ishita Dasgupta, Fangyu Liu, Bahare Fatemi, Pranjal Awasthi, Dee Guo, Sreenivas Gollapudi, and Ahmed Qureshi. 2024. Remi: a dataset for reasoning with multiple images. In *Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS '24*, Red Hook, NY, USA. Curran Associates Inc.
- Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning, ICML'23*. JMLR.org.

- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *Proceedings of Machine Learning Research*, 162:12888–12900. Publisher Copyright: Copyright © 2022 by the author(s); 39th International Conference on Machine Learning, ICML 2022 ; Conference date: 17-07-2022 Through 23-07-2022.
- Shengzhi Li and Nima Tajbakhsh. 2023. Scigraphqa: A large-scale synthetic multi-turn question-answering dataset for scientific graphs. *CoRR*, abs/2308.03349.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhui Chen, Nigel Collier, and Yasemin Altun. 2023a. DePlot: One-shot visual language reasoning by plot-to-table translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10381–10399, Toronto, Canada. Association for Computational Linguistics.
- Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. 2023b. MatCha: Enhancing visual language pretraining with math reasoning and chart derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12756–12770, Toronto, Canada. Association for Computational Linguistics.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023c. Visual instruction tuning. In *NeurIPS*.
- Macrotrends LLC. 2026. [Macrotrends data downloads](#). Accessed: 2025-08.
- Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehdi Shahmohammadi, Megh Thakkar, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2025a. ChartQAPro: A more diverse and challenging benchmark for chart question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19123–19151, Vienna, Austria. Association for Computational Linguistics.
- Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. 2023. UniChart: A universal vision-language pretrained model for chart comprehension and reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14662–14684, Singapore. Association for Computational Linguistics.
- Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.
- Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. 2025b. ChartGemma: Visual instruction-tuning for chart reasoning in the wild. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 625–643, Abu Dhabi, UAE. Association for Computational Linguistics.
- Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. ChartAssistant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7775–7803, Bangkok, Thailand. Association for Computational Linguistics.
- Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*.
- OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- Plotly Technologies Inc. 2015. [Collaborative data science](#).
- Qwen Team. 2026. [Qwen3.5: Towards native multimodal agents](#).
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). *Preprint*, arXiv:2103.00020.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 465 others. 2025. [Openai gpt-5 system card](#). *Preprint*, arXiv:2601.03267.
- Visual Crossing Corporation. 2026. [Visual crossing weather \(2021–2023\)](#). <https://www.visualcrossing.com/>. [data service] Accessed: 2025-08.
- Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, Alexis Chevalier, Sanjeev Arora, and Danqi Chen. 2024. [Charxiv: Charting](#)

- gaps in realistic chart understanding in multimodal llms. In *Advances in Neural Information Processing Systems*, volume 37, pages 113569–113697. Curran Associates, Inc.
- Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Min Dou, Botian Shi, Junchi Yan, and Yu Qiao. 2024. *Chartx and chartvlm: A versatile benchmark and foundation model for complicated chart reasoning*. *IEEE Transactions on Image Processing*, 34:7436–7447.
- Zhengzhuo Xu, Sinan Du, Yiyan Qi, Chengjin Xu, Chun Yuan, and Jian Guo. 2023. *Chartbench: A benchmark for complex visual reasoning in charts*. *CoRR*, abs/2312.15915.
- Zhengzhuo Xu, Bowen Qu, Yiyan Qi, Sinan Du, Chengjin Xu, Chun Yuan, and Jian Guo. 2025. *Chartmoe: Mixture of diversely aligned expert connector for chart understanding*. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Yahoo Finance. 2026. Historical data for 100 stocks. <https://finance.yahoo.com>. Accessed: 2025-08.
- Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, Gongye Liu, Xiaomei Nie, Deng Cai, and Yujiu Yang. 2025. *Chartmimic: Evaluating Imm’s cross-modal reasoning capability via chart-to-code generation*. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. *Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi*. In *Proceedings of CVPR*.
- Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. 2024. *TinyChart: Efficient chart understanding with program-of-thoughts learning and visual token merging*. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1898, Miami, Florida, USA. Association for Computational Linguistics.
- Xin Zhang, Xingyu Li, Rongguang Wang, Ruizhong Miao, Zheng Wang, Dan Roth, and Chenyang Li. 2026. *Chart-rl: Generalized chart comprehension via reinforcement learning with verifiable rewards*. *Preprint*, arXiv:2603.06958.
- Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2025. *ChartCoder: Advancing multimodal large language model for chart-to-code generation*. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7333–7348, Vienna, Austria. Association for Computational Linguistics.
- Zifeng Zhu, Mengzhao Jia, Zhihan Zhang, Lang Li, and Meng Jiang. 2025. *MultiChartQA: Benchmarking vision-language models on multi-chart problems*. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11341–11359, Albuquerque, New Mexico. Association for Computational Linguistics.

# A Dataset Examples

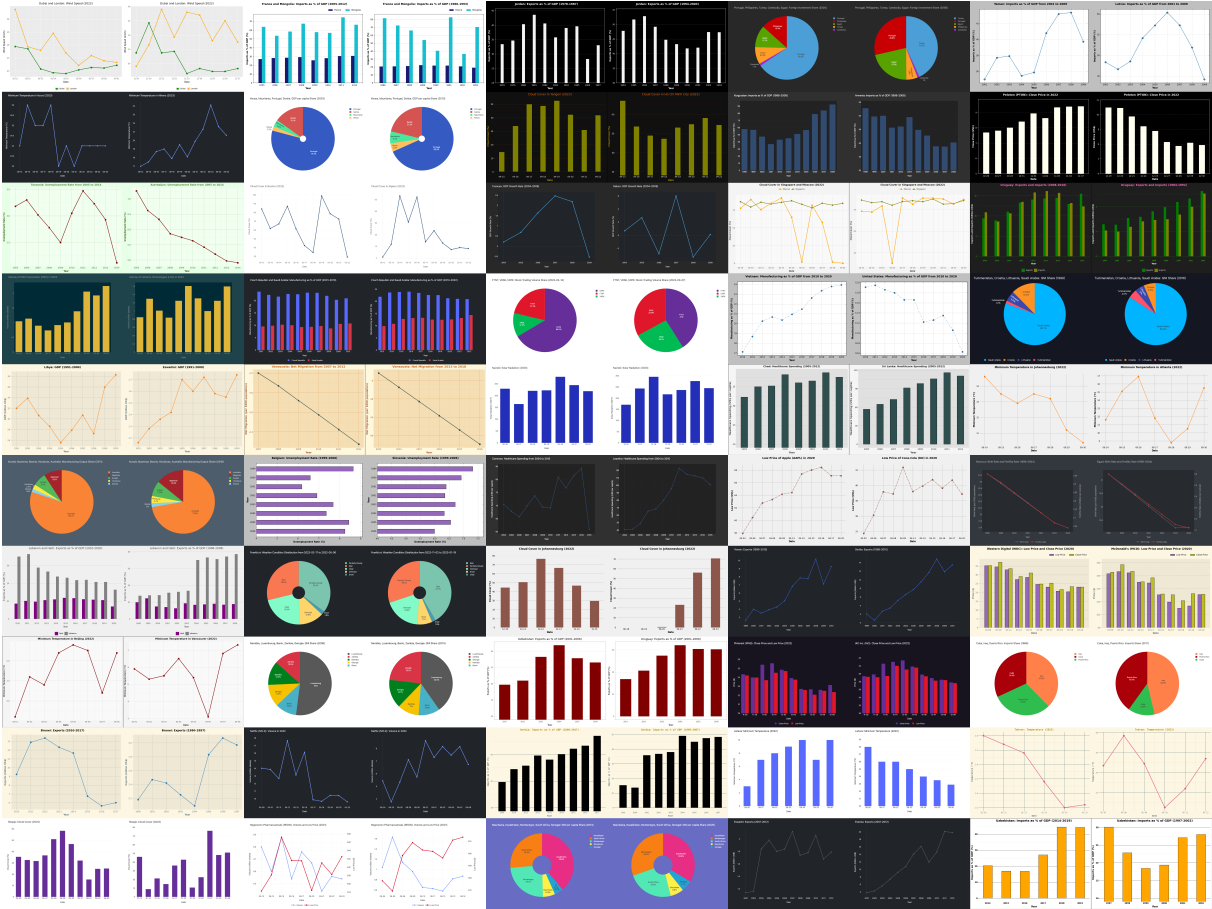
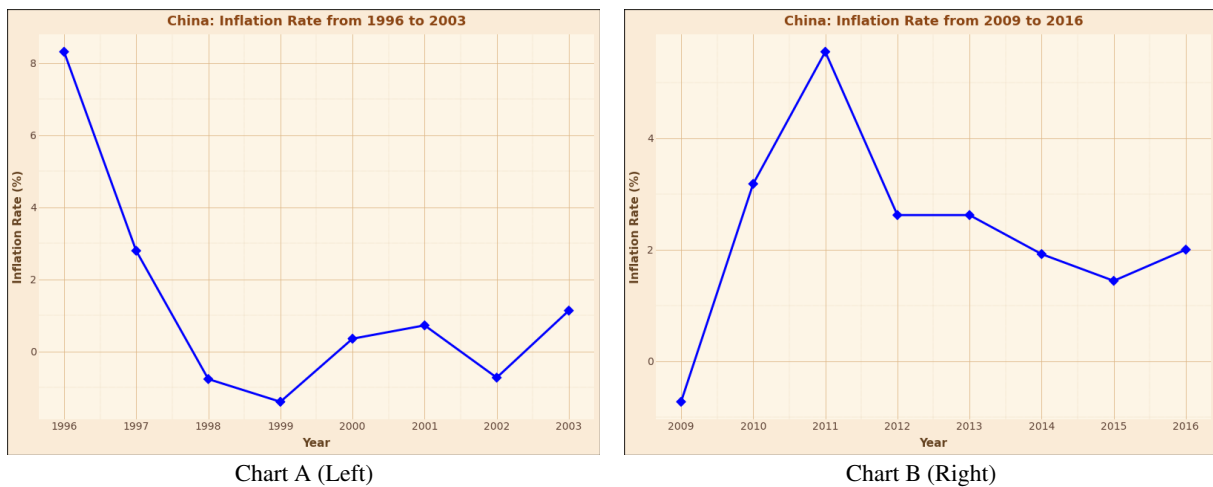
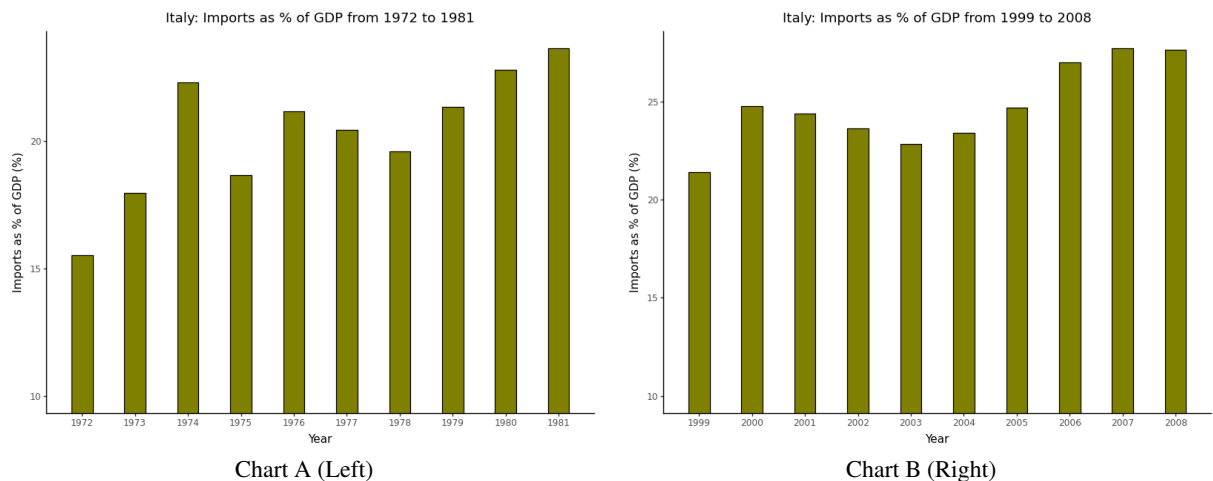


Figure 2: Fifty randomly selected chart pairs from the ChartDiff dataset.



**Comparison Summary:** A comparison of China’s inflation rates between 1996–2003 and 2009–2016 reveals distinct differences in economic volatility and overall trends. The 1996–2003 period was characterized by a dramatic plunge from an initial high of 8.31% to recurring deflationary episodes, hitting a low of -1.4% in 1999 and remaining near or below zero for most of the timeframe. In contrast, the 2009–2016 period began with slight deflation at -0.73% but quickly spiked to a peak of 5.55% in 2011 before gradually cooling and stabilizing around 2%. Ultimately, while the earlier dataset highlights a persistent downward trajectory into multiple years of negative inflation, the later dataset demonstrates a rapid post-deflation recovery followed by sustained, moderate positive inflation.

Figure 3: An example pair of line charts.



**Comparison Summary:** A comparison of Italy’s imports as a percentage of GDP between the 1972–1981 and 1999–2008 periods reveals a substantially higher baseline for imports in the later decade. During the 1970s, the import share started at a low of 15.51% and experienced significant volatility, notably spiking to 22.28% in 1974 before dropping sharply to 18.67% the following year. Conversely, the 1999–2008 period operated at an elevated level, beginning at 21.42% and demonstrating a much smoother overall growth trajectory. This later decade saw steady annual increases from 2003 onward, peaking at 27.70% in 2007 before slightly plateauing in 2008. Despite the differences in volatility and baseline values, both datasets ultimately share a consistent long-term upward trend in Italy’s import-to-GDP ratio over their respective ten-year spans.

Figure 4: An example pair of bar charts.

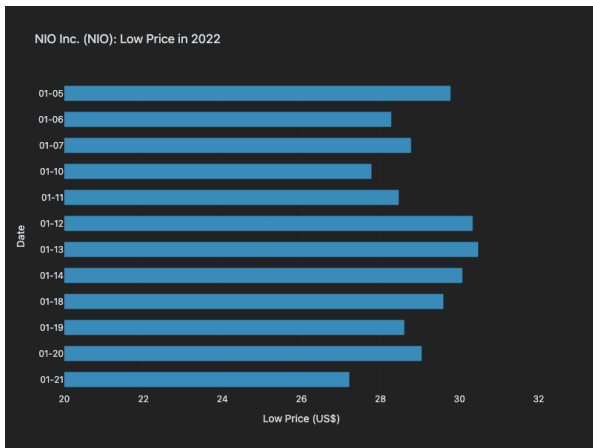


Chart A (Left)

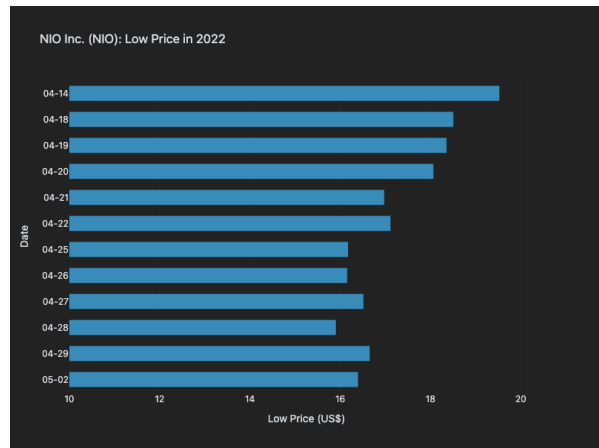


Chart B (Right)

**Comparison Summary:** A comparison of NIO Inc.’s daily low stock prices in 2022 reveals a significant depreciation in the stock’s value between January and April. In January, the low prices fluctuated at a relatively high baseline, ranging from \$27.22 to a peak of \$30.48 mid-month. By contrast, the late April to early May period shows that prices had plummeted to a much lower bracket of \$15.91 to \$19.53. Furthermore, while the January data displayed mild volatility centered around the \$28 to \$30 mark, the April data experienced a consistent downward trajectory that bottomed out at \$15.91 on April 28. Overall, the datasets highlight a drastic downward shift in NIO’s market valuation over the first four months of the year.

Figure 5: An example pair of horizontal bar charts.

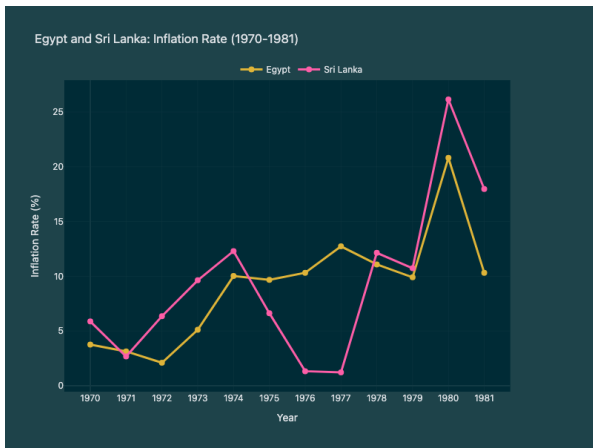


Chart A (Left)

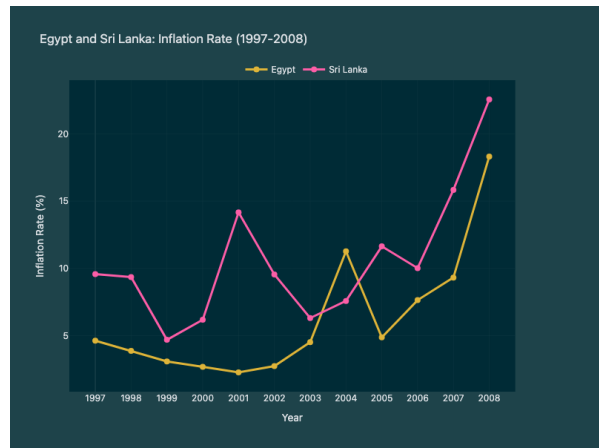


Chart B (Right)

**Comparison Summary:** Across both periods, Egypt and Sri Lanka show generally moderate inflation punctuated by occasional spikes, but the earlier period is more volatile overall, especially for Sri Lanka. From 1970–1981, Egypt rises from low single digits to a late surge in 1980 before easing in 1981, while Sri Lanka swings much more sharply, dropping to near 1% in 1976–1977 before jumping above 12% in 1978 and peaking at 26.15% in 1980. In 1997–2008, Egypt is comparatively stable at low inflation through the late 1990s and early 2000s, then trends upward after 2003 to 18.32% in 2008, whereas Sri Lanka remains consistently higher than Egypt in most years and shows repeated double-digit readings, notably in 2001, 2005, 2007, and 2008. A key difference is that the first period features deeper troughs and more abrupt reversals, especially for Sri Lanka, while the second period shows a clearer upward inflation trend for both countries toward 2008. Overall, Sri Lanka tends to have higher and more erratic inflation than Egypt in both datasets, with both countries ending each period with elevated inflation.

Figure 6: An example pair of multi-series line charts.

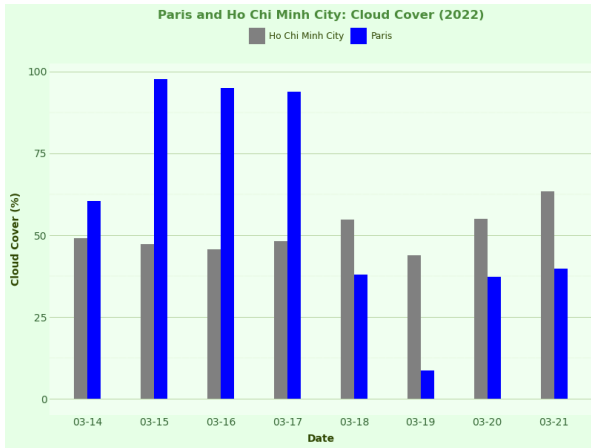


Chart A (Left)

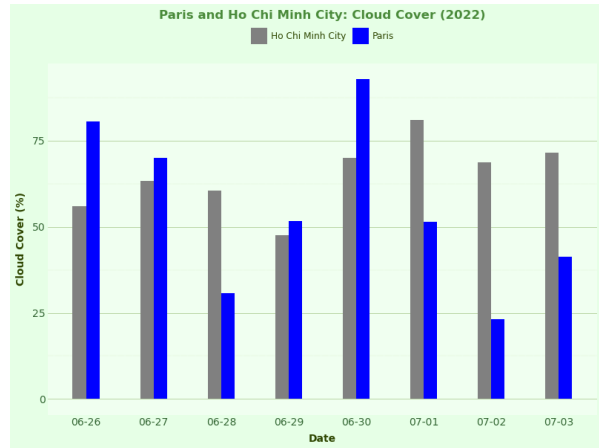


Chart B (Right)

**Comparison Summary:** A comparison of cloud cover between Paris and Ho Chi Minh City across two periods in 2022 reveals distinct volatility patterns and seasonal shifts. In mid-March, Paris experienced extreme fluctuations, with cloud cover plummeting from a near-overcast peak of 97.6% to a low of 8.8%, while Ho Chi Minh City remained relatively stable between 43.9% and 63.4%. Conversely, during the late June to early July period, Ho Chi Minh City experienced an overall increase in cloudiness, trending upward to a peak of 81.0% on July 1. Paris continued to show high atmospheric volatility during this summer timeframe, oscillating wildly between a low of 23.2% and a high of 92.8%. Overall, while Paris consistently demonstrated erratic, sharp shifts in both timeframes, Ho Chi Minh City transitioned from moderate, consistent cloud cover in the spring to much cloudier conditions in the summer.

Figure 7: An example pair of multi-series bar charts.

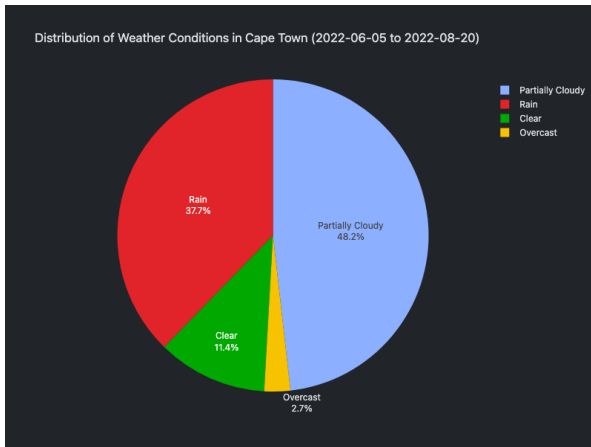


Chart A (Left)

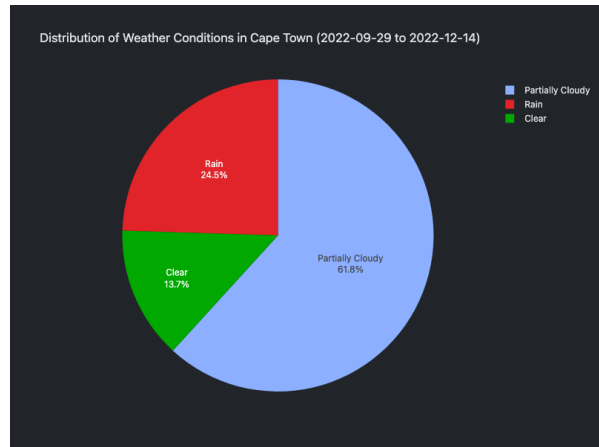


Chart B (Right)

**Comparison Summary:** A comparison of Cape Town's weather distributions reveals a clear transition toward drier and brighter conditions between the June-August 2022 and September-December 2022 periods. The most notable shift is the significant decline in rainy days, which dropped from 37.7% in the earlier period to 24.5% in the later months. Concurrently, partially cloudy conditions surged to become even more dominant, increasing from a 48.2% share to 61.8%. Clear days also experienced a slight uptick, rising from 11.4% to 13.7%. Additionally, while overcast conditions made up 2.7% of the weather in the first period, they disappeared entirely from the distribution during the latter timeframe.

Figure 8: An example pair of pie charts.



Chart A (Left)

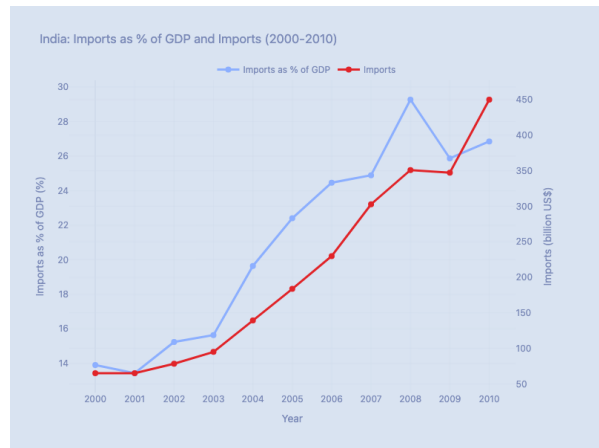


Chart B (Right)

**Comparison Summary:** Between 1980 and 1990, India’s imports were relatively stagnant, with absolute values growing modestly from \$17.23 billion to \$27.13 billion and the import share of GDP dipping mid-decade before recovering to 8.45%. In stark contrast, the 2000 to 2010 period exhibited explosive growth, with absolute imports surging from \$65.12 billion to nearly \$450 billion. Similarly, India’s imports as a percentage of GDP nearly doubled during the 2000s, climbing rapidly from 13.9% in 2000 to a peak of 29.27% in 2008. While the 1980s data shows mostly flat trends with only mild late-decade growth, the 2000s dataset reveals a powerful upward trajectory that was only briefly interrupted by a minor contraction in both metrics in 2009. Ultimately, the comparison illustrates a massive shift in economic scale, highlighting India’s drastically increased integration into global trade during the 2000s compared to its low import reliance in the 1980s.

Figure 9: An example pair of multi-series line charts.

Comparison of Stock Trading Volume Shares on 2021-12-08: JNJ, LMT, LULU, ZM, BMY

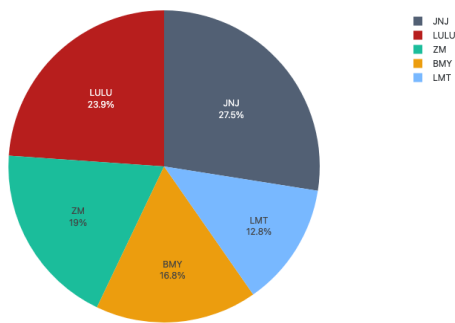


Chart A (Left)

Comparison of Stock Trading Volume Shares on 2020-10-07: JNJ, LMT, LULU, ZM, BMY

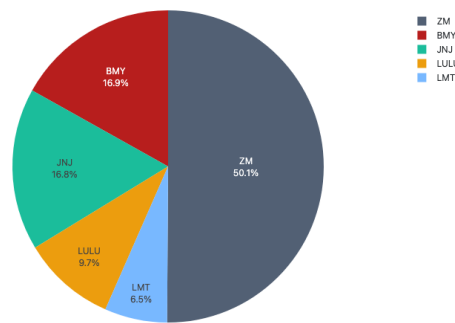


Chart B (Right)

**Comparison Summary:** A comparison of stock trading volume shares between October 7, 2020, and December 8, 2021, reveals a dramatic shift in market focus among the five observed entities. Most notably, ZM dominated the trading volume in 2020 with a massive 50.1% share but experienced a steep decline to just 19.0% by late 2021. Conversely, JNJ and LULU absorbed much of this redistributed volume, with JNJ’s share surging from 16.8% to a leading 27.5%, and LULU’s more than doubling from 9.7% to 23.9%. LMT also saw a notable increase, nearly doubling its share from 6.5% to 12.8% over the same period. Meanwhile, BMY was the only remarkably stable stock in the group, maintaining a consistent share of roughly 16.8% across both dates.

Figure 10: An example pair of pie charts.

## B Prompt templates

### Prompt Template for Generating Candidate Annotations

You are a professional data analyst.

You will compare two small datasets (CSV format) that describe one of the following:

1. Data of the same entity across two time ranges
2. Data of two entities across the same time range
3. Data of two entities across two time ranges
4. Multiseries data of the same entity across two time ranges
5. Multiseries data of two entities across the same time range
6. Comparison of multiple entities' shares across two time ranges
7. Comparison of two entities' shares across the same time range
8. Comparison of the same entity's shares across two time ranges

Your task is to identify the main differences between the datasets in terms of trends, fluctuations, or anomalies.

Your response should be concise, accurate, and informative.

Dataset A:

<CSV\_A>

Dataset B:

<CSV\_B>

Write your comparison as a single cohesive paragraph of no more than five sentences. Do not use bullet points or lists.

Figure 11: Prompt template for generating candidate annotations.

## Prompt Template for Judging Candidate Annotations

You are an expert evaluator for chart-comparison annotations.

You will receive:

1. Dataset A in CSV format
2. Dataset B in CSV format
3. A candidate comparison summary

Your task is to decide whether the candidate summary should be accepted as a valid annotation. Judge the summary **ONLY** against the CSV data.

Accept the summary only if:

- it is factually supported by the data
- it captures the main differences between the datasets
- it does not omit the dominant trend, anomaly, ranking change, or share change
- it is clear and specific rather than generic

Reject the summary if:

- it contains any material factual error
- it confuses Dataset A and Dataset B
- it invents unsupported claims
- it is too vague to be useful
- it misses an important difference shown in the data

Material factual errors include:

- wrong trend direction
- wrong relative comparison
- wrong anomaly/peak/trough/crossover
- wrong share or ranking statement
- incorrect entity/category/time interpretation

Return only one word:

ACCEPT

or

REJECT

Dataset A:

<CSV\_A>

Dataset B:

<CSV\_B>

Candidate summary:

<CANDIDATE\_SUMMARY>

Figure 12: Prompt template for judging candidate annotations.

### Prompt Template for Generating Comparison Summaries

You are a professional data analyst.

Compare Chart A (left) and Chart B (right) using only the information visible in the charts.

Focusing on how their data differs in terms of overall trends, fluctuations, and any notable anomalies and emphasize the most important contrasts between the two charts rather than describing each chart independently.

Your analysis must be concise, accurate, and written as a single cohesive paragraph of no more than five sentences.

Avoid bullet points, lists, or redundant phrasing.

<PAIR\_IMAGE>

Figure 13: Prompt template for generating comparison summaries.

### Prompt Template for Generating Comparison Summaries in Pipeline Methods

You are a professional data analyst.

Compare table A and table B using only the information in the tables.

Focusing on how their data differs in terms of overall trends, fluctuations, and any notable anomalies and emphasize the most important contrasts between the two tables rather than describing each table independently.

Your analysis must be concise, accurate, and written as a single cohesive paragraph of no more than five sentences.

Avoid bullet points, lists, or redundant phrasing.

Table A:

<TABLE\_A>

Table B:

<TABLE\_B>

Figure 14: Prompt template for generating comparison summaries in pipeline methods.

### Prompt Template for Generating LLM Random Guesses

You are a professional data analyst.

You will compare two small datasets (CSV format) that describe one of the following:

1. Data of the same entity across two time ranges
2. Data of two entities across the same time range
3. Data of two entities across two time ranges
4. Multiseries data of the same entity across two time ranges
5. Multiseries data of two entities across the same time range
6. Comparison of multiple entities' shares across two time ranges
7. Comparison of two entities' shares across the same time range
8. Comparison of the same entity's shares across two time ranges

Your task is to identify the main differences between the datasets in terms of trends, fluctuations, or anomalies.

Your response should be concise, accurate, and informative.

Randomly guess a reasonable comparison based on the above instruction only as a single cohesive paragraph of no more than five sentences.

Directly write the comparison as if you access two small datasets.

Do not use bullet points or lists.

Figure 15: Prompt template for generating random guesses from an LLM.

## Prompt Template for Generating GPT Score - Part 1

You are an expert data analyst and evaluator.

You will receive:

1. Dataset A (CSV format), corresponding to Chart A (the left chart)
2. Dataset B (CSV format), corresponding to Chart B (the right chart)
3. A reference analysis (intended correct comparison)
4. A candidate analysis (to be evaluated)

Both analyses describe the differences between two charts derived from the datasets.

Your task is to evaluate the quality of the candidate analysis.

### IMPORTANT PRINCIPLES:

- The datasets are the ultimate source of truth.
- The reference analysis is a guideline for expected coverage and importance, but it may contain minor imperfections.
- Do NOT reward surface similarity to the reference if the content is incorrect.
- Do NOT penalize the candidate for wording differences if the meaning is correct.

Evaluation Procedure (follow internally, do not output):

1. First, analyze Dataset A and Dataset B to identify the key differences:
  - overall trends (increasing, decreasing, stable)
  - fluctuations (volatility, variability)
  - notable anomalies (peaks, drops, outliers)
  - major contrasts between the two datasets
2. Check whether the reference analysis correctly reflects these differences.
  - If the reference is partially incorrect, rely on the data instead.
3. Evaluate the candidate analysis based on:
  - (a) Accuracy
    - Are the statements factually consistent with the datasets?
    - Any contradictions or incorrect claims should be heavily penalized.
  - (b) Completeness
    - Does the candidate cover the main differences identified from the data?
    - Missing minor details is acceptable, but missing key trends is not.
  - (c) Faithfulness
    - Does the candidate avoid hallucinating patterns not supported by the data?
  - (d) Clarity
    - Is the analysis coherent, concise, and easy to understand?

(Continued in Figure 17)

Figure 16: Prompt template (Part 1) for generating GPT Score.

## Prompt Template for Generating GPT Score - Part 2

(Continued from Figure 16)

Scoring:

Give a single integer score from 0 to 5:

- 5 = Excellent: Factually correct, captures all key differences, clear and concise
- 4 = Good: Mostly correct, minor omissions or small inaccuracies
- 3 = Fair: Partially correct, noticeable gaps or some incorrect statements
- 2 = Poor: Major errors or missing important trends
- 1 = Very poor: Mostly incorrect or largely irrelevant
- 0 = Fail: Completely incorrect, nonsensical, or empty

Scoring Rules:

- Prioritize factual accuracy over similarity to the reference.
- If the candidate contradicts the data, score must be  $\leq 2$ .
- If the candidate misses the main trend, score must be  $\leq 3$ .
- Minor wording or structure issues should NOT significantly reduce the score.

Output Format:

Return ONLY a single integer (0-5).

Do NOT provide any explanation or additional text.

Dataset A (Chart A - LEFT):

<CSV\_A>

Dataset B (Chart B - RIGHT):

<CSV\_B>

Reference analysis:

<GROUND\_TRUTH\_ANNOTATION>

Candidate analysis:

<CANDIDATE\_ANALYSIS>

Figure 17: Prompt template (Part 2) for generating GPT Score.

# Formal Machine Interpretation for the Semasiographic Mixtec Codices of Precolonial and Early Colonial Mesoamerica

Christopher Driggers-Ellis  
University of Florida  
driggersellis.cw@ufl.edu

Gabriel Ayoubi  
University of Florida  
gabriel.ayoubi@ufl.edu

Girish Salunke  
Independent  
girish.salunke811@gmail.com

Christan Grant  
University of Florida  
christan@ufl.edu

## Abstract

The precolonial and early colonial Mixtec codices describe the history and stories of the region in a semasiographic medium that is full of symbolic representations and meant to be narrated. Recently, the community has introduced datasets of XML representations of related media, including Aztec codices and Mayan hieroglyphic script, in a step towards symbolic machine interpretation of these historic Mesoamerican artifacts. In this work, we propose formal symbolic machine interpretation of XML encodings representing facsimile images from the Mixtec Codex Zouche-Nuttall. We demonstrate the efficacy of symbolic machine interpretation from XML step-by-step, showing how our parser and interpreter process text capturing a scene from the Mixtec Codex Zouche-Nuttall. We hope our contribution and the example we provide motivate collaboration among the archaeological, historical, linguistic, and natural language processing research communities to apply machine interpretation to Mixtec codices and similar manuscripts.

## 1 Introduction

The Mixtec codices are pictographic manuscripts left by the Mixtec peoples of precolonial and early colonial Mesoamerica and are an example of semasiographic media. Modern examples of semasiographic media include comics, manga, many street signs, storyboards, etc. Like comics or motion picture storyboards (Boone, 2000; Byland and Pohl, 1994), Mixtec Codices use human glyphs interacting with objects and each other, mediated by artistic conventions, to convey legends and histories about ancestors and/or to record those ancestors' genealogies (Williams, 2013; Boone, 2000). A long body of historical and linguistic literature explores human interpretation of Mixtec Codices (Williams, 2013), but there is little or no work investigating machine interpretation of the codices. Our primary contribution is therefore a recursive

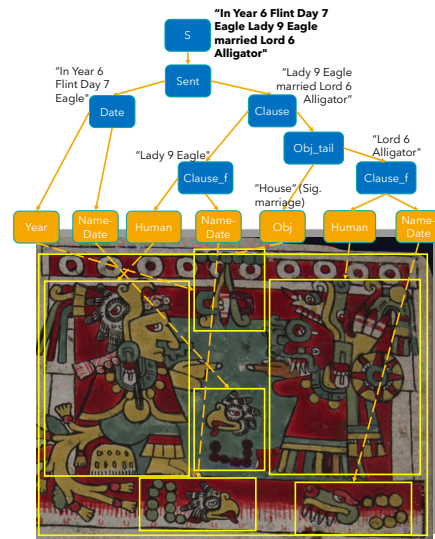


Figure 1: The abstract syntax tree (AST) for the sentence “In Year 6 Flint Day 7 Eagle Lady 9 Eagle married Lord 6 Alligator” from Page 26 of the Codex Zouche-Nuttall (obverse) (Provided courtesy of the British Museum). **Nonterminal** tree nodes are in **Blue**. **Tokens** corresponding to the glyphs are in **Orange**.

descent parser and interpreter for XML representations of Mixtec scenes, and we will discuss this contribution in the context of a broader machine interpretation pipeline to drive discussions about future work necessary to automatically generate this XML representation.

Figure 1 provides an example of the parser at work. The semasiographic interpretation pipeline is elaborated in Figure 2, and the grammar is provided in Figure 6. Our code is available online.<sup>1</sup>

## 2 Related Work

For more than a century, interpretations of the pictographic codices left behind by the indigenous peoples of Mexico, the Mixtec, have been offered in the academic literature (Williams, 2013). In

<sup>1</sup><https://github.com/ufdatastudio/mixtec-parser>

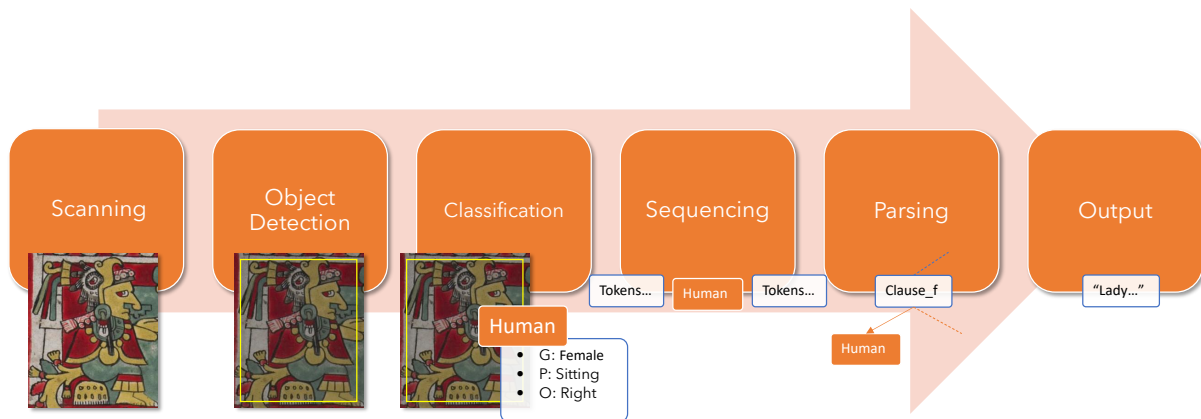


Figure 2: Semasiographic machine interpretation pipeline processing a single human glyph. (Example provided courtesy of the British Museum.) *Scanning* receives the input image. *Object Detection* detects the human glyph. *Classification* extracts its features and assigns them to a human token. *Sequencing* places the token in sequence with other tokens. *Parsing* places it into an AST. Finally, *Output* says “Lady” corresponding to the glyph’s gender.

this work, we focus on an example of pictographic Mixtec writing from Page 26 of the Codex Zouche-Nuttall (CZN) (obverse) which was first published in facsimile by Zelia Nuttall and the Peabody Museum at Harvard University in 1902 (Williams, 2013; Nuttall, 1902). Williams (2013) dates the writing of this manuscript to the 14th or 15th centuries. Since its initial publication for the modern audience more than 120 years ago, the CZN has been reproduced, analyzed, and interpreted by numerous authors (Long, 1926; Caso, 1977; Troike, 1978; Boone, 2000; Williams, 2013). Despite this rich history of human interpretation and recent advancements in computer vision and machine learning technologies, machine interpretation of Mixtec codices like the CZN is not well studied.

Related works in the current century take steps toward machine interpretation for similar Mesoamerican manuscripts. Perri et al. (2024) explore encoding and interpreting the nested features corresponding to agglutinative linguistic elements present in a folio of the Aztec Codex Mendoza (Perri et al., 2024; Fedorova et al., 2024). Other research develops an XML encoding schema for the hieroglyphic script of the Maya (de la Iglesia et al., 2021). We are therefore inspired to pursue symbolic interpretation of the Codex Zouche Nuttall through XML representation. However, unlike the agglutinative Codex Mendoza or Mayan hieroglyphic scripts in these examples, Mixtec codices are pictographic manuscripts that do not correspond systematically to specific linguistic content. Instead, they are read semasiographically like a sto-

ryboard or comic book (Boone, 2000; Byland and Pohl, 1994).

We are most interested in three glyph types from Mixtec codices: human glyphs, name-dates, and toponyms. Human glyphs represent persons, normal human beings or deities, in the Mixtec Codices; name-dates correspond to calendar dates or names of characters; and toponyms convey setting information (Boone, 2000; Williams, 2013).

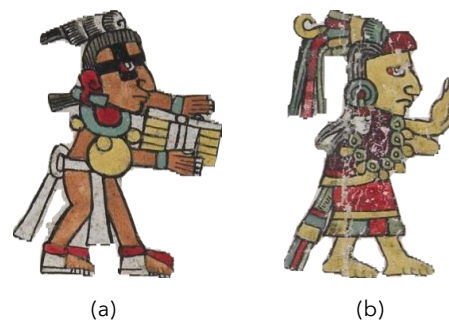


Figure 3: **Human Glyphs:** male Lord (a) and female Lady (b) human glyphs from the CZN (Provided courtesy of the British Museum).

**Human Glyphs** Interpretation of human glyphs relies on artistic conventions regarding the characters’ genders, poses and orientations. Webber et al. (2024) propose classifiers to determine whether glyphs are male/female (Gender) or standing/sitting (Pose). Binary classifiers learn classification on isolated glyphs for the gender and pose attributes with >97% test accuracy. Examples of male and female human glyphs are shown in Figure 3.

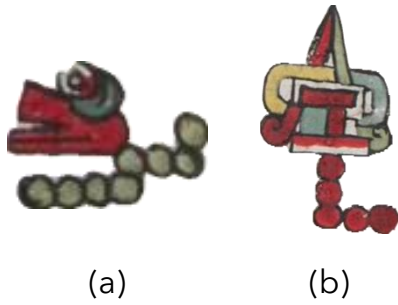


Figure 4: **Name-Date Glyphs:** Name/Date 8 Wind (a) and Year 5 House (b) from the CZN (Provided courtesy of the British Museum). Note the distinguishing presence of the AO symbol in (b).

**Names-Date Glyphs** Transcriptions of name-date glyphs take forms like “4 Wind” or “10 Serpent,” when they represent names, or forms like “Day 1 Alligator” or “Year 13 Rabbit,” when they represent days and years, respectively. Examples of a name-date and a year are provided by Figure 4. Name-date interpretation is more mechanical than interpreting the other glyphs. Glyphs are comprised of one of 20 symbols (only 4 are valid for years) and a number 1 through 13 according to the number of beads affixed to the glyph. First, one classifies glyphs as names/dates or years based on the presence of an ‘AO’ symbol (Boone, 2000), visible in Figure 4 (b), corresponding to a binary classification task. Second are two independent multiclass classification tasks. One is to recognize the name-date’s symbol, and the other to *count* (classify according to quantity) bead subglyphs that convey the name-date’s number (Boone, 2000). Results for these classification tasks already exist in the Mixtec machine interpretation literature (Salunke et al., 2025).

**Toponyms** Toponyms (place signs), complex glyphs in Mixtec codices that typically correspond to a place, often indicate the setting of events recorded in the codices (Boone, 2000; Williams, 2013). Interpretation of toponyms is nuanced and often relies on archaeological evidence (Smith, 1973; Pohl, 1994; Boone, 2000) and sometimes word-play in the Mixtec language (Boone, 2000). As an example, we include a toponym that reads “Yucu Dzaa” or “Hill of the Bird.” (Boone, 2000) from page 45 of the CZN (reverse) as Figure 5.



Figure 5: **Toponyms:** Toponym representing “Yucu Dzaa” or “Hill of the Bird” (Boone, 2000) from page 45 of the CZN (reverse) (Provided courtesy of the British Museum).

### 3 Methodology

In this work, we introduce a formal symbolic machine interpretation of an XML representation of Mixtec codices with aspirations of integrating it into a broader pipeline. Figure 2 lays out this pipeline in detail. As our contribution, we provide a context free grammar (CFG), parser, and interpreter for the Mixtec codices. We situate the XML parser and interpreter at the final *Parsing* and *Output* steps of the semasiographic machine interpretation pipeline we introduce. Assuming correct tokenization (in the XML form defined by the grammar) at the *Sequencing* step and correctly populated attributes at the *Classification* step, we implement *Parsing* by devising a CFG and implementing an ANTLR4-generated parser (Parr, 2013). We accomplish *Output* via the interpreter.

#### 3.1 Rationale

We employ *ANTLR recursive descent parsing* since the relationship between the Mixtec glyphs is hierarchical in nature. By defining the last stages of the pipeline first, we establish a precise design contract for the earlier feature extraction steps. In particular, the parser’s need to resolve nested structures, such as a name-date modifying a human, requires that a finished pipeline must capture not just glyph identity and position, but relational attributes like orientation and containment in *Sequencing*

#### 3.2 Token Structure

In the CFG elaborated by Figure 6, the following token types appear: **Humans** (H), **Years** (Y), **Name-Dates** (ND), **Objects** (OBJ), and **Near Objects** (NEAR\_OBJ). These correspond to classes of glyphs that appear in Mixtec

```

document → XML_DECL DOCUMENT_OPEN (
  SCENE_OPEN sent SCENE_CLOSE
)+ DOCUMENT_CLOSE
sent → clause
      | OBJ ( date clause | clause )
      | date ( OBJ clause | clause )
clause → clause_f+ ( date_tail | obj_tail | ε )
date_tail → date ( OBJ clause_f+ | clause_f+ | ε )
obj_tail → OBJ ( date clause_f+ | clause_f+ | ε )
date → Y ( ND | ε )
clause_f → H ( ND | near_date | ε )
near_date → NEAR_OBJ ( ND | ε )

```

Figure 6: The Context Free Grammar (CFG) implemented by the ANTLR4 parser to check whether Mixtec tokens sequences are valid for interpretation. Uppercase items denote lexer tokens; document is the start rule enclosing one or more scenes, and sentences (sent) take object-initial, date-initial, or clause-only forms.

codices. In addition, XML\_DECL, DOCUMENT\_OPEN, DOCUMENT\_CLOSE, SCENE\_OPEN, and SCENE\_CLOSE are structural tokens used to define the document format. The start symbol document requires an XML declaration and a document wrapper containing one or more scene elements. An enumeration of the token types appears in Table 1.

Tokens of type H represent human glyphs from the Mixtec codices and have three attributes: **Gender (Male/Female)**, **Pose (Sitting/Standing)**, and **Orientation (Left/Right)**. Year and Name-Date tokens encapsulate **Symbol (One of 20 Symbol Names)** and **Number (Integer, 1-13)** attributes.

Tokens of the obj and near\_obj types simply have an **Identity** attribute that labels the object for the parser and interpreter. Assignment of the **Identity** attribute corresponds to a straightforward object recognition task performed on object glyphs. The distinction between these two types of token is that near\_obj represent objects that are associated with a specific person, giving context about them.

### 3.3 Parser

The Mixtec parser is designed for the *Parsing* step of the semasiographic interpretation pipeline in Figure 2 and implements the grammar in Figure 6. It takes XML input, and an example of this XML follows this paragraph. It is built using an ANTLR4 lexer and parser, which process the tokens assuming the token structure we elaborate in the previous section. The lexer recognizes each glyph-bearing XML element (e.g., <human .../>, <year .../>, <obj .../>) as a single token, and the parser applies the context-free rules for documents, scenes, and sentences.

As a concrete example, consider the following XML-form input:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <scene>
    <year x="0" y="0" symbol="House" number="5" />
    <human x="0" y="0" gender="Lord" pose="standing" orientation="right" />
    <name_date x="0" y="0" symbol="Wind" number="4" />
    <human x="0" y="0" gender="Lady" pose="sitting" orientation="left" />
    <name_date x="0" y="0" symbol="Serpent" number="10" />
  </scene>
</document>

```

The parser and interpreter process this input into the following output:

*“In Year 5 House Lord 4 Wind consulted Lady 10 Serpent.”*

The ANTLR4 parser produces a concrete parse tree for valid inputs. A custom visitor then traverses this tree, converts the ANTLR tokens into the internal token classes described in the previous section, and constructs the corresponding abstract syntax tree (AST).

### 3.4 Interpreter

Our interpreter implements the *Output* step of the machine interpretation pipeline by applying a visitor pattern over the ANTLR4 parse tree. The visitor methods construct an abstract syntax tree (AST) from the parser rule contexts (PRCs) and attach interpretation rules to each node.

The algorithm is as follows. For each internal node  $r$  (corresponding to a parser rule context):

1. Visit all of  $r$ 's children and collect their return values.
2. Construct a `TreeNode` object of the appropriate type (e.g., `Clause`, `Date`, `Sent`) using the results from step (1).
3. Return the constructed object if  $r$  is not the start symbol. If  $r$  is the start symbol (document), return the result of calling `interpret()` on the root node, which produces the final textual interpretation.

For each leaf node  $l$  (terminal token):

1. Read the XML string of the token.
2. Construct the appropriate token object (e.g., `Human`, `Year`, `Name-Date`, `Object`, `Near-Object`), populating its attributes from the XML.
3. Wrap the token in a `LeafNode` and return it.



Figure 7: Example from Page 26 of the CZN (obverse) (Provided courtesy of the British Museum) shown in Figure 1. The AST and annotations are removed to exhibit image features. **Interpretation: “In Year 6 Flint Day 7 Eagle Lady 9 Eagle married Lord 6 Alligator”**

Each `TreeNode` implements an `interpret()` method that specifies how its children combine into semasiographic meaning. For example, a `Clause` node may contain a rule recognizing the Mixtec motif of two seated persons of opposite gender in a house, which corresponds to a marriage (Boone, 2000). A `Date` node joins a year (Y) and an optional name-date (ND), while a `Sent` node concatenates its parts into a sentence. The start rule (`document`) aggregates across multiple sentences to yield the final interpretation.

This visitor-based design cleanly separates parsing from interpretation: ANTLR4 enforces the grammar, the visitor constructs the AST, and the `interpret()` methods provide domain-specific semantics grounded in the Mixtec codices.

## 4 Illustrative Example

In this section, we will demonstrate the present contribution using the example in Figure 7. In particular, we will show how the parser and interpreter process an XML representing that scene into the following output:

*“In Year 6 Flint Day 7 Eagle Lady 9 Eagle married Lord 6 Alligator”*

We begin with the XML document that the parser assumes from previous stages in the pipeline, show how it is lexed into a token stream, how it is parsed, and how it is finally interpreted into the above output.

### 4.1 Lexing Mixtec Scenes in XML Encoding

From the stages prior to *Parsing*, we begin with the following XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <scene>
    <year x="0" y="0" symbol="Flint" number="6" />
    <name_date x="0" y="0" symbol="Eagle" number="7" />
    <human x="0" y="0" gender="Lady" pose="sitting" orientation="right" />
    <name_date x="0" y="0" symbol="Eagle" number="9" />
    <object x="0" y="0" identity="house">
    <human x="0" y="0" gender="Lord" pose="sitting" orientation="left" />
    <name_date x="0" y="0" symbol="Alligator" number="6" />
  </scene>
</document>
```

The ANTLR parser combines the lexing and parsing required to convert this raw text in to an AST. In this section, we cover the former. Following the tokens shown in Figure 6, we define a document as an XML declaration (`<?xml ...>`) followed by a document opening tag (`<document>`). Within the document, a number of scenes (at least one) are nested. Each begins with scene opening tag (`<scene>`) and ends with the corresponding closing tag (`</scene>`). The document ends with the closing (`</document>`). Each of these lexical features corresponds in the obvious way to the tokens in the grammar, but they are enumerated explicitly in Table 1.

Token	Feature	Regular Expression
Y	Year	'<year' .*? '/>'
H	Human	'<human' .*? '/>'
ND	Name-Date	'<name_date' .*? '/>'
OBJ	Object	'<obj' .*? '/>'
NEAR_OBJ	Near Object	'<near_obj' .*? '/>'
DOCUMENT_OPEN	Meta	'<document>'
DOCUMENT_CLOSE	Meta	'</document>'
SCENE_OPEN	Meta	'<scene>'
SCENE_CLOSE	Meta	'</scene>'
XML_DECL	Meta	'<?xml' .*? '?>'

Table 1: Token classes, their corresponding features in the Mixtec input, and their regular expression as it appears in our ANTLR grammar file. “Meta” in the **Feature** column indicates that the token is metadata and represents an XML tag rather than input features.

Within the scenes, things are more interesting. The straightforward correspondence between tag names and token types is maintained, but information is encoded within tags that must be retained by the tokens after lexing. However, lexing is only concerned with the type of tokens and their proper order in its output: the token stream, so it merely records the tags within scenes as the corresponding token type (H, ND, NEAR\_OBJ, OBJ, Y) and encoded

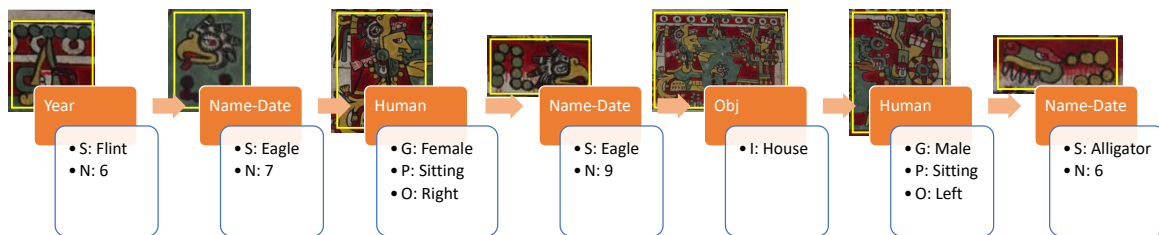


Figure 8: The token stream produced by lexing the XML received from the *Scanning* stage with insets of corresponding features in the example from Page 26 of the CZN (obverse) (Provided courtesy of the British Museum). Full example provided in Figure 7.

the information inside of the tags in these tokens’ attributes. The result is shown in Figure 8. The tokens are ordered according to their appearance within the scene, and the scene is nested within SCENE\_OPEN and SCENE\_CLOSE tokens, which are nested inside the document tokens.

Astute readers may have noted two things about the regular expressions in Table 1 and the XML we assume from the pipeline’s early steps in this section. We discuss these observations here.

First, the attributes populated in each of the XML tags corresponding to feature tokens are not governed by the regular expressions. In each case, there is a wild card inside the that says any input can be placed between the open and close of the XML tag. This policy may seem too loose, but it is actually an implementation decision taken with a view toward the parser’s continued development as other steps of the interpretation pipeline emerge. Likewise, the second observation stems from taking advantage of the loose regular expressions. In each of the XML tags provided in our example, we have ‘x’ and ‘y’ coordinates of 0. These are left as 0, even though the XML parser and interpreter we contribute is not responsible for filling in these attributes, as an example of attributes that future parsers and interpreters will require of the XML. Future contributions at previous steps in the overall pipeline will decorate XML tags with these positional attributes. This exemplifies how permissive regular expressions in this contribution more readily catalyze such advancements in the project.

## 4.2 Parsing From Tokens

The stream of tokens produced in the previous subsection is the input for the parser. The grammar dictated in Figure 6 has been encoded into a gram-

mar file for the ANTLR compiler compiler package complete with token definitions from Table 1. ANTLR generates a parser and lexer for the XML input. While lexing has been covered in depth, we now review how the parser produces an AST for the input through the recursive descent parsing algorithm.

From tokenization of the XML, according to the regular expressions given by Table 1, the parser implements recursive descent and creates an AST for valid inputs by expanding rules from the CFG (see Figure 6). For example, the rule for a document is expanded to get at the one or more sentence constructions within it. In turn, the parser decomposes each sentence into the appropriate form according to the corresponding <scene> tag’s internal lexical features. In Figure 1, we give the AST our XML parser generates for the scene in Figure 7 as a first impression of our contribution.

## 4.3 Unhandled Features

Before our concluding remarks, we will review shortly some examples that our current parser and interpreter are fit to handle. Here, we will give several negative examples and explain briefly why handling them has been left to future work.

In Section 2, we first visited toponyms to describe their place in the human Mixtec interpretation literature. Since that point, we have left them out of the discussion, and analysis of the grammar and regular expressions we provide will show no token type(s) dedicated to them or place for these features in the grammar. The reasons for this follow from existing Mixtec literature, but we have left it for this section to explain.

Recall that the toponym in Figure 5 is read “Yucu Dzaa” which translates to the English “Hill of the

Bird” (Boone, 2000). We observe the head of a bird drawn behind an ornate pattern on the right side of a hill-shaped figure. Less distinct in this example is the human chin drawn through the bird’s lower jaw. This feature is a phonetic component that specifies we should read the bird in this toponym as “Dzaa” (Boone, 2000).

Mixtec toponyms have been actively studied since the 1970s (Smith, 1973; Pohl, 1994; Boone, 2000). Despite this rich tradition, interpretation is very nuanced, as Figure 5’s example shows. Interpretation can rely on phonetic features in the original language (Smith, 1973; Pohl, 1994; Boone, 2000) or even fall back on archaeology (Boone, 2000). Interpretations of most toponyms remained elusive even at the turn of this century (Boone, 2000). For this reason, toponyms are excluded from the present work, and a parser we detail in previous sections is not designed for them.

The number of objects in the CZN alone is large enough that enumerating, detecting, and classifying them could be its own project. We do not try to attempt it here; and the range of objects supported by the current implementation of *Parsing* and *Output* is quite limited in comparison to sum total of objects and near-objects that one could find in the source material.

Likewise, distinguishing between what is an object on its own, such as a house, and what is a near-object, like parts Lord 6 Alligator’s headdress, has not yet been accomplished, either here or in the related work. While it may in that case seem paradoxical to include them in the present implementation, we have reason to include the NEAR\_OBJ type anyway.

Despite the important setting details they impart, we cover above how toponyms are presently excluded because they cannot be well understood by the current method. It stands to reason we would exclude near-objects or incorporate them with the generic object tokens for the same reason. Instead, however, we understand that such features are not just pivotal to understanding the codices but also know they have a place within the sentences we form from scenes. Further, while no method currently exists, the problem does not seem as doubtful as toponym interpretation from a machine learning perspective. We therefore include near-objects in the grammar to anticipate future works that will enable us to distinguish them at earlier pipeline steps.

## 5 Future Work

Future work must expand the limited grammar, parser, and interpreter we propose before these tools are suitable for broad application. In particular, the coverage of sentence structures in the current grammar is narrow and should be extended to include additional motifs and constructions described in the Mixtec literature. While the present contribution and example serve as proof of concept, significant work remains before it can be applied to a variety of examples.

Another key direction is the development of systems that complete steps in the machine interpretation pipeline prior to *Parsing*. In particular, we anticipate systems that automatically segment glyphs in codex facsimiles and output the XML required by our regular expressions. The community must devise computer vision tools that are competent to segment scenes in the CZN into the various token classes we identify.

Equally important is extending classifier coverage beyond humans and name-dates. In particular, automated methods for detecting OBJ and NEAR\_OBJ tokens are necessary for coherent scene-level interpretation.

Toponyms are, as we have already discussed, important sources of setting information for Mixtec interpretation. Without sound methods for deciphering toponyms, setting details in Mixtec Codices, important context for overall narratives, will remain out of reach even as the parser and interpreter support for other features expands. However, as we review previously, the correct interpretations of these complex, enigmatic glyphs are often in debate among Mixtec experts, which significantly complicates evaluation of automatic toponym recognition. Thus, it is difficult to say whether a machine interpretation of toponyms is possible given current understanding of those glyphs.

## Limitations

The present grammar, corresponding parser, and the interpreter are quite limited in their scope, only covering a handful of Mixtec scene structures.

Related works have yet to produce any system that completes the steps of the machine interpretation pipeline up to *Scanning*, and these steps are necessary for the generation of XML encoded scenes that the present contribution is fit to interpret. The current work cannot serve any practical function without intensive human supervision. Without

an automated method for producing XML, the significant hurdle of manually preprocessing any input into the XML encoding our system expects stands in the way of future progress and improvements to interpreting Mixtec Codices. Further, while we devise the machine interpretation pipeline as a framing device in which to place our contribution, its use presupposes that the early steps are feasible with present computer vision and natural language processing techniques. While we are confident in the ability of language-vision researchers, we make very few claims about how automated construction of the required XML may be affected.

In particular, there is currently no classifier to label object glyphs or to populate the **Identity** attribute of OBJ and NEAR\_OBJ tokens, but these tokens are included in the grammar and handled by the machine parser and interpreter because object handling is necessary for any coherent interpretation of Mixtec codices. There is also not yet recognition systems to distinguish objects in the Mixtec Codices as OBJs versus NEAR\_OBJs.

Curiously, our example in Figure 7 illustrates a final limitation. Citing nearby context, experts conclude that the scribe who authored the CZN (obverse) mistakenly renders the Lord’s name as “6 Alligator” instead of “5 Alligator” (Williams, 2013). Detecting such errors is an important task but well beyond our present capabilities. The present contribution merely interprets XML representations of the Mixtec Codices as they are written. Indeed, it will be an interesting challenge for language-vision research to detect typographical errors such as the one in Page 26 of the CZN, and we look forward to future contributions in this direction.

## Acknowledgments

The authors gratefully acknowledge Arnold and Lisa Goldberg, whose financial support helped to make this work possible.

## References

- Elizabeth Hill Boone. 2000. *Stories in red and black : pictorial histories of the Aztecs and Mixtecs / Elizabeth Hill Boone.*, 1st ed. edition. University of Texas Press, Austin.
- B.E. Byland and J.M.D. Pohl. 1994. *In the Realm of 8 Deer: The Archaeology of the Mixtec Codices.* University of Oklahoma Press.
- Alfonso Caso. 1977. *Reyes y reinos de la mixteca /*

*Alfonso Caso.*, 1. ed. edition. Sección de obras de antropología. Fondo de Cultura Económica, México.

- Martin de la Iglesia, Franziska Diehr, Uwe Sikora, Sven Gronemeyer, Maximilian Behnert-Brodhun, Christian Prager, and Nikolai Grube. 2021. *The code of maya kings and queens: Encoding and markup of maya hieroglyphic writing.* *Journal of the Text Encoding Initiative.*
- Liudmila Fedorova, Antonio Perri, et al. 2024. Emblematic techniques as textual strategies in non-linear and linear scripts. *GRAPHOLINGUISTICS AND ITS APPLICATIONS*, 9:75–99.
- Richard C. E. Long. 1926. *The zouche codex.* *The Journal of the Royal Anthropological Institute of Great Britain and Ireland*, 56:239–258.
- Zelia Nuttall. 1902. *Codex Nuttall; facsimile of an ancient Mexican codex belonging to Lord Zouche of Harynworth, England.* Cambridge, Mass., Peabody Museum of American Archaeology and Ethnology, Harvard University.
- Terence Parr. 2013. *The Definitive ANTLR 4 Reference.* Pragmatic Programmers.
- Antonio Perri, Luciano Perondi, and Daniele Capo. 2024. Rethinking unicode: how to digitally encode non-linear written artefacts? a tentative encoding of codex mendoza, folio 2r. In *Proceedings of Grapholinguistics in the 21st Century, 2024.*
- J.M.D. Pohl. 1994. *The Politics of Symbolism in the Mixtec Codices.* Vanderbilt University publications in anthropology. Vanderbilt University.
- Girish Salunke, Christopher Driggers-Ellis, and Christian Grant. 2025. *Classifying name-date and year figures in mixtec codices.* In Taylor Arnold, Margherita Fantoli, and Ruben Ros, editors, *Computational Humanities Research 2025*, pages 1210–1218. Anthology of Computers and the Humanities.
- M.E. Smith. 1973. *Picture Writing from Ancient Southern Mexico; Mixtec Place Signs and Maps.* Civilization of the American Indian series. University of Oklahoma Press.
- Nancy P. Troike. 1978. *Fundamental changes in the interpretations of the mixtec codices.* *American Antiquity*, 43(4):553–568.
- Alexander Webber, Zachary Sayers, Amy Wu, Elizabeth Thorner, Justin Witter, Gabriel Ayoubi, and Christian Grant. 2024. *Analyzing finetuned vision models for Mixtec codex interpretation.* In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pages 42–49, Mexico City, Mexico. Association for Computational Linguistics.
- Robert Lloyd Williams. 2013. *The complete codex Zouche-Nuttall : Mixtec lineage histories and political biographies / by Robert Lloyd Williams ; foreword*

*by Rex Koontz.*, 1st ed. edition. Linda Schele series in Maya and pre-Columbian studies. University of Texas Press, Austin.

# Temporal-Linguistic Adaptive Streaming for Continuous Sign Language Translation

Arshia Kermani, Habib Irani, Deautun Ross, Vangelis Metsis

Department of Computer Science

Texas State University

San Marcos, TX 78666, USA

{arshia.kermani, habibirani, drr175, vmetsis}@txstate.edu

## Abstract

Real-time sign language translation must generate text incrementally as signs arrive, yet existing streaming policies treat glosses as a flat token sequence and discard the temporal rhythm of signing. Inter-gloss pauses reliably mark sentence boundaries in continuous discourse, but policies such as Wait-k cause arbitrary cross-boundary fragmentation. We propose Temporal-Linguistic Adaptive Streaming (TLAS), which fuses a Temporal Pause Detector (TPD, tracking inter-gloss interval statistics via an exponential moving average) and a Linguistic Readiness Estimator (LRE, a trained neural head on a frozen T5 encoder) through an Adaptive Fusion Gate (AFG). A proactive timeout fires *before* the next gloss arrives when the inter-gloss gap exceeds a threshold, producing clean sentence segmentation without oracle boundary information. We also contribute a synthetic discourse dataset of 1,400 ASL discourse groups with LLM-generated per-gloss timestamps and introduce a continuous-stream evaluation paradigm requiring autonomous boundary detection from an unbroken gloss stream. Under such conditions, TLAS significantly outperforms current heuristic baselines, such as Wait-k, and methods relying solely on linguistic content.

## 1 Introduction

Approximately 70 million people worldwide rely on sign languages as their primary mode of communication (World Health Organization, 2021). In healthcare triage, legal proceedings, and emergency response, the absence of a real-time interpreter can delay diagnoses, produce inadmissible testimony, or prevent timely access to services. Automated sign language translation systems capable of operating in real time would substantially reduce these barriers; yet the large majority of existing systems require a complete, pre-recorded input before producing any output, making them unsuitable for interactive deployment.

A central challenge in building such systems is the structural mismatch between visual sign production and spoken-language text. American Sign Language (ASL) employs spatial grammar, topic-comment ordering, and non-manual markers that do not correspond one-to-one with English syntax. A direct video-to-text approach must simultaneously solve recognition (mapping continuous visual motion to discrete linguistic units) and translation (mapping one language to another), two tasks whose error rates compound. The standard solution is a two-stage pipeline: a vision module converts video frames into a sequence of *glosses* (discrete lexical labels for individual signs), and a translation model maps the resulting gloss sequence to fluent target text. The monotonic gloss-to-token correspondence makes streaming translation tractable: as each gloss arrives, the system can decide, based on accumulated context, whether to wait for additional signs or commit to generating output. This paper addresses the gloss-to-text stage under real-time, continuous-stream conditions.

Existing streaming policies borrowed from spoken-language simultaneous translation fail to exploit a modality-specific signal that is readily available in the gloss stream: the temporal rhythm of signing. Within a sentence, inter-gloss intervals range from approximately 300 to 650 ms; between sentences, signers pause for 2 to 7 seconds. Wait-k (Ma et al., 2019) and TransLLaMa (Koshkin et al., 2024) treat the gloss stream as a flat token sequence and discard this temporal information entirely, fragmenting multi-sentence discourse arbitrarily across boundaries and forcing the translation backend to complete incoherent partial sentences or produce hallucinated continuations. A policy that monitors inter-gloss gap statistics could instead detect sentence boundaries before the next sentence begins, enabling clean segmentation and preserving discourse coherence across translations.

A further obstacle to progress is the absence of

publicly available datasets that pair continuous discourse gloss streams with per-gloss arrival timestamps. Without such a benchmark, it is impossible to measure continuous-stream segmentation quality or to study the effect of timestamp degradation (e.g., jitter introduced by a real-time vision module) on translation policies. Existing resources such as ASLG-PC12 (Othman and Jemni, 2012) provide isolated sentence pairs without temporal annotations, and sentence-level evaluation conceals the boundary-detection failures that dominate real-world deployment.

We propose **Temporal-Linguistic Adaptive Streaming** (TLAS), a streaming policy that fuses temporal pause detection and neural linguistic readiness estimation through an adaptive gate, with a proactive timeout mechanism for clean inter-sentence segmentation in continuous discourse. We further contribute (a) a synthetic discourse dataset of 1,400 ASL discourse groups with per-gloss timestamps spanning three conversation types, and (b) a continuous-stream evaluation paradigm in which policies must segment and translate an unbroken multi-sentence gloss stream without oracle boundaries. Timestamp robustness experiments demonstrate that the temporal and linguistic signals are complementary: TLAS-temporal maximizes quality under reliable timing while TLAS-linguistic provides a timestamp-invariant floor. The main contributions of this work are:

- **TLAS architecture:** a streaming policy that fuses temporal pause detection and neural linguistic readiness estimation through an adaptive gate, with a proactive timeout mechanism for clean inter-sentence segmentation in continuous discourse.
- **Continuous-stream evaluation paradigm (E2):** a discourse-level benchmark protocol in which policies must segment and translate an unbroken multi-sentence gloss stream, together with quantitative analysis of segmentation quality across five streaming policies.
- **Synthetic discourse dataset:** 1,400 ASL discourse groups with per-gloss timestamps spanning three conversation types, released to support future research in continuous-stream sign language translation.
- **Timestamp robustness analysis:** a systematic study of translation quality under three

timestamp conditions (realistic, uniform, and noisy), demonstrating that TLAS-linguistic is effectively timestamp-invariant while TLAS-temporal degrades gracefully, with implications for deployment in the presence of vision-module latency.

## 2 Related Work

### 2.1 Sign Language Translation

Sign language translation decomposes into two stages: visual recognition, which maps continuous video to a discrete gloss sequence, and gloss-to-text translation, which produces fluent target language output (Kermani et al., 2025). Early recognition systems combined hand-crafted visual features with HMM-based sequence models to handle multiple signers across large vocabularies (Koller et al., 2015). Camgöz et al. (Camgöz et al., 2018) introduced the first neural gloss-to-text system, adapting attention-based sequence-to-sequence architectures originally developed for spoken language translation and establishing BLEU as the field’s standard metric. Transformer-based systems that jointly optimize recognition and translation end-to-end further improved performance (Camgöz et al., 2020). Data augmentation via sign back-translation (Zhou et al., 2021) and pretraining with multilingual models such as mBART (Liu et al., 2020) and T5 (Raffel et al., 2020) have subsequently pushed translation quality on standard benchmarks. Continuous-discourse resources such as How2Sign (Duarte et al., 2021) for American Sign Language and the TVB-HKSL-News corpus (Niu et al., 2024) for Hong Kong Sign Language provide multi-sentence content beyond isolated sentence pairs, but they do not pair gloss streams with the per-gloss arrival timestamps required to evaluate streaming policies under realistic temporal conditions. All of these systems, however, assume batch access to a complete gloss sequence before producing any output, rendering them unsuitable for the interactive scenarios in which real-time translation is most urgently needed.

### 2.2 Simultaneous Translation

Simultaneous (streaming) translation generates target tokens incrementally as source tokens arrive. Ma et al. (Ma et al., 2019) proposed the Wait- $k$  policy, which delays output by a fixed  $k$  source tokens and then alternates read and write steps; its deterministic simplicity makes it a durable baseline

despite the inability to adapt to content boundaries. Arivazhagan et al. (Arivazhagan et al., 2019) introduced Monotonic Infinite Lookback (MILk) attention, which replaces the hard lag with a learned, monotonically constrained attention distribution; Monotonic Multihead Attention (MMA) (Ma et al., 2020) extended this mechanism to multi-head settings with independent per-head step decisions, enabling end-to-end training of the read/write policy. More recently, Koshkin et al. (Koshkin et al., 2024) demonstrated that large language models can be fine-tuned to emit a special <WAIT> token when context is insufficient for translation, integrating the policy decision into the model itself.

These approaches share a critical limitation when applied to sign language: read/write decisions are conditioned exclusively on linguistic content, and the temporal dimension of the gloss stream is discarded entirely. In continuous signing, inter-gloss intervals within a sentence (300–650 ms) differ by an order of magnitude from the pauses between sentences (2–7 s). To our knowledge, no prior streaming translation policy explicitly models inter-gloss timing as a boundary signal, and applying existing policies to a continuous discourse stream causes arbitrary cross-boundary fragmentation rather than coherent sentence-by-sentence translation.

## 3 Methodology

### 3.1 Problem Formulation

We formalize streaming sign language translation as a read/write decision problem over a time-stamped token stream. Each incoming gloss is a pair  $(g_t, \tau_t)$ , where  $g_t \in \mathcal{V}$  is a discrete gloss token and  $\tau_t \in \mathbb{R}^+$  is the wall-clock arrival time in seconds since stream onset. The system maintains a gloss buffer  $\mathcal{B}_t = \{(g_1, \tau_1), \dots, (g_t, \tau_t)\}$  and at each step issues either **READ** (wait for the next gloss) or **WRITE** (flush  $\mathcal{B}_t$  to the translation backend and reset). In the continuous-stream setting, the input is a multi-sentence discourse group delivered as a single unbroken sequence with no oracle boundary markers; a **WRITE** at a sentence boundary yields a coherent translation segment, while a premature or delayed **WRITE** produces a fragment or cross-boundary concatenation that no translation model can recover from. The objective is to maximize translation quality while maintaining discourse coherence across consecutive translations.

### 3.2 TLAS Architecture

The TLAS policy integrates two complementary readiness signals through a learned decision gate. A Temporal Pause Detector (TPD) monitors inter-gloss arrival times and emits a high score during inter-sentence pauses. A Linguistic Readiness Estimator (LRE) scores the grammatical completeness of the accumulated buffer using a neural head on a frozen T5 encoder (Raffel et al., 2020). An Adaptive Fusion Gate (AFG) combines both scores to issue **READ** or **WRITE** decisions. A proactive timeout mechanism fires between glosses when the current inter-gloss gap exceeds  $M$  times the signer’s running average, producing clean sentence boundaries before the first gloss of the next sentence arrives. Figure 1 illustrates the complete data flow.

#### 3.2.1 Temporal Pause Detector (TPD):

Upon receiving gloss  $g_t$  at time  $\tau_t$ , the TPD computes the interval  $\Delta_t = \tau_t - \tau_{t-1}$  and updates an exponential moving average:

$$\hat{\mu}_t = \alpha \cdot \Delta_t + (1 - \alpha) \cdot \hat{\mu}_{t-1} \quad (1)$$

where  $\alpha = 0.3$  controls the adaptation rate. The EMA is initialized at  $\hat{\mu}_0 = 450$  ms, set to the midpoint of the within-sentence inter-gloss range (300–650 ms), to suppress cold-start fluctuations during the first few glosses. The pause score is a linear ramp between normal pace and the pause multiplier  $M = 2.5$ :

$$s_t^{\text{TPD}} = \text{clamp}\left(\frac{\Delta_t/\hat{\mu}_t - 1}{M - 1}, 0, 1\right) \quad (2)$$

When  $\Delta_t = \hat{\mu}_t$  (ratio 1.0), the score is 0; when  $\Delta_t \geq M \cdot \hat{\mu}_t$ , the score saturates at 1. Because the score is defined relative to  $\hat{\mu}_t$  rather than a fixed threshold, the TPD adapts automatically to each signer’s natural pace without recalibration.

#### 3.2.2 Linguistic Readiness Estimator (LRE):

The LRE estimates the semantic completeness of the accumulated gloss buffer using a lightweight neural head on the T5 encoder. Given encoder hidden states  $H \in \mathbb{R}^{B \times L \times d}$ , the head computes an attention-masked mean pool  $\bar{h} \in \mathbb{R}^d$  over non-padding positions and maps it to a scalar readiness score:

$$l_t = \sigma(W_2 \text{ReLU}(W_1 \bar{h} + b_1) + b_2) \quad (3)$$

where  $W_1 \in \mathbb{R}^{256 \times 768}$  and  $W_2 \in \mathbb{R}^{1 \times 256}$ , with dropout 0.1 between layers ( $\approx 197\text{K}$  parameters).

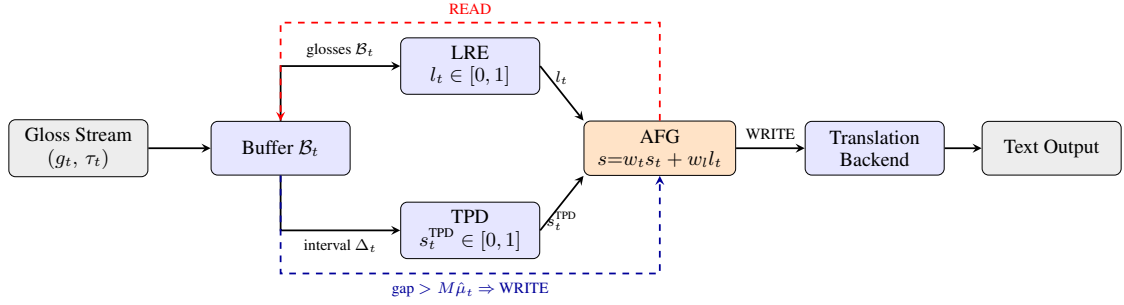


Figure 1: TLAS architecture. The LRE encodes accumulated glosses with a frozen T5 encoder and maps the pooled representation to a completeness score  $l_t$ ; the TPD maps the inter-gloss interval  $\Delta_t$  to a pause score  $s_t^{\text{TPD}}$ . The AFG combines both signals and issues WRITE or READ (dashed red). A proactive timeout (dashed blue) issues WRITE directly when the elapsed gap exceeds  $M \cdot \hat{\mu}_t$ .

The head is trained separately after T5 fine-tuning. Oracle readiness labels are generated by translating every gloss prefix  $G_{1:t}$  with the frozen T5 model and computing ROUGE-L (Lin, 2004) against the reference:

$$r_t^* = \text{ROUGE-L}(T5(G_{1:t}), y^*) \quad (4)$$

Monotonicity is enforced by replacing each label with the running maximum:  $r_t^* \leftarrow \max(r_t^*, r_{t-1}^*)$ . The head is then trained with MSE loss on these oracle scores. For non-T5 backends, the LRE operates as a standalone scorer: the fine-tuned T5 encoder and LRE head are loaded locally and shared across all TLAS instances, incurring no per-gloss API calls.

### 3.2.3 Adaptive Fusion Gate (AFG):

The AFG combines both signals through a weighted sum:

$$s = w_t \cdot s_t^{\text{TPD}} + w_l \cdot l_t \quad (5)$$

with  $w_t = 0.4$  and  $w_l = 0.6$ , giving linguistic completeness a slightly higher weight because it directly predicts translation quality. The weights  $w_t$  and  $w_l$  are empirically tuned hyperparameters. The gate evaluates three ordered conditions and issues WRITE on the first satisfied:

1. **Safety valve:** buffer length  $|\mathcal{B}_t| \geq L_{\max} = 6$ . Prevents unbounded accumulation when both signals are weak.
2. **Joint threshold:**  $s \geq \theta = 0.40$ . With  $w_l = 0.6$ , this requires  $l_t \geq 0.67$  in the absence of any temporal signal, i.e., the LRE is confident that the buffer forms a complete translatable unit.

3. **Strong pause override:**  $s_t^{\text{TPD}} \geq 0.8$  and  $l_t \geq 0.3$ . Trusts an unambiguous signer pause even when the LRE is not yet fully confident, reflecting the high correlation between long inter-sentence pauses and syntactic completion in ASL.

If none is satisfied, the AFG returns READ. We note that the fusion weights  $w_t$  and  $w_l$  and the AFG thresholds are tuned on a held-out validation set rather than learned end-to-end (see Section 4.4); treating the gate as a differentiable policy and optimizing it directly under a quality–latency reward is a natural direction for future work (Section 6).

### 3.2.4 Proactive Timeout:

The most discriminative boundary signal in continuous discourse arrives *between* glosses. The proactive timeout operates outside the per-gloss TPD→LRE→AFG pipeline: when the elapsed time since the last gloss exceeds  $M \cdot \hat{\mu}_t$  and at least one gloss is buffered, the system issues a WRITE directly, before the next token arrives. Given typical within-sentence averages of  $\hat{\mu}_t \approx 450$  ms and  $M = 2.5$ , this timeout fires approximately 1.1 s into an inter-sentence pause, cleanly separating two sentences without waiting for the first gloss of the following sentence to confound segmentation. The proactive timeout is disabled for the TLAS-linguistic ablation, which must operate purely on linguistic content.

## 3.3 Ablations and Baselines

We evaluate three TLAS configurations to isolate each component’s contribution:

- **TLAS (full):**  $w_t=0.4$ ,  $w_l=0.6$ ; both TPD and LRE active; proactive timeout enabled.

- **TLAS-temporal**:  $w_t=1.0$ ,  $w_l=0.0$ ; the LRE returns a constant 0 but the AFG structure is otherwise unchanged; proactive timeout enabled. Isolates the temporal signal.
- **TLAS-linguistic**:  $w_t=0.0$ ,  $w_l=1.0$ ; the TPD EMA is updated normally but its score is clamped to 0 before reaching the AFG; proactive timeout disabled. Isolates the linguistic signal.

All three configurations use identical AFG thresholds, ensuring that performance differences reflect signal quality rather than threshold tuning. We compare against four baselines: **Batch (oracle)**, which splits at ground-truth boundaries (upper bound); **Batch (non-oracle)**, which accumulates every gloss in the discourse group and submits them as a single concatenated string to the translator, producing one output for what may be five or six sentences of content—because the single output aligns only partially with the first reference sentence and has no correspondence to later ones, BLEU collapses at positions 1 and beyond, and the gap between Batch (non-oracle) and Batch (oracle) quantifies the full value of correct sentence boundary detection; **Wait-k** ( $k=3$ ) (Ma et al., 2019), which alternates read/write with fixed lag; and **TransLLaMa** (Koshkin et al., 2024), which emits <WAIT> when context is insufficient. All baseline policies are evaluated with a discourse context window of zero prior translations, while TLAS variants use a sliding window of three. This asymmetric configuration reflects each policy’s natural operating regime rather than disadvantaging the baselines: when a baseline fragments a discourse stream across sentence boundaries, its prior outputs are cross-boundary fragments rather than coherent sentence translations, so populating its context buffer with such outputs would propagate segmentation errors into subsequent prompts and further degrade quality. Baselines therefore operate under the more favorable zero-context setting. An evaluation under matched zero-context conditions for all policies, which would further isolate the segmentation contribution from any residual benefit of context, is left to future work.

### 3.4 Translation Backends

TLAS is backend-agnostic: the policy layer calls `translate(buffer, context)` and receives a text string, with no assumptions about the underlying model. We evaluate three backends.

**T5** (local, fine-tuned): We fine-tune T5-base (Raffel et al., 2020) on a multi-source training mixture comprising 50K ASLG-PC12 gloss–English pairs, SIGNUM German Sign Language pairs for cross-domain vocabulary exposure, and TransLLaMa-style streaming examples in which early prefixes (first third of glosses) target the special <WAIT> token and mid-prefixes (first half) target partial translations. Discourse context is incorporated by prepending a sliding window of the three most recent prior translations as a prefix, separated by a context delimiter. Training uses AdamW with effective batch size 32, learning rate  $5 \times 10^{-5}$ , warmup ratio 0.1, weight decay 0.01, label smoothing 0.1, and FP16 mixed precision for 5 epochs on a single GPU.

**Gemini** (cloud API): We use `gemini-3.1-flash-lite-preview` via the Google AI API. Each WRITE invocation sends a structured prompt containing the discourse context window and the buffered glosses; the model returns a single-sentence translation. No fine-tuning is applied.

**Ollama** (local API): We deploy `gpt-oss:120b` through a local Ollama server, providing an on-premise alternative that avoids data egress. The same prompt format as Gemini is used. For both non-T5 backends, the LRE standalone scorer loads the fine-tuned T5 encoder locally and computes readiness scores without invoking the external API.

## 4 Experimental Setup

In this section, we describe our experiments and evaluation methodology. All hyperparameters, training scripts, evaluation code, and the synthetic discourse dataset are publicly available.<sup>1</sup>

### 4.1 Datasets

**ASLG-PC12** (Othman and Jemni, 2012) is a parallel corpus of approximately 87,000 ASL gloss–English pairs. We use a shuffled split of 50,000 training pairs, 10,000 validation pairs, and 10,000 test pairs (seed 42); sentence-level evaluation (E1) draws 100 test examples with synthetic uniform timestamps (450 ms per gloss).

**Synthetic Discourse Dataset** (contributed in this work) comprises 1,400 multi-sentence discourse groups generated via Gemini LLM, spanning monologue ( $\approx 40\%$ ), deaf-deaf dialog

<sup>1</sup><https://github.com/imics-lab/tlas-gloss2text>

( $\approx 30\%$ ), and deaf-hearing dialog ( $\approx 30\%$ ). Per-gloss timestamps are calibrated to real ASL timing: 300–650 ms within sentences, 1.5–7 s between sentences. The first 200 groups (888 deaf sentences) constitute the test split; the remaining 1,200 groups ( $\approx 5,234$  context pairs) form the training split.

**SIGNUM** consists of 779 German Sign Language sentence pairs from everyday conversational topics; we include it during T5 fine-tuning for cross-domain vocabulary exposure and reserve a held-out subset for cross-dataset generalization evaluation.

## 4.2 Evaluation Paradigms

**E1** (sentence-level) evaluates 100 ASLG-PC12 test sentences with synthetic uniform timestamps (450 ms per gloss). Since all policies operate within known sentence boundaries, the temporal signal is neutralized and TLAS degenerates to max-lag triggering. E1 isolates pure linguistic translation quality but cannot assess boundary detection.

**E2** (continuous-stream, primary experiment) feeds 200 test discourse groups as unbroken gloss streams using LLM-generated timestamps, with no oracle sentence boundaries provided. Inter-sentence pauses of 1.5–7 s give TLAS a strong temporal signal via the proactive timeout, while baselines fragment the stream arbitrarily. We report corpus-level BLEU and break results down by discourse position to measure quality drift within a group.

## 4.3 Evaluation Metrics

We report four metrics: **BLEU** (Papineni et al., 2002) (corpus-level n-gram precision, computed with SacreBLEU); **ROUGE-L** (Lin, 2004) (longest common subsequence recall); **SBERT cosine similarity** (semantic similarity via Sentence-BERT, robust to paraphrase); and **chrF++** (character n-gram F-score with word unigram recall, informative for morphologically varied output). For E2 we additionally report **retention rate** (policy BLEU divided by oracle BLEU) and **position-stratified BLEU** (computed per discourse position to reveal whether segmentation errors accumulate across a group).

## 4.4 Hyperparameters

Table 1 summarizes all TLAS hyperparameters. The TPD smoothing factor  $\alpha$  and pause multiplier  $M$  were set from the inter-gloss timing statistics of the training data; the AFG weights and thresholds were validated on 20 held-out discourse groups not

Table 1: TLAS hyperparameter settings.

Component	Parameter	Value
TPD	EMA smoothing $\alpha$	0.3
	Pause multiplier $M$	2.5
	EMA prior $\hat{\mu}_0$	450 ms
LRE	Hidden dim	256
	Dropout	0.1
AFG	$w_t$ (temporal weight)	0.4
	$w_l$ (linguistic weight)	0.6
	Joint threshold $\theta$	0.40
	Strong pause threshold	0.80
	Min. readiness for pause	0.30
	Max lag $L_{\max}$	6
Policy	Discourse context window	3
	Wait-k lag $k$	3

Table 2: E2 continuous-stream results, T5 backend (200 groups, 888 sentences). Ret. = BLEU / oracle BLEU. Best streaming value in **bold**.

Policy	BLEU	ROUGE-L	Ret.
Batch (oracle) $\uparrow$	24.49	.593	—
Batch	1.74	.068	7.1%
Wait-k ( $k=3$ )	1.86	.286	7.6%
TransLLaMa	4.76	.385	19.4%
TLAS-linguistic	17.18	.474	70.1%
TLAS	21.51	.536	87.9%
TLAS-temporal	<b>23.70</b>	<b>.564</b>	<b>96.8%</b>

included in the test split. None of these values are learned end-to-end; the gate is a tuned policy rather than a trained one.

## 5 Results and Discussion

### 5.1 E2: Continuous-Stream Discourse Evaluation

Table 2 presents T5 results on 200 discourse groups (888 sentences). Two distinct performance tiers are immediately apparent. The three TLAS variants cluster between 17 and 24 BLEU (70–97% retention), while all baselines fall below 5 BLEU (under 20% retention). This separation holds across every metric (BLEU, ROUGE-L, SBERT, and chrF++), confirming that the gap is not an artifact of the BLEU brevity penalty.

#### 5.1.1 Cross-Backend Results:

Table 3 extends the comparison across all three backends. TLAS-temporal and full TLAS consistently achieve retention above 80%, while traditional baselines peak at 56.8%. The higher absolute retention of the Gemini and Ollama baselines relative to T5 is consistent with the larger LLMs partially masking segmentation failures by com-

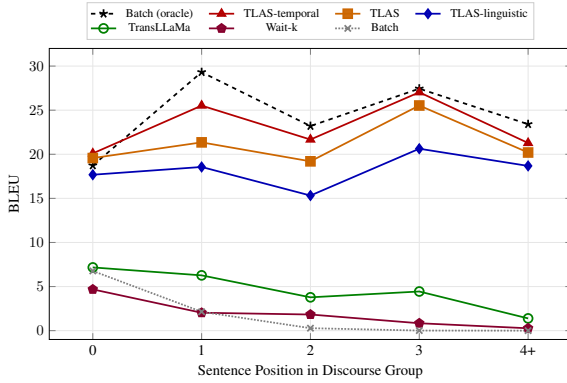


Figure 2: Position-stratified BLEU across discourse positions in E2 (T5 backend, 200 groups). TLAS variants track the oracle trajectory at all positions, while baselines collapse by position 2 due to cross-boundary fragmentation. The gap between tiers widens at later positions, confirming that segmentation errors accumulate monotonically for policies lacking temporal boundary detection.

pleting incoherent fragments into plausible-looking text rather than reflecting genuinely better segmentation; in either case, TLAS variants still lead every traditional baseline by at least 24 percentage points on every backend. Figure 2 confirms this: TLAS variants track the oracle trajectory across all five discourse positions, while Wait-k and Batch decline monotonically to near-zero BLEU by position 3, indicating cumulative segmentation failure rather than translation model weakness.

### 5.1.2 Position-Stratified Analysis:

Table 4 reports BLEU stratified by sentence position within the discourse group (T5 backend). This analysis reveals the mechanism underlying the two-tier separation. Wait-k collapses from 4.68 BLEU at position 0 to 0.27 at position 4+; Batch (non-oracle) reaches 0.00 by position 3. The three TLAS variants maintain stable quality across all positions, with TLAS-temporal staying within 5 BLEU points of the oracle at every depth. The root cause is boundary detection, not translation capability per se: baselines that fragment across sentence boundaries submit incoherent gloss subsets to the translator, producing outputs with near-zero overlap with any individual reference sentence.

### 5.1.3 Timestamp Robustness:

Table 5 reports BLEU under three timestamp conditions for TLAS variants on E2 (T5 backend). TLAS-linguistic varies by less than 0.1 BLEU points across all three conditions, confirming zero

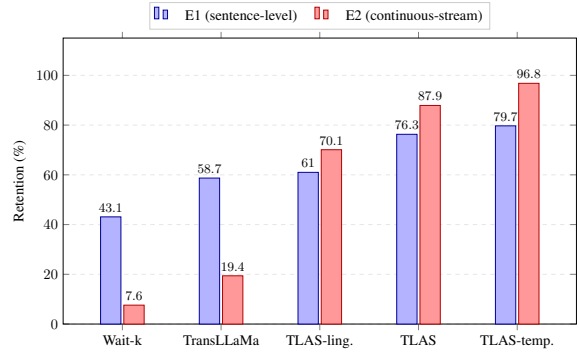


Figure 3: Retention rates under sentence-level (E1) vs. continuous-stream (E2) evaluation (T5 backend). Wait-k and TransLLaMa collapse from E1 to E2 (43.1%→7.6% and 58.7%→19.4%, respectively), while TLAS variants maintain or improve retention under discourse conditions, exposing a critical failure mode that sentence-level evaluation conceals.

sensitivity to timestamp quality. TLAS-temporal degrades by 57% from realistic to noisy timestamps (23.70 → 10.07): Gaussian jitter of  $\sigma=500$  ms corrupts the EMA, causing premature or delayed proactive timeouts. A crossing point occurs at uniform timestamps: TLAS-linguistic (17.09) overtakes TLAS-temporal (15.61), because uniform gaps suppress the temporal signal to exactly zero while the LRE continues to operate normally. Full TLAS degrades more gracefully than temporal-only (11.53 vs. 10.07 under noise), since the linguistic weight provides a stabilizing floor; however, the remaining 46% decline from realistic to noisy conditions confirms that the temporal component remains the dominant vulnerability when timestamp quality is poor.

## 5.2 E1: Sentence-Level Results

Table 6 presents sentence-level results on 100 ASLG-PC12 test examples with synthetic uniform timestamps (T5 backend). Because all policies receive isolated sentences with no inter-sentence pauses, the TPD emits near-zero scores throughout and TLAS degenerates to max-lag ( $L_{\max}=6$ ) triggering. Under these conditions TLAS-temporal (79.7% retention) and full TLAS (76.3%) lead all streaming methods, demonstrating that the max-lag safety valve provides adequate within-sentence segmentation quality even when the temporal cue is suppressed.

Table 3: E2 results across all three backends. Ret. = BLEU / backend oracle BLEU.

Policy	T5 (oracle: 24.49)		Gemini (oracle: 24.39)		Ollama (oracle: 12.53)	
	BLEU	Ret.	BLEU	Ret.	BLEU	Ret.
Batch	1.74	7.1%	5.71	23.4%	1.90	15.2%
Wait-k	1.86	7.6%	11.49	47.1%	7.12	56.8%
TransLLaMa	4.76	19.4%	12.54	51.4%	5.64	45.0%
TLAS-linguistic	17.18	70.1%	15.79	64.7%	7.53	60.1%
TLAS	21.51	87.9%	21.46	88.0%	10.23	81.6%
TLAS-temporal	<b>23.70</b>	<b>96.8%</b>	<b>22.13</b>	<b>90.7%</b>	<b>11.56</b>	<b>92.3%</b>

Table 4: Position-stratified BLEU in E2 (T5 backend). Position 0 = first sentence in group; 4+ = fifth or later.

Policy	Pos 0	Pos 1	Pos 2	Pos 3	Pos 4+
Batch (oracle) $\uparrow$	18.73	29.31	23.20	27.44	23.41
Batch	6.77	2.14	0.28	0.01	0.00
Wait-k	4.68	2.03	1.83	0.84	0.27
TransLLaMa	7.17	6.27	3.78	4.44	1.39
TLAS-linguistic	17.68	18.56	15.32	20.63	18.68
TLAS	19.57	21.35	19.20	25.53	20.20
TLAS-temporal	<b>20.10</b>	<b>25.52</b>	<b>21.67</b>	<b>27.03</b>	<b>21.28</b>

Table 5: Timestamp robustness: BLEU under three conditions (T5, E2). Oracle: 24.49.

Policy	Realistic	Uniform	Noisy
TLAS-temporal	<b>23.70</b>	15.61	10.07
TLAS	21.51	16.68	11.53
TLAS-linguistic	17.18	<b>17.09</b>	<b>17.18</b>

### 5.3 Discussion

The E2 results establish that correct sentence boundary detection is the primary determinant of translation quality in continuous discourse. TLAS-temporal’s 96.8% retention demonstrates that the temporal signal alone is sufficient for near-perfect segmentation when timestamps are reliable; the EMA  $\hat{\mu}_t$  tracks within-sentence timing and cleanly detects inter-sentence pauses via the proactive timeout. The fusion design is not a simple average: the strong-pause override ensures unambiguous temporal boundaries are respected even when the LRE is uncertain, while the joint threshold allows the LRE to accelerate translation for linguistically complete prefixes when temporal cues are weak. Figure 3 quantifies this paradigm gap: baselines lose 35–39 percentage points of retention moving from E1 to E2, while TLAS-temporal gains 17 points (79.7%  $\rightarrow$  96.8%), demonstrating that continuous-stream evaluation is essential for any policy intended for real-world deployment.

The contrast between E1 and E2 underscores the necessity of continuous-stream evaluation. Wait-

Table 6: E1 sentence-level results, T5 backend ( $n=100$ , synthetic timestamps). Ret. = BLEU / oracle BLEU.

Policy	BLEU	ROUGE-L	Ret.
Batch $\uparrow$	73.84	.919	—
TLAS-temporal	58.88	.869	79.7%
TLAS	56.36	.867	76.3%
TLAS-linguistic	45.04	.824	61.0%
TransLLaMa	43.35	.821	58.7%
Wait-k ( $k=3$ )	31.84	.757	43.1%

k achieves 43.1% retention on E1 but only 7.6% on E2; TransLLaMa achieves 58.7% on E1 but only 19.4% on E2. These collapses do not appear in E1 because oracle boundaries are implicit in the single-sentence structure, eliminating cross-boundary fragmentation entirely. Prior streaming translation benchmarks operate exclusively in E1-like settings, which cannot detect this failure mode. Under E2, TLAS-temporal’s retention advantage over Wait-k grows from 36.6 percentage points (E1) to 89.2 percentage points (E2), a three-fold amplification that sentence-level evaluation conceals. Any streaming policy intended for deployment in continuous signing should be evaluated under an E2-like paradigm; an E1-only evaluation cannot assess the core challenge of real-time boundary detection.

The timestamp robustness experiments (Table 5) carry direct deployment implications. In systems with reliable sub-100 ms timestamp precision (purpose-built sign recognition pipelines), TLAS-temporal maximizes quality. In systems with known timing variance (video codecs, high-latency GPU inference), TLAS-linguistic provides a robust fallback. The full fusion is recommended as the default when timestamp reliability is unknown or variable, as it outperforms linguistic-only under reliable timing and outperforms temporal-only under noise.

Several limitations bound the current study. First, the discourse dataset is synthetically generated; while per-gloss timestamps are validated against

empirical signing rates, the conversation topics, turn structures, and inter-sentence pauses may not fully capture the rhythm of authentic Deaf community interactions, which include hesitations, self-corrections, and more variable pacing. Second, the end-to-end system from raw video to text has not been integrated, and recognition errors from an upstream vision module would compound with the TPD and LRE in ways not measured here: gloss insertions and deletions would corrupt the EMA, perturbing both the inter-gloss interval statistic and the proactive timeout, while noisy gloss content would degrade the LRE’s readiness estimates. Third, wall-clock inference latency under real-time constraints, where GPU memory bandwidth, tokenizer overhead, and API round-trip times all compound, has not been characterized; a full latency analysis is necessary before deployment claims can be substantiated. Finally, the AFG fusion weights and thresholds are tuned on a held-out validation set rather than learned end-to-end, and an evaluation under matched zero-context conditions for all policies, which would most cleanly isolate the segmentation contribution from any residual context benefit, is left for future work.

## 6 Conclusion

We presented TLAS, a streaming policy that fuses temporal pause detection and learned linguistic readiness as complementary signals for continuous sign language segmentation and translation. On the primary E2 evaluation (200 discourse groups, 888 sentences), TLAS-temporal achieves 96.8% retention of oracle quality while Wait-k reaches only 7.6%. Position-stratified BLEU confirms that TLAS sustains near-uniform quality across all discourse positions, whereas baselines collapse by position 2 due to cross-boundary fragmentation. Timestamp robustness analysis shows the signals are complementary: TLAS-temporal maximizes performance under reliable timing while TLAS-linguistic provides a stable 70.1% floor invariant to timestamp noise. Future work will explore replacing the T5 decoder with a state-space model for  $O(1)$  per-step inference and implicit cross-sentence context propagation, and optimize the AFG weights via reinforcement learning to replace the currently tuned gate with a policy adapted directly to the deployment environment.

## References

- Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. [Monotonic infinite lookback attention for simultaneous machine translation](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28–August 2, 2019, Volume 1: Long Papers*, pages 1313–1323. Association for Computational Linguistics.
- Necati Cihan Camgöz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. [Neural sign language translation](#). In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018*, pages 7784–7793. IEEE.
- Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. [Sign language transformers: Joint end-to-end sign language recognition and translation](#). In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020*, pages 10020–10030. Computer Vision Foundation / IEEE.
- Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. 2021. [How2Sign: A large-scale multimodal dataset for continuous American sign language](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2735–2744.
- Arshia Kermani, Habib Irani, and Vangelis Metsis. 2025. [Finetuning pre-trained language models for bidirectional sign language gloss to text translation](#). In *Proceedings of the Workshop on Sign Language Processing (WSLP)*, pages 73–81.
- Oscar Koller, Jens Forster, and Hermann Ney. 2015. [Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers](#). *Computer Vision and Image Understanding*, 141:108–125.
- Roman Koshkin, Katsuhito Sudoh, and Satoshi Nakamura. 2024. [TransLLaMa: LLM-based simultaneous translation system](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12–16, 2024*, pages 461–476. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: a package for automatic evaluation of summaries](#). In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.

- Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. [STACL: simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28–August 2, 2019, Volume 1: Long Papers*, pages 3025–3036. Association for Computational Linguistics.
- Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020. [Monotonic multihead attention](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*. OpenReview.net.
- Zhe Niu, Ronglai Zuo, Brian Mak, and Fangyun Wei. 2024. A Hong Kong sign language corpus collected from sign-interpreted TV news. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 636–646, Torino, Italy. ELRA Language Resources Association (ELRA) and the International Committee on Computational Linguistics (ICCL).
- Achraf Othman and Mohamed Jemni. 2012. English-ASL gloss parallel corpus 2012: ASLG-PC12. In *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC 2012*, pages 151–154, Istanbul, Turkey.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [BLEU: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6–12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- World Health Organization. 2021. Deafness and hearing loss. <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss>. Accessed: 2026-03-18.
- Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. 2021. [Improving sign language translation with monolingual data by sign back-translation](#). In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19–25, 2021*, pages 1316–1325. IEEE.

# FADE: Probing the Limits of VLMs on fine-grained OCR

Deep Shah  
Google LLC

shahdeep@google.com

Nehal Kathrotia  
Google LLC

nehalk@google.com

Sanket Badhe  
Google LLC

sanketbadhe@google.com

## Abstract

Multimodal Large Language Models (MLLMs) have achieved remarkable success in semantic visual reasoning, yet their capacity for fine-grained, low-level perception remains critically under-evaluated. This perceptual fragility limits their reliability in noisy, real-world environments where visual signals are degraded. Furthermore, existing benchmarks often entangle visual perception with language priors, masking these underlying deficits. To address this, we introduce the **FAint numeric Detection Evaluation (FADE)** dataset, a novel evaluation suite designed to probe the limits of zero-shot Optical Character Recognition (OCR) in frontier MLLMs. By embedding synthetic, strictly numerical sequences over cluttered natural backgrounds at varying levels of transparency ( $\alpha$ ), FADE explicitly disentangles pure visual perception from semantic predictability. We evaluate state-of-the-art models including Gemini 3.0, Claude 4.5 Sonnet, and Gemma 3 against a specialized UNet segmentation baseline. Our results reveal a striking limitation in frontier architectures: while they achieve near-perfect transcription at high visibility, their performance collapses under high transparency. Conversely, the UNet pipeline maintains robust spatial grounding, significantly outperforming generalist models at the lowest visibility thresholds. FADE provides a reproducible dataset to expose and diagnose the perceptual breakage points of modern multimodal systems.

**Dataset:** [FADE on Hugging Face](#)

## 1 Introduction

The trajectory of Vision-Language Models (VLMs) and Multimodal Large Language Models (MLLMs) has been marked by extraordinary advancements, fundamentally bridging the gap between visual perception and natural language understanding (Achiam et al., 2023; Team et al., 2023). Trained on vast web-scale image-text datasets, frontier architectures such as GPT-4, Gemini, and Claude

have achieved unprecedented success in high-level semantic tasks, including visual question answering (VQA), complex scene interpretation, and nuanced image captioning (Liu et al., 2023; Li et al., 2023). Consequently, these models are increasingly deployed in real-world, open-ended environments where they are expected to act as general-purpose visual agents.

Current MLLMs often rely heavily on powerful language priors and global semantic context, rather than robust, low-level visual processing (Tong et al., 2024; Villa et al., 2025). While they excel at identifying prominent objects or reasoning about a scene’s overall narrative, their performance deteriorates sharply when confronted with fine-grained perceptual tasks that lack semantic anchors. One of the most demanding tests of this capability is Optical Character Recognition (OCR) in the wild. In real-world scenarios, text is frequently obscured by poor lighting, complex background textures, or partial transparency—conditions where context alone cannot recover the missing characters (Zhu et al., 2024).

The inability of VLMs to reliably perform robust text-in-image extraction carries profound security and safety implications for digital platforms. As automated content moderation systems increasingly rely on multimodal models to enforce community guidelines, malicious actors have adapted by embedding harmful content such as hate speech, scam URLs, phone-numbers or illicit narratives directly into images and memes (Wang et al., 2025, 2024). Because standard text-only filters cannot parse image pixels, a VLM’s OCR along with the MLLM reasoning capability serves as the primary line of defense. When VLMs fail to detect visually perturbed, highly transparent, or cluttered text, bad actors can successfully execute visual prompt injections and moderation evasion tactics. Consequently, poor OCR performance directly translates to a degraded safety posture, allowing harmful content to

propagate across digital ecosystems unchecked.

Despite these high-stakes implications, existing evaluation paradigms have largely failed to isolate and quantify this specific perceptual fragility. Traditional OCR benchmarks predominantly feature high-contrast, opaque text, or rely on semantic contexts (like street signs or document headers) that allow language models to seamlessly "guess" obscured words (Singh et al., 2021). Conversely, modern VLM benchmarks tend to entangle visual perception with complex logical reasoning, making it difficult to determine whether a model failed because it could not reason, or simply because it could not *see* the underlying visual signal (Villa et al., 2025; Tong et al., 2024). There is a need for a controlled, objective benchmark that tests the exact breakage thresholds of multimodal perception without the confounding variable of semantic predictability.

To address this critical gap, we introduce the **FAint numeric Detection Evaluation**, a novel evaluation suite explicitly designed to stress-test the limits of fine-grained visual grounding and zero-shot OCR in frontier VLMs. Our dataset embeds synthetic, strictly numerical sequences, ensuring no linguistic context can aid prediction over highly cluttered, diverse backgrounds sampled from the COCO dataset. Crucially, we isolate visual degradation as a single independent variable by systematically modulating the transparency ( $\alpha$ ) of the watermark across a gradient of visibility, ranging from highly visible ( $\alpha = 0.8$ ) to near-imperceptible ( $\alpha = 0.2$ ).

By benchmarking state-of-the-art models—including Gemini 3.0, Claude 4.5 Sonnet, and Gemma 3 against a specialized UNet segmentation baseline, we reveal a striking deficit in frontier architectures. While VLMs achieve near-perfect transcription at high visibility, their zero-shot capabilities collapse under high transparency and background clutter. In contrast, our dedicated segmentation pipeline maintains robust spatial grounding, significantly outperforming the generalist models at the lowest visibility thresholds. Through this work, we provide the community with a reproducible framework to diagnose and rectify the low-level perceptual shortcomings of modern multimodal systems.

## 2 Related Work

### 2.1 Multimodal Large Language Models

With the remarkable advancements of Large Language Models (LLMs), recent research has extended their capabilities to multimodal domains by integrating visual information, giving rise to Multimodal Large Language Models (MLLMs) (Achiam et al., 2023; Team et al., 2023; Li et al., 2023; Liu et al., 2023). These models typically align visual features from pre-trained image encoders with LLMs via modality adaptation layers (Dosovitskiy et al., 2020). Early works like BLIP-2 (Li et al., 2023) pioneered this architecture by pre-training on image-text datasets and fine-tuning on task-specific benchmarks, while subsequent models like LLaVA (Liu et al., 2023) advanced this approach by leveraging synthetic instruction-following data. Frontier models exhibit strong performance in complex scene interpretation (Zhou et al., 2023). Yet, because these architectures leverage joint vision-language spaces, evaluating whether their success arises from robust spatial perception or powerful language priors remains an ongoing methodological challenge. This reliance on semantic priors can overshadow a critical need to scrutinize their fundamental, low-level perception and spatial grounding skills.

### 2.2 Evaluating Fine-Grained Perception

While MLLMs excel at global image understanding, they often struggle with fine-grained tasks requiring precise recognition, localization, and data extraction (Li et al., 2024; Zhou et al., 2025). Existing benchmarks designed to probe these limitations face primary methodological challenges. In many traditional perception benchmarks, visual assessment is often entangled with reasoning; questions focusing on semantic concepts allow models to rely on language priors rather than pure visual input. Furthermore, to improve evaluation reliability, many benchmarks (such as MME (Fu et al., 2023) and SEED-Bench (Li et al., 2024)) adopted multiple-choice formats drawn from existing Visual Question Answering (VQA) datasets, which raises concerns about data contamination and true zero-shot evaluation.

Other benchmarks have focused on specific spatial reasoning deficits, such as object counting in dense scenes (Amini-Naieni and Zisserman, 2025) or chart comprehension (Masry et al., 2022; Methani et al., 2020). While these datasets have

driven progress in visual data extraction, they largely feature high-contrast, clearly delineated targets.

### 2.3 Visual Noise and Character Recognition

The capacity of MLLMs to comprehend abstract or low-signal visual data, such as reading text overlaid on complex backgrounds remains a critical frontier. While traditional Optical Character Recognition (OCR) (Wang et al., 2023) systems are highly specialized, frontier MLLMs are increasingly expected to perform zero-shot text extraction in the wild. Our Watermark Benchmark Dataset departs from existing evaluation methodologies by explicitly disentangling perception from semantics. By utilizing numerical watermarks with modulated transparency ( $\alpha$ ) across cluttered COCO backgrounds, we remove any possible semantic cues; the model must perceive the underlying patterns directly. This provides a reproducible benchmark to identify the exact breakage thresholds of multimodal perception, testing how deeply frontier models can ground themselves spatially to extract low-signal numeric data.

## 3 The Watermark Benchmark Dataset

A key contribution of this work is the curation and release of a specialized evaluation dataset designed to probe the limits of fine-grained visual reasoning and Optical Character Recognition (OCR) in frontier Vision-Language Models (VLMs). While standard benchmarks focus on high-contrast, legible text, this dataset introduces controlled transparency to identify the exact breakage thresholds of multimodal perception.

### 3.1 Composition and Diverse Backgrounds

To ensure the dataset reflects the complexity of natural scenes, background images were sampled from the COCO (Common Objects in Context) (Lin et al., 2014) dataset. These images encompass a diverse variety of textures, lighting conditions, and cluttered environments. Synthetic watermarks comprising random numerical digit sequences were overlaid onto these backgrounds. The digits are colored white, isolating transparency (opacity) as the single independent variable, neutralizing color-contrast bias.

### 3.2 Fine-Grained Transparency Modulation ( $\alpha$ )

The primary feature of this dataset is its systematic modulation of the alpha blending parameter ( $\alpha$ ). This design allows researchers to evaluate how models transition from confident recognition to perceptual failure as a visual signal degrades.

We generate four distinct subsets across a gradient of visibility (See Fig. 1)

- **High Visibility** ( $\alpha = 0.8$ ): Serves as the control baseline to establish upper-bound model performance.
- **Medium Visibility** ( $\alpha = 0.5$ ): Mirrors standard, visible watermarking standards.
- **Subtle Visibility** ( $\alpha = 0.3$ ): Designed to test the boundary of standard feature extraction.
- **Low Visibility** ( $\alpha = 0.2$ ): Pushes the boundaries of native multimodal zero-shot capabilities.

By structuring the dataset around these transparent text, we provide a reproducible benchmark for evaluating how deeply frontier models can ground themselves spatially to extract low-signal text.

### 3.3 Dataset Composition and Structural Properties

To ensure the dataset (Fig. 1) is robust and statistically sound, we standardized the structural properties of the generated images and text overlays.

- **Scale and Splitting:** The dataset contains 2,600 images designated for training the segmentation baselines. For evaluation, a separate test bank of 1,000 images was evaluated independently across each of the four transparency levels ( $\alpha \in \{0.2, 0.3, 0.5, 0.8\}$ ) where lower  $\alpha$  corresponds to more transparent, totaling 4,000 test inferences per model.
- **Resolution:** All images are standardized to a resolution of  $480 \times 640$  pixels in RGB format.
- **Watermark Characteristics:** The watermark vocabulary is strictly numerical, consisting of digits 0–9. To maintain uniformity and control for length-based bias, each watermark contains exactly nine digits. As seen in Fig. 2, digits are uniformly distributed for all the different  $\alpha$ .



Figure 1: Visual examples of watermarked images with varying opacity levels.

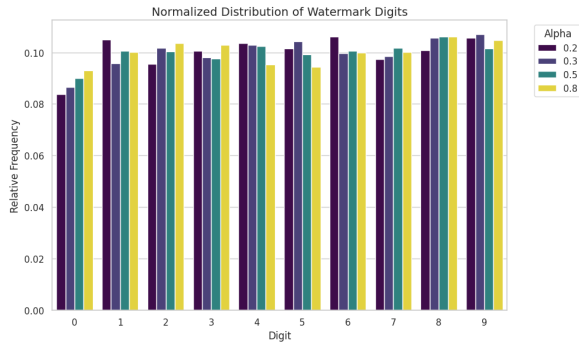


Figure 2: Distribution of digits in the dataset

- **Spatial Grounding:** To isolate the visual degradation caused by transparency from the difficulties of edge-of-frame detection (Chen et al., 2025), all digits are statically positioned at the center of the image.

The richness of the benchmark stems from the background imagery. Sourced from the COCO dataset, these backgrounds capture complex, everyday scenes where diverse objects are depicted in their natural context. The models are therefore subjected to realistic challenges such as background occlusion, varying object scales, cluttered visual gradients, and intricate spatial relationships.

## 4 Setup

### 4.1 Baseline: UNet Architecture and Training

To perform watermarking mask generation, we employed a UNet (Ronneberger et al., 2015) architecture characterized by its symmetrical encoder-decoder structure and skip connections (See 2). This design ensures that high-resolution spatial features from the contracting path are preserved and combined with the upsampled output to produce precise segmentation masks, predicting the watermarked text in our case.

- **Inference Pipeline:** During the inference stage, the trained UNet model processes the

input image to generate a predicted binary mask. This mask serves as a spatial filter to isolate potential watermark regions. The segmented regions defined by the mask are then passed to an Optical Character Recognition (OCR) engine, which performs the final text prediction and extraction.

- **Data Representation:** The model was trained using pairs of source images and their corresponding binary watermarking masks. An example of this training input is illustrated in Figure 3.
- **Model Configuration:** The encoder follows a progressive series of filter depths (16, 32, 64, and 128), leading to a central bottleneck bridge of 256 channels. Each block in the network consists of two  $3 \times 3$  convolutional layers followed by Batch Normalization and ReLU activation.
- **Training Parameters:** The model was optimized using the Adam algorithm with a learning rate of 0.001 and a momentum of 0.9. Training was conducted with a batch size of 32 on images resized to  $480 \times 640$  pixels.

### 4.2 Vision Language Models Evaluation Setup

To assess the native text extraction and visual reasoning capabilities of frontier Vision-Language Models (VLMs), we conducted a series of direct image-to-text experiments. Each model is provided with the source image and a zero-shot prompt (See Fig. 4) instructing it to identify and extract the subtly embedded numbers within the image.

We evaluated the benchmark against a diverse range of state-of-the-art multimodal architectures:

- **Gemini 3.0 Pro and Flash:** Google’s latest frontier models, designed for high-fidelity visual reasoning and multimodal data extraction.

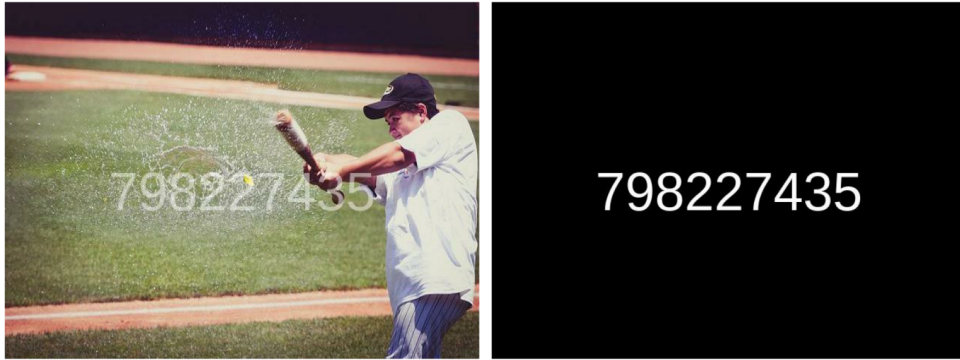


Figure 3: Image and corresponding mask used to train UNet network

**Zero-Shot Prompt**

You are an OCR.  
 Extract the 9-digit number subtly embedded in the foreground of the image.  
 Return the numbers between the tag `<number>` and `</number>`.  
 If you are not able to find any number return `<number>None</number>`.

Figure 4: The exact zero-shot instruction prompt provided to all Vision-Language Models during evaluation.

- **Gemma 3 27B:** A high-capacity open-weights model (Team et al., 2025) utilized to evaluate the performance of multimodal understanding in a resource-efficient framework.
- **Claude 4.5 Sonnet:** Anthropic’s advanced multimodal model, included to provide a comparative baseline for cross-architecture robustness in transcribing text from complex visual contexts.

### 4.3 Evaluation Metrics

To rigorously assess the performance of both the segmentation and the subsequent text extraction by various Vision-Language Models (VLMs), we employed two primary metrics. These metrics evaluate the model’s ability to correctly identify and transcribe the numeric sequences embedded within the images.

- **Exact Match (EM):** This is a strict metric that requires the predicted string to be identical to the ground-truth string. A score of 1 is assigned if the strings match perfectly (ignoring case), and 0 otherwise. This metric is

particularly useful for assessing the reliability of the system in high-precision scenarios.

- **Character Error Rate (CER):** The CER provides a more granular view of the OCR performance by calculating the Levenshtein distance (the number of insertions, deletions, and substitutions required) between the predicted text and the ground truth, normalized by the length of the ground truth (Neudecker et al., 2021).

During evaluation, these metrics were tracked across varying levels of transparency ( $\alpha$ ) to determine the breakdown point for each model’s visual reasoning capabilities.

## 5 Results

We evaluate our proposed UNet+OCR pipeline against four state-of-the-art Vision-Language Models (VLMs) across standard and cropped image settings. The primary metrics tracked are Exact Match Accuracy (Acc.) and Character Error Rate (CER) over a transparency gradient ( $\alpha \in \{0.2, 0.3, 0.5, 0.8\}$ ). The results are summarized in Table 1.

### 5.1 Overall Watermark Extraction Accuracy

As transparency increases (lower  $\alpha$ ), we observe a non-linear decay in accuracy across all tested models. At the control baseline of  $\alpha = 0.8$ , the Gemini family of models achieves near-perfect performance, with Gemini Flash 3.0 topping zero-shot accuracy at 0.955. However, performance deteriorates rapidly as  $\alpha$  scales down to 0.2.

Our proposed UNet (Ours) segmentation pipeline significantly outperforms all zero-shot frontier models under standard image settings. At the most difficult visibility threshold ( $\alpha = 0.2$  and  $\alpha = 0.3$ ), UNet maintains an accuracy which is

Table 1: Comparison of Accuracy and CER across models at varying transparency levels ( $\alpha$ ).

Image	$\alpha$	Sonnet 4.5		Flash 3.0		Pro 3.0		Gemma 3		UNet (Ours)	
		Acc.	CER	Acc.	CER	Acc.	CER	Acc.	CER	Acc.	CER
Standard	0.2	0.040	0.732	0.233	0.324	0.229	0.326	0.062	0.648	<b>0.478</b>	<b>0.210</b>
	0.3	0.104	0.546	0.446	0.165	0.441	0.173	0.141	0.480	<b>0.792</b>	<b>0.053</b>
	0.5	0.329	0.261	0.776	0.046	0.765	0.049	0.385	0.215	<b>0.834</b>	<b>0.043</b>
	0.8	0.672	0.075	<b>0.955</b>	<b>0.009</b>	0.952	0.009	0.768	0.050	0.948	0.014
Cropped	0.2	0.030	0.771	0.289	0.294	0.284	0.291	0.062	0.647	—	—
	0.3	0.088	0.581	0.480	0.149	0.493	0.144	0.140	0.479	—	—
	0.5	0.315	0.282	0.797	0.040	0.813	0.036	0.387	0.214	—	—
	0.8	0.628	0.095	0.951	0.010	0.947	0.010	0.766	0.050	—	—

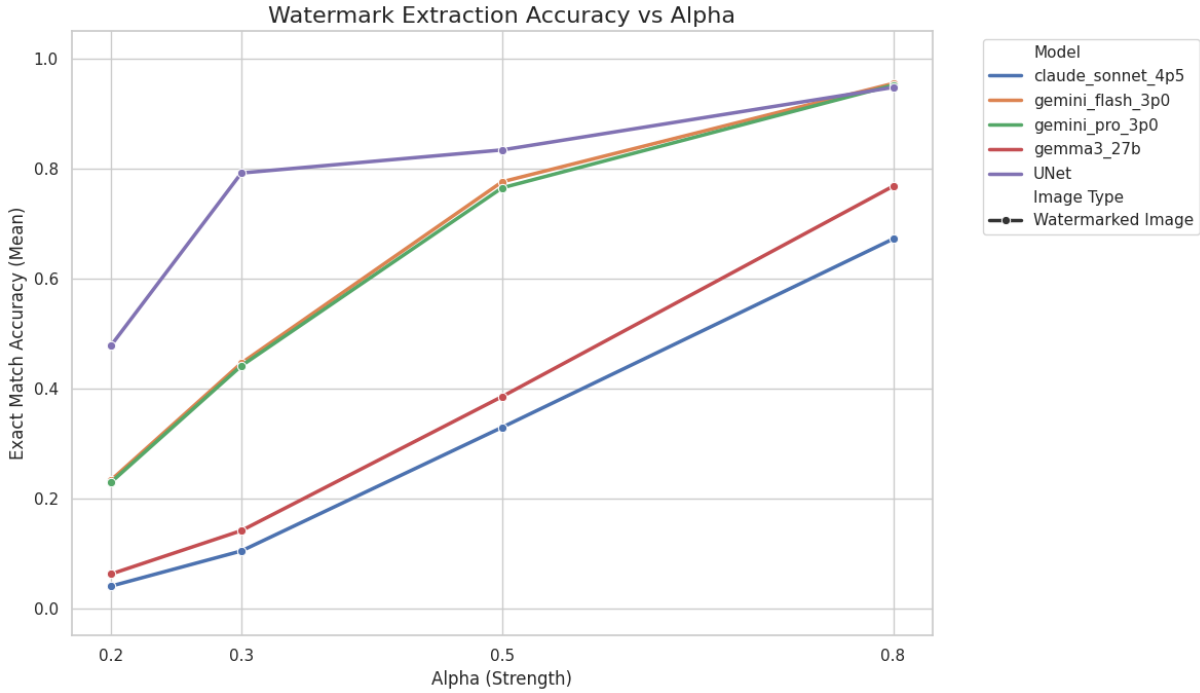


Figure 5: Accuracy detection at different alpha

double that of best performing VLM models. This validates our hypothesis that global attention mechanisms of current general-purpose VLMs fails to detect details when the visual signals are sparse.

## 5.2 Character-Level Performance and Error Rate

While accuracy measures perfect string matches, the Character Error Rate (CER) (Fig. 6 gives a more granular view of localized failures. We observe that models often perceive the presence of a watermark but struggle to transcribe all nine digits without substitutions or deletions.

Gemma 3 27B suffers the highest degradation in character fidelity, yielding a CER of 0.648 at  $\alpha = 0.2$  on standard imagery. In contrast, the UNet pipeline holds a low CER of 0.210 under identical conditions. The tracking of  $(1 - \text{CER})$

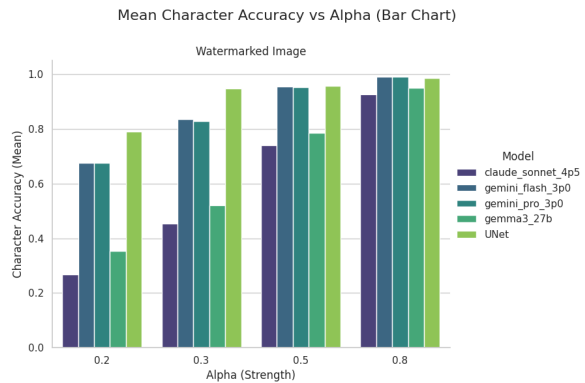


Figure 6: Mean Character Accuracy as a function of Watermark Strength (Alpha). Higher Alpha values indicate more visible watermarks, leading to significantly higher extraction accuracy across all tested vision-language models.

as a proxy for character-level accuracy reveals that while zero-shot models fragment on long numerical strings, segmentation-coupled OCR reliably preserves token-by-token alignment even against high-clutter COCO backgrounds.

### 5.3 Impact of Image Cropping on Detection Success

To improve the attention signals which are driving force of Vision transformer, we cropped the image which only captures the region containing the numeric text (see B). Comparing the Standard and Cropped partitions in Table 1 reveals a distinct architectural behavioral shift. Isolating the watermark via localized bounding box improves the zero-shot extraction.

For instance, Gemini Flash 3.0 jumps from 0.233 Accuracy at  $\alpha = 0.2$  on standard imagery to 0.289 when restricted to cropped dimensions. The improvements are also observed at  $\alpha = 0.3$  & 0.5. No strong improvements were observed for images with  $\alpha = 0.8$ , which is primarily due to the fact that text is already very clear and visible.

### 5.4 Digit-Specific Confusion Analysis

To understand the precise failure modes of frontier VLMs, we analyze the confusion matrix of prediction mistakes for Gemini 3 Pro at  $\alpha = 0.3$  (Fig. 7). To ensure a direct character-to-character comparison, this analysis is restricted to watermarked images where the predicted string length matches the ground truth exactly.

The visual data reveals several structural insights into model perception under high transparency:

- **High-Frequency Morphological Confusion:** The most significant error mode involves true digit 9 being misidentified as 0 (52 instances) and true digit 3 being misidentified as 8 (51 instances). These high-frequency errors suggest that subtle closures in curved digits are easily obscured by background textures, leading the model to hallucinate complete ellipsoids or connected loops.
- **Vertical and Diagonal Feature Loss:** There is a notable density of errors involving the misidentification of true digit 8 as 3 (45 instances) and true digit 6 as 0 (39 instances). Furthermore, true digit 3 is frequently confused with 2 (40 instances) or 5 (38 instances), highlighting a difficulty in resolving the spe-

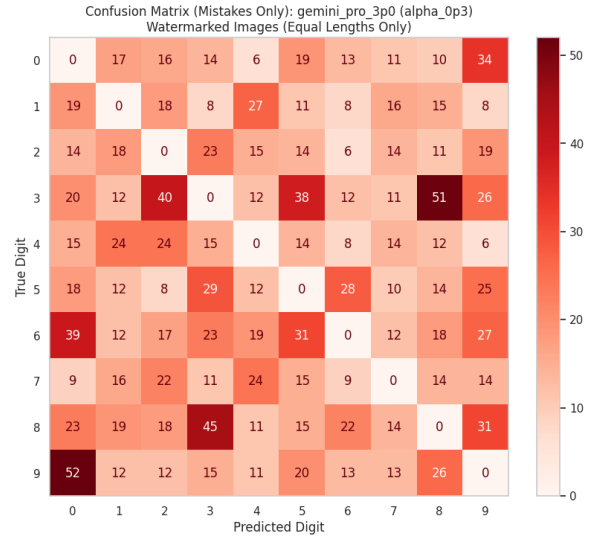


Figure 7: Confusion matrix (Mistakes Only) for Gemini 3 Pro at  $\alpha = 0.3$ , filtered for equal-length predictions. Rows represent the true digits, while columns represent the predicted digits.

cific orientation of horizontal and diagonal strokes when  $\alpha$  is low.

- **Structural Simplification and Reconstructive Bias:** The model exhibits a tendency to hallucinate non-existent strokes when resolving faint visual signals, frequently misidentifying 1 as 4 (27 instances) and 0 as 9 (34 instances). Rather than failing gracefully by outputting a blank or a partial stroke, the model forces a completion of the character. This suggests a perceptual collapse driven by dataset priors, where the model relies on top-down linguistic and structural biases to fill in the blanks of ambiguous visual inputs, resulting in false positives for structurally similar digits.

## 6 Conclusion

In this work, we introduced the **FAint numeric Detection Evaluation (FADE)** benchmark to rigorously evaluate the fine-grained visual perception of frontier Multimodal Large Language Models (MLLMs). By systematically modulating the transparency ( $\alpha$ ) of numerical sequences against cluttered natural backgrounds, FADE provides a unique framework to disentangle pure visual grounding from semantic and linguistic predictability.

Our comprehensive evaluation of state-of-the-art models—including Gemini 3.0, Claude 4.5 Sonnet, and Gemma 3—reveals a significant percep-

tual gap in general-purpose architectures. While these models demonstrate high proficiency in high-visibility contexts, their performance decays nonlinearly as transparency increases, often collapsing at  $\alpha \leq 0.3$ .

A critical finding of our error analysis is the presence of *reconstructive bias*: when faced with low-signal visual inputs, MLLMs tend to hallucinate non-existent strokes to complete digits likely based on dataset priors rather than failing gracefully. Furthermore, our experiments with image cropping demonstrate that reducing the spatial search space provides only modest improvements, suggesting that the bottleneck lies in the visual encoder’s inability to register sparse signals rather than a failure of global attention.

As MLLMs are increasingly integrated into safety-critical domains such as content moderation and autonomous agents, addressing these low-level perceptual blind spots is important. We hope that FADE serves as a standard diagnostic tool for the community to facilitate the development of more resilient, spatially aware visual encoders capable of genuine fine-grained visual reasoning.

## 7 Future Directions

### 7.1 Analysis of Visual Token Perturbation

A critical next step is investigating the representational shift within the latent space. Future work should quantify how drastically the output embeddings of the Vision Transformer (ViT) (Dosovitskiy et al., 2020) differ between pristine backgrounds and their watermarked counterparts. Understanding the magnitude and nature of this token-level perturbation, particularly at low transparency ( $\alpha$ ) thresholds, will help isolate whether the primary failure mode is the visual encoder failing to register the weak signal or the language model failing to interpret it.

### 7.2 Task-Aware Visual Encoding

Current VLM architectures typically utilize a ViT to encode images into a static sequence of tokens, which are subsequently processed by the language model. While this query-agnostic approach is efficient, the generated visual tokens remain identical regardless of the user’s prompt. This limits the encoder’s ability to dynamically focus on task-relevant, low-contrast features. Recent research into task-aware or query-conditioned visual encoders (Ganz et al., 2024) presents a promising

alternative. Applying such dynamic architectures to this benchmark would allow the visual encoder to actively search for and amplify fine-grained,  $\alpha$ -transparent signals based on the specific extraction prompt, potentially bridging the perceptual gap observed in our zero-shot evaluations.

## Limitations

While the FADE dataset provides a rigorous framework for evaluating visual perception, it currently has scope constraints. First, the watermark vocabulary is restricted to numerical digits. Incorporating full alphanumeric characters and symbols would provide a broader range of morphological complexities. Second, all text is statically positioned at the center of the image to isolate the variable of transparency. Shifting the spatial distribution of text such as placing watermarks near margins would determine how VLMs handle positional embeddings and edge-of-frame detection challenges.

## Generative AI Usage

All study design, literature review, synthesis, and writing were conducted by the authors. Generative AI tools (Gemini) were used only for grammar checking and proofreading during the final polishing of the manuscript. No generative AI system was used to generate content, interpret prior work, or draw conclusions. The authors reviewed and approved all final text and remain fully responsible for the content of the paper.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Niki Amini-Naieni and Andrew Zisserman. 2025. Countgd++: Generalized prompting for open-world counting. *arXiv preprint arXiv:2512.23351*.
- Shiqi Chen, Tongyao Zhu, Ruochen Zhou, Jinghan Zhang, Siyang Gao, Juan Carlos Niebles, Mor Geva, Junxian He, Jiajun Wu, and Manling Li. 2025. Why is spatial reasoning hard for vlms? an attention mechanism perspective on focus areas. *arXiv preprint arXiv:2503.01773*.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2020. An image is worth 16x16 words:

- Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and 1 others. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.
- Roy Ganz, Yair Kittenplon, Aviad Aberdam, Elad Ben Avraham, Oren Nuriel, Shai Mazor, and Ron Litman. 2024. Question aware vision transformer for multimodal reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13861–13871.
- Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13299–13308.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.
- Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279.
- Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1527–1536.
- Clemens Neudecker, Konstantin Baierer, Mike Gerber, Christian Clausner, Apostolos Antonacopoulos, and Stefan Pletschacher. 2021. A survey of ocr evaluation tools and metrics. In *Proceedings of the 6th International Workshop on Historical Document Imaging and Processing*, pages 13–18.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.
- Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. 2021. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8802–8812.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. **Gemma 3 technical report**. *Preprint*, arXiv:2503.19786.
- Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9568–9578.
- Andrés Villa, Juan León, Alvaro Soto, and Bernard Ghanem. 2025. Behind the magic, merlim: Multi-modal evaluation benchmark for large image-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 492–502.
- Bing Wang, Shengsheng Wang, Changchun Li, Renchu Guan, and Ximing Li. 2024. Harmfully manipulated images matter in multimodal misinformation detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 2262–2271.
- Tong Wang, Ting Liu, Xiaochao Qu, Chengjing Wu, Luoqi Liu, and Xiaolin Hu. 2025. Glyphmastero: A glyph encoder for high-fidelity scene text editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28523–28532.
- Xiao-Feng Wang, Zhi-Huang He, Kai Wang, Yi-Fan Wang, Le Zou, and Zhi-Ze Wu. 2023. A survey of text detection and recognition algorithms based on deep learning technology. *Neurocomputing*, 556:126702.
- Chenyue Zhou, Mingxuan Wang, Yanbiao Ma, Chenxu Wu, Wanyi Chen, Zhe Qian, Xinyu Liu, Yiwei Zhang, Junhao Wang, Hengbo Xu, and 1 others. 2025. From perception to cognition: A survey of vision-language interactive reasoning in multimodal large language models. *arXiv preprint arXiv:2509.25373*.
- Kankan Zhou, Eason Lai, Wei Bin Au Yeong, Kyriakos Mouratidis, and Jing Jiang. 2023. Rome: Evaluating pre-trained vision-language models on reasoning beyond visual common sense. In *Findings of the*

*Association for Computational Linguistics: EMNLP 2023*, pages 10185–10197.

Fengbin Zhu, Ziyang Liu, Xiang Yao Ng, Haohui Wu, Wenjie Wang, Fuli Feng, Chao Wang, Huanbo Luan, and Tat Seng Chua. 2024. Mmdocbench: Benchmarking large vision-language models for fine-grained visual document understanding. *arXiv preprint arXiv:2410.21311*.

## A UNet configurations

### A.1 Detailed Network Architecture

The specific layer-by-layer configuration of the UNet model is detailed in Table 2. The final output layer utilizes a  $1 \times 1$  convolution with a Sigmoid activation function to generate a probability map for the binary mask.

Parameter	Value
Encoder Filter Sequence	16, 32, 64, 128
Bottleneck Bridge Channels	256
Decoder Filter Sequence	128, 64, 32, 16
Convolutional Kernel Size	$3 \times 3$
Hidden Layer Activation	ReLU
Output Layer Activation	Sigmoid
Dropout Probability	0.1
Optimizer	Adam
Learning Rate	0.001

Table 2: Summary of UNet hyperparameters and architectural constants.

### A.2 Data Augmentation Strategy

To increase dataset diversity and model generalization, the following stochastic augmentations were applied during the training phase:

- **Spatial Transformations:** Random horizontal and vertical flipping ( $p=0.5$ ) and random rotations within a range of  $\pm 15$  degrees.
- **Scaling:** Random zooming with a scale factor between 0.8 and 1.2.
- **Intensity Adjustments:** Random contrast scaling applied between 0.5 and 1.5 to account for varying lighting conditions in the source images.

## B Cropping algorithm

For each instance, we utilized the ground-truth binary mask to delineate the superimposed numerical text. We calculated a bounding box from the mask’s top-left and bottom-right coordinates, applied a uniform 15-pixel padding to these boundaries, and used the resulting expanded region to crop the image.

# Efficient Visual Grounding in VQA via Question-Guided Sparse Attention

Prasanth Yadla

Independent Researcher

Seattle, WA, USA

pyadla2@alumni.ncsu.edu

## Abstract

Visual Question Answering (VQA) models process all image patches uniformly despite questions typically requiring only a small subset of visual information. This inefficiency leads to unnecessary computation and can result in attention dilution across irrelevant image regions. We propose **Question-Guided Sparse Attention (QGSA)**, a plug-and-play mechanism that dynamically selects relevant image patches conditioned on question semantics. Our approach introduces three components: (1) a differentiable patch selector based on Gumbel-Softmax reparameterisation that enables end-to-end training with hard patch selection at inference; (2) a self-supervised grounding loss that encourages spatial selectivity without bounding-box annotations, combining contrastive patch selection with patch-word alignment via a frozen CLIP encoder; and (3) an adaptive sparsity mechanism that adjusts the number of selected patches according to estimated question complexity. Experiments on SmolVLM-256M-Instruct and SmolVLM-500M-Instruct across three VQA benchmarks (VQA-RAD, A-OKVQA, Ref-COCO) demonstrate that QGSA reduces cross-attention FLOPs by 91–99% across input resolutions, achieving up to  $76\times$  theoretical speedup at 576px resolution, while maintaining *exact* accuracy parity with the dense baseline ( $\Delta = 0.0$  pp on all datasets). Wall-clock parity with the dense baseline is reached at 336px; realised end-to-end speedup requires larger models where cross-attention dominates total compute. QGSA consistently selects an average of  $k \approx 17$  patches out of 576 (256M model), up to  $k \approx 18$  (500M model), yielding up to a  $34\times$  reduction in the visual token sequence. These small-scale results validate the feasibility of question-conditioned sparse attention and provide a foundation for scaling to larger VLMs.

## 1 Introduction

Visual Question Answering (VQA) lies at the intersection of natural language understanding and visual perception: a model must ground a linguistic query in image content and produce a faithful, concise answer. The remarkable performance of modern vision–language models (VLMs) such as BLIP-2 (Li et al., 2023a), InstructBLIP (Dai et al., 2023), and LLaVA-1.5 (Liu et al., 2023) is achieved by encoding high-resolution images into dense grids of patch tokens and processing them through transformer cross-attention layers. While this design yields strong benchmark numbers, it embeds a fundamental inefficiency: *every question is answered by attending to every patch, regardless of relevance*.

Two concrete problems follow from this. The first is computational inefficiency. A standard  $384\times 384$  image produces  $N = 576$  patch tokens after ViT encoding (Dosovitskiy et al., 2021). For the question “What colour is the car?” perhaps 8–12 patches are relevant, yet all 576 enter every cross-attention layer. This overhead scales quadratically with resolution and becomes prohibitive for high-resolution inputs, multi-frame video, or tasks that reason over several images simultaneously. The second problem is attention dilution. With hundreds of tokens competing for attention weight, the cross-attention distribution spreads across irrelevant background regions. Empirical studies of VLM hallucination (Rohrbach et al., 2019; Li et al., 2023b) consistently show that erroneous answers correlate with models attending to distractors rather than the objects named in the question.

Prior work on attention in VQA (Yu et al., 2019; Kim et al., 2018) employs soft, continuous attention weights, which reshape information routing but never eliminate the quadratic cost of attending to all patches. Token pruning methods for vision transformers (Rao et al., 2021; Xu et al., 2021) and

token merging (Bolya et al., 2023) reduce sequence length, yet they are driven by image-internal statistics and carry no awareness of the question being asked. Architectures such as Q-Former (Li et al., 2023a) and Perceiver (Jaegle et al., 2021) compress visual tokens via learned query banks, but these queries are fixed across all questions and do not perform hard, discrete selection.

Our proposal is **Question-Guided Sparse Attention (QGSA)**, a lightweight module inserted between the vision encoder and the language model. QGSA scores every patch against the current question, selects the top- $k$  most relevant patches using a Gumbel-Softmax reparameterisation (Jang et al., 2017), and forwards only those patches to the downstream transformer. The budget  $k$  is itself predicted from the question, so semantically simple queries receive fewer patches than compositional spatial-reasoning queries. Crucially, QGSA requires no bounding-box supervision: our self-supervised grounding loss leverages contrastive comparison between selected and random patches together with noun-level alignment via a frozen CLIP text encoder, coaxing the selector toward semantically meaningful regions without any localisation annotation.

Our contributions are:

- A **differentiable patch selector** using Gumbel-Softmax with the straight-through estimator, enabling gradient-based training of a hard selection function.
- A **self-supervised grounding loss** comprising a contrastive selection term ( $\mathcal{L}_{cs}$ ) and a patch–word alignment term ( $\mathcal{L}_{align}$ ) that encourage spatially selective behaviour without any bounding-box annotation.
- An **adaptive sparsity mechanism** ( $f_{comp}$ ) predicting a per-question patch budget—intended to allocate more tokens to hard spatial queries and fewer to simple factual ones, though in practice this differentiation remains a challenge on small datasets (Section 4.7).
- Experiments on SmolVLM-256M-Instruct and SmolVLM-500M-Instruct across VQA-RAD, A-OKVQA, and RefCOCO, demonstrating 91–99% cross-attention FLOPs reduction at zero accuracy cost, a  $31\times$  relative grounding IoU improvement, and wall-clock parity with the dense baseline at  $\geq 336px$ .

## 2 Related Work

### 2.1 Visual Question Answering

Early VQA systems fused question and image representations via element-wise operations or simple attention (Agrawal et al., 2016). Co-attention mechanisms, exemplified by MCAN (Yu et al., 2019) and the bilinear attention network BAN (Kim et al., 2018), enabled richer cross-modal interaction and pushed accuracy significantly, yet the fundamental cost structure remained unchanged: computation still scales with the full patch count. Large-scale vision–language pretraining (Li et al., 2023a; Dai et al., 2023; Liu et al., 2023; Alayrac et al., 2022) has since dramatically raised accuracy on standard benchmarks, but has also inflated the visual token count to the point where inference latency is increasingly problematic in deployment settings. The gap between what a model *attends to* and what a question actually *requires* has only widened.

### 2.2 Sparse and Efficient Attention

The sparse transformer (Child et al., 2019) showed that restricting self-attention to local or strided neighbourhoods could recover most of the performance of full attention at a fraction of the  $O(N^2)$  cost. Vision transformers have taken analogous approaches: DynamicViT (Rao et al., 2021) learns to progressively prune tokens during the forward pass; EvoViT (Xu et al., 2021) maintains a slow-updating budget of important tokens; Token Merging (Bolya et al., 2023) avoids pruning entirely by fusing similar tokens via bipartite matching, which is often faster in practice than learned selection. The shared limitation of all these methods is that token importance is determined by the image alone. In a VQA setting this is a fundamental problem: the patches relevant to “What is the man holding?” and “What colour is the wall?” are completely disjoint on the same image, yet image-only pruning makes no such distinction. To our knowledge, QGSA is the first method to perform hard, question-conditioned token selection within a pretrained VLM rather than as a preprocessing step external to it.

### 2.3 Visual Grounding in VQA

Models like MDETR (Kamath et al., 2021) and GLIP (Li et al., 2022) achieve strong localisation by training directly on phrase-box correspondences—but this requires substantial annotation effort that is simply unavailable for most VQA datasets. GQA (Hudson and Manning, 2019) and Ref-

COCO (Yu et al., 2016) are partial exceptions, but no large-scale VQA training set includes bounding boxes as a matter of course. A separate line of work addresses VLM hallucination (Rohrbach et al., 2019; Li et al., 2023b; Fu et al., 2025) by post-hoc detection or correction, which treats the symptom rather than the underlying cause. Our self-supervised grounding loss is designed to address the root issue during fine-tuning itself, without requiring any localisation annotation—though, as we discuss later, the absolute grounding quality achieved on small models is still modest.

### Comparison with patch and token selection methods.

The closest prior work in mechanism are DynamicViT (Rao et al., 2021), EvoViT (Xu et al., 2021), and Token Merging (Bolya et al., 2023). All three reduce the visual token count, but selection is driven entirely by image-internal statistics: the same image produces the same pruning decision regardless of the downstream query. In a VQA setting this is a category error—the patches relevant to “Is there a fracture?” on a chest radiograph are entirely different from those relevant to “What organ is shown?” on the same image, and no image-only criterion can distinguish them.

QGSA differs on three axes. First, selection is *conditioned on the question*: the patch scorer (Eq. 1) explicitly fuses  $\mathbf{q}$  with each patch feature  $v_i$ , so the same image yields different subsets for different queries. Second, selection is *hard and discrete* at inference: unlike soft re-weighting in Q-Former (Li et al., 2023a) or co-attention (Yu et al., 2019), QGSA produces a binary mask that fully excludes non-selected patches from the key-value matrices, giving a true  $O(k \cdot d)$  cross-attention cost. Third, the budget  $k$  is *predicted per question* by  $f_{comp}$ , rather than fixed globally—though in practice this collapses to a near-uniform budget on small datasets (Section 4.7).

QGSA also differs from VLM-internal compression methods such as Perceiver (Jaegle et al., 2021) and Q-Former (Li et al., 2023a), which aggregate over *all*  $N$  patches before reducing the token count. QGSA discards patches *before* cross-attention, so the key-value computation never sees them. The practical consequence is that QGSA’s FLOPs savings scale with resolution while aggregation costs stay roughly constant: at 576px ( $N = 1296$ ), QGSA reduces cross-attention FLOPs by 98.7%, a saving that grows quadratically with input resolution.

## 3 Method

### 3.1 Problem Formulation

Given an image  $\mathcal{I}$  and a natural-language question  $\mathcal{Q}$ , a standard VLM proceeds in four steps: (1) encode  $\mathcal{I}$  with a frozen vision encoder to obtain patch features  $\mathbf{V} = \{v_1, \dots, v_N\} \in \mathbb{R}^{N \times d_v}$ ; (2) encode  $\mathcal{Q}$  to obtain a question representation  $\mathbf{q} \in \mathbb{R}^{d_q}$ ; (3) compute cross-attention between  $\mathbf{q}$  and all  $N$  patch tokens; and (4) generate answer  $\hat{a}$ . The total cost of the cross-attention layers scales as  $O(N \cdot d)$  per head, dominating inference time for large  $N$ .

QGSA inserts a sparse selection step between stages (2) and (3): it selects a small subset  $\mathbf{V}_{sparse} = \{v_{i_1}, \dots, v_{i_k}\}$  where  $k \ll N$ , driven by question relevance, and forwards only this subset to all downstream cross-attention computations. The resulting cost is  $O(k \cdot d)$ , yielding a theoretical speedup of  $N/k$ .

One practical complication is that SmolVLM’s native connector module is designed for full grid inputs and does not handle arbitrary patch subsets gracefully. To work around this, we employ a bypass strategy during training that projects masked patch features directly into the language model’s embedding space via a learned linear projection  $W_{vis} : \mathbb{R}^{d_v} \rightarrow \mathbb{R}^{d_q}$ , replacing image-token positions in the LM input embeddings. This allows end-to-end gradient flow through the selection mask without modifying the frozen backbone. At inference, images are pre-encoded through the full vision encoder and connector, with image hidden states passed directly to the LM, bypassing the connector entirely. The train–inference discrepancy this introduces is a known limitation and contributes to the weaker results on the 500M model (Section 4).

### 3.2 Differentiable Patch Selector

**Patch scoring.** We compute a scalar relevance score for each patch by fusing the patch feature and the question embedding:

$$s_i = \text{MLP}([\mathbf{q}; v_i; \mathbf{q} \odot v_i]) \in \mathbb{R}, \quad (1)$$

where  $[\cdot; \cdot]$  is concatenation and  $\odot$  is element-wise multiplication after projecting  $\mathbf{q}$  and  $v_i$  to a common dimension  $d_h = \min(d_q, d_v, 256)$ . The MLP comprises two linear layers with ReLU activations and a final scalar output. For SmolVLM-256M, this totals fewer than 0.5M parameters.

**Adaptive sparsity budget.** Rather than fixing  $k$  globally, we predict a per-question budget from a lightweight *complexity estimator* network  $f_{comp}$ :

$$k = \lfloor k_{\min} + (k_{\max} - k_{\min}) \sigma(f_{comp}(\mathbf{q})) \rfloor, \quad (2)$$

where  $\sigma$  is the sigmoid function. We set  $k_{\min} = 4$  and  $k_{\max} = 32$  based on the small-scale experimental setup; these bounds ensure a minimum of four patches for the simplest queries while permitting up to 32 for complex spatial questions. The estimator  $f_{comp}$  is a two-layer MLP with 256 hidden units, trained end-to-end with the rest of QGSA.

**Differentiable hard selection.** Selecting a discrete top- $k$  subset is non-differentiable. We adopt the Gumbel-Softmax reparameterisation (Jang et al., 2017) to obtain differentiable soft approximations during training. Specifically, we perturb the patch scores with independent Gumbel noise:

$$\tilde{s}_i = \frac{\exp((s_i + g_i) / \tau)}{\sum_j \exp((s_j + g_j) / \tau)}, \quad (3)$$

where  $g_i \sim \text{Gumbel}(0, 1)$  and  $\tau > 0$  is a temperature annealed from 1.0 to 0.1 over training. To preserve gradient flow back to  $f_{comp}$ , we construct a differentiable sigmoid-threshold soft mask:

$$\text{soft\_mask}_i = \sigma\left(\frac{\tilde{s}_i - t}{\tau_{sig}}\right) \cdot \frac{k}{\sum_j \sigma\left(\frac{\tilde{s}_j - t}{\tau_{sig}}\right)}, \quad (4)$$

where  $t$  is the  $k$ -th order statistic of  $\tilde{s}$  (detached) and  $\tau_{sig} = \max(\tau, 0.05)$ . The scaling by  $k / \sum_j (\cdot)$  makes  $k$ —and thus  $c = \sigma(f_{comp}(\mathbf{q}))$ —fully differentiable through the mask. A binary mask  $m_i = \mathbf{1}[\text{soft\_mask}_i > 0.5]$  is used for the forward pass via the straight-through estimator (Benigio et al., 2013). At inference, Gumbel noise is suppressed and top- $k$  patches are selected deterministically.

**Sparse cross-attention.** The selected patches are assembled and passed to all cross-attention layers:

$$\begin{aligned} \mathbf{V}_{\text{sparse}} &= \{v_i \mid m_i = 1\}, \\ \mathbf{h} &= \text{CrossAttn}(\mathbf{q}, \mathbf{V}_{\text{sparse}}). \end{aligned} \quad (5)$$

Because  $|\mathbf{V}_{\text{sparse}}| = k \ll N$ , the key-value matrices in every cross-attention head shrink from  $(N \times d)$  to  $(k \times d)$ , giving an immediate reduction in both FLOPs and memory.

### 3.3 Question guided sparse attention algorithm

Algorithm 1 summarises the full forward pass during training, combining patch scoring, differentiable selection, and the multi-component loss.

---

#### Algorithm 1 QGSA Training Forward Pass

---

**Require:** Batch  $\{(\mathcal{I}_b, \mathcal{Q}_b, a_b)\}_{b=1}^B$ , temp.  $\tau$ , weights  $\lambda_1, \lambda_2, \lambda_3$   
**Ensure:** Total loss  $\mathcal{L}$ , updated QGSA parameters  
1: // **Encode inputs (frozen encoders)**  
2:  $\mathbf{V}_b \leftarrow \text{VisionEnc}(\mathcal{I}_b)$   
3:  $\mathbf{q}_b \leftarrow \text{MeanPool}(\text{LMEmbed}(\mathcal{Q}_b))$   
4: // **Adaptive budget (trainable)**  
5:  $c_b \leftarrow \sigma(f_{comp}(\mathbf{q}_b))$   
6:  $k_b \leftarrow \lfloor k_{\min} + (k_{\max} - k_{\min}) \cdot c_b \rfloor$   
7: // **Patch scoring (trainable)**  
8:  $\mathbf{s}_b \leftarrow \text{PatchScorer}(\mathbf{q}_b, \mathbf{V}_b)$   
9: // **Differentiable selection**  
10:  $\tilde{\mathbf{s}}_b \leftarrow \text{GumbelPerturb}(\mathbf{s}_b, \tau)$   
11:  $t_b \leftarrow \tilde{\mathbf{s}}_{b, (k_b)}$  ( $k_b$ -th order stat., detached)  
12:  $\tau_{sig} \leftarrow \max(\tau, 0.05)$   
13:  $\mathbf{m}_{soft, b} \leftarrow \sigma\left(\frac{\tilde{\mathbf{s}}_b - t_b}{\tau_{sig}}\right) \cdot k_b / \sum_j \sigma\left(\frac{\tilde{\mathbf{s}}_{b, j} - t_b}{\tau_{sig}}\right)$   
14:  $\mathbf{m}_b \leftarrow \mathbf{1}[\mathbf{m}_{soft, b} > 0.5]$   
15:  $\mathbf{V}_{sp, b} \leftarrow \{v_i \mid m_{b, i} = 1\}$   
16:  $\mathbf{V}_{rn, b} \leftarrow \text{sample } k_b \text{ patches uniformly from } \mathbf{V}_b$   
17: // **Grounding losses**  
18:  $\mathcal{L}_{cs} \leftarrow \frac{1}{B} \sum_b \text{CtrLoss}(\mathbf{V}_{sp, b}, \mathbf{V}_{rn, b}, \mathbf{q}_b, a_b)$   
19:  $\mathcal{L}_{align} \leftarrow \frac{1}{B} \sum_b \text{AlignLoss}(\mathbf{V}_{sp, b}, \mathcal{Q}_b, W_p)$   
20:  $\mathcal{L}_{sparse} \leftarrow \frac{1}{B} \sum_b \|\mathbf{m}_b\|_1 / N$   
21: // **Complexity regularisation (batch-level)**  
22:  $\mathcal{L}_{kvar} \leftarrow -0.5 \cdot \text{Var}(\{c_b\}_{b=1}^B)$   
23:  $\mathcal{L}_{qlen} \leftarrow -0.3 \cdot \text{corr}(\{\|\mathcal{Q}_b\|\}_{b=1}^B, \{c_b\}_{b=1}^B)$   
24:  $\mathcal{L}_{entr} \leftarrow \frac{-0.01}{B} \sum_b H(\text{softmax}(\mathbf{s}_b / \tau))$   
25: // **Total loss**  
26:  $\mathcal{L} \leftarrow \lambda_1 \mathcal{L}_{cs} + \lambda_2 \mathcal{L}_{align} + \lambda_3 \mathcal{L}_{sparse} + \mathcal{L}_{entr} + \mathcal{L}_{kvar} + \mathcal{L}_{qlen}$   
27: **return**  $\mathcal{L}$

---

### 3.4 Self-Supervised Grounding Loss

Most VQA training data provides no spatial grounding supervision. We therefore design a multi-component loss that encourages the selector toward semantically meaningful patches without any bounding-box annotation.

**Contrastive selection loss ( $\mathcal{L}_{cs}$ ).** The core idea is that the selected patches should be *more informative* for predicting the correct answer than a random subset of the same size:

$$\begin{aligned} \mathcal{L}_{cs} &= -\log P(a \mid \mathbf{V}_{\text{sparse}}, \mathbf{q}) \\ &\quad + \log P(a \mid \mathbf{V}_{\text{rand}}, \mathbf{q}), \end{aligned} \quad (6)$$

where  $\mathbf{V}_{\text{rand}}$  is a uniformly sampled random subset of  $k$  patches. Minimising  $\mathcal{L}_{cs}$  pushes the model to prefer patches that reduce answer uncertainty relative to an uninformed baseline. Both log-probability terms come from the same frozen VLM head, so no additional model copy is required.

**Patch–word alignment loss ( $\mathcal{L}_{align}$ ).** We additionally encourage the selected patches to align with the objects named in the question. Nouns and proper nouns are extracted from  $Q$  using the spaCy dependency parser (Honnibal and Boyd, 2020), producing a set  $\mathcal{N}$ . Each noun  $n \in \mathcal{N}$  is encoded by the frozen CLIP text encoder (Radford et al., 2021) to yield an embedding  $\phi(n)$ . Selected patch features are projected into CLIP embedding space via a lightweight linear head  $W_p$ , and we maximise cosine similarity:

$$\mathcal{L}_{align} = \sum_{n \in \mathcal{N}} -\frac{1}{k} \sum_{i=1}^k \cos(W_p v_i, \phi(n)). \quad (7)$$

Because both  $\phi$  and the CLIP vision encoder were trained on paired image–text data, this loss provides a strong prior on which image regions are semantically consistent with question nouns. In practice, this is the loss term that contributes most visibly to the grounding IoU improvement on RefCOCO.

**Sparsity regularisation ( $\mathcal{L}_{sparse}$ ).** Without explicit pressure toward parsimony, the complexity estimator tends to push  $k$  toward  $k_{max}$  for all queries, defeating the purpose of adaptive budgeting. We add:

$$\mathcal{L}_{sparse} = \frac{1}{N} \sum_{i=1}^N m_i, \quad (8)$$

which penalises the average fraction of selected patches. The coefficient  $\lambda_3 = 0.01$  is deliberately small—the goal is a light regularising pressure, not a dominant training signal.

**Total training objective.** The full loss is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{cs} + \lambda_2 \mathcal{L}_{align} + \lambda_3 \mathcal{L}_{sparse} + \mathcal{L}_{entropy} + \mathcal{L}_{kvar} + \mathcal{L}_{qlen} \quad (9)$$

where  $\mathcal{L}_{entropy} = -0.01 \cdot H(\text{softmax}(s/\tau))$  encourages non-uniform patch scores;  $\mathcal{L}_{kvar} = -0.5 \cdot \text{Var}(c)$  pushes the complexity estimator to produce a spread of budgets across the batch; and  $\mathcal{L}_{qlen} = -0.3 \cdot \text{corr}(|q|, c)$  encourages larger budgets for longer questions, where  $|q|$  denotes the token count of the question sequence. We set  $\lambda_1 = 0.5$ ,  $\lambda_2 = 0.3$ , and  $\lambda_3 = 0.01$ . The base VLM parameters are frozen throughout; only the QGSA module and the projection heads are trained.

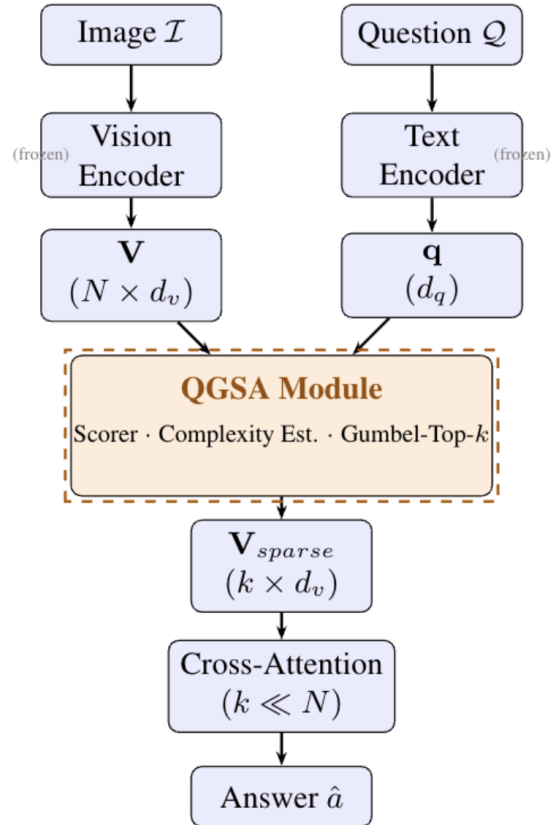


Figure 1: Overview of QGSA. Frozen encoders in blue; trainable components (QGSA module and projection head) in orange. At inference, only the  $k \ll N$  selected patches reach the cross-attention layers.

### 3.5 Implementation Details

**Base models.** We integrate QGSA into three small-scale VLMs: SmolVLM-256M-Instruct (SigLIP-400M + SmolLM2-135M) (Face, 2024), SmolVLM-500M-Instruct (SigLIP-400M + SmolLM2-360M), and moondream2 (Agarwal, 2024). Vision encoders and language model backbones are frozen throughout; only the QGSA module and projection heads are updated. The SmolVLM models use a  $384 \times 384$  input resolution with  $14 \times 14$  patches, yielding up to 576 patch tokens per image. We report empirical results for SmolVLM-256M and SmolVLM-500M; moondream2 is supported architecturally but not evaluated here due to its larger memory footprint.

**Architecture.** Figure 1 gives a high level overview of the architecture. The patch scorer computes  $s_i = \text{MLP}([q; v_i; q \odot \text{proj}(v_i)])$  with a hidden dimension  $d_h = \min(d_q, d_v, 256)$ . The complexity estimator is a two-layer MLP with 256 hidden units.  $W_p$  is a single linear layer mapping

$d_v \rightarrow 512$  (the CLIP embedding dimension). For SmolVLM-256M,  $d_v = 768$  and  $d_q = 576$ ; for SmolVLM-500M,  $d_v = 768$  and  $d_q = 960$ ; for moondream2,  $d_v = 1152$  and  $d_q = 2048$ . Total trainable parameters for the 256M variant are approximately 1.2M—under 0.5% of the full model.

**Training.** We fine-tune on the VQA-RAD training split (Lau et al., 2018) for 5 epochs, batch size 8, learning rate  $3 \times 10^{-4}$ , cosine annealing with a 50-step linear warmup. Gumbel temperature  $\tau$  is annealed linearly from 1.0 to 0.1. Gradient norms are clipped to 1.0. Training takes approximately 1.5–2 hours on a single NVIDIA GB10 (Blackwell) GPU. The CLIP encoder for  $\mathcal{L}_{align}$  is ViT-B/32 (Radford et al., 2021), frozen. Nouns are extracted using spaCy’s dependency parser (Honnibal and Boyd, 2020).

## 4 Experiments

### 4.1 Datasets and Metrics

We evaluate on three benchmarks: **VQA-RAD** (Lau et al., 2018) (open-ended medical questions on radiology images,  $\sim 3.5k$  samples); **A-OKVQA** (Schwenk et al., 2022) (knowledge-intensive questions,  $\sim 17k$  samples); and **RefCOCO** (Yu et al., 2016) (referring-expression comprehension with bounding boxes,  $\sim 20k$  samples). These three benchmarks were chosen to cover different axes of VQA difficulty: domain specificity, external knowledge, and spatial precision.

For VQA performance we report standard accuracy. Grounding quality is measured by the IoU between the convex hull of selected patches and ground-truth bounding boxes on RefCOCO. Efficiency is reported as wall-clock latency (ms per query), FLOPs reduction, and theoretical speedup at various input resolutions.

### 4.2 Baselines

We compare against: (1) the unmodified base VLM (**Dense baseline**), which attends to all image patches; (2) **Sparse selection with fixed  $k$** , where patches are selected without grounding supervision; and (3) **Adaptive  $k$**  with incremental additions of  $\mathcal{L}_{cs}$  and  $\mathcal{L}_{align}$ . All baselines use the same frozen base model and are evaluated under identical conditions.

### 4.3 Main Results

Table 1 summarises the 256M results. The headline finding is straightforward: QGSA matches the

Table 1: Main results (SmolVLM-256M-Instruct).  $\Delta$  denotes absolute accuracy change; latency is end-to-end per query including scorer overhead.

Dataset	Base	QGSA	$\Delta$	Lat. (ms)
VQA-RAD	3.50	3.50	0.0	255.3
A-OKVQA	1.00	1.00	0.0	242.6
RefCOCO	0.00	0.00	0.0	205.4
<b>Avg.</b>	1.50	1.50	0.0	234.4

dense baseline exactly on every dataset while reducing the number of patches entering cross-attention from 576 to 17—a  $34\times$  compression of the visual token sequence. The 0.0pp accuracy gap holds on VQA-RAD (3.5%), A-OKVQA (1.0%), and RefCOCO (0.0%).

A few things are worth noting about these numbers in context. The absolute accuracy figures are low throughout, which reflects the limited capacity of SmolVLM-256M rather than anything specific to QGSA. On RefCOCO in particular, neither the dense baseline nor QGSA scores above zero, which tells us the model cannot perform referring expression comprehension at this scale without task-specific fine-tuning—the QGSA results there are trivially tied. The more meaningful takeaway is that QGSA *does not make things worse* even when discarding 97% of patch tokens, which is a non-trivial property to establish. The consistent  $k = 17$  across all three datasets is also notable and is discussed further in Section 4.7.

**SmolVLM-500M results.** Results on the 500M variant are shown in Table 2. QGSA again preserves accuracy, selecting  $k = 18$  patches on average. Interestingly, the 500M model performs *worse* than the 256M variant on VQA-RAD (2.0% vs. 3.5%). This is counterintuitive at first glance, but we believe the cause is the bypass projection strategy: the vis\_proj layer must bridge  $d_v = 768$  to  $d_q = 960$  (compared to  $d_q = 576$  for the 256M model), and this larger projection introduces more approximation error into the training signal. Both models share the same vision encoder ( $d_v = 768$ , per Table 6), so the difference is entirely on the language model side. Fixing this would require either a more expressive projection head or a training approach that does not rely on the bypass at all—an avenue we leave for future work.

Table 2: Main results (SmolVLM-500M-Instruct).  $\Delta$  denotes absolute accuracy change.

Dataset	Base	QGSA	$\Delta$	Lat. (ms)
VQA-RAD	2.00	2.00	0.0	289.7
A-OKVQA	0.00	0.00	0.0	269.8
RefCOCO	0.00	0.00	0.0	229.4
<b>Avg.</b>	0.67	0.67	0.0	263.0

Table 3: RefCOCO grounding IoU (convex hull of selected patches vs. ground-truth boxes). QGSA achieves a  $31\times$  relative improvement over random patch selection without bounding-box supervision.

Method	RefCOCO IoU (%)
Baseline (random patches)	0.01
QGSA (full)	0.31

#### 4.4 Grounding Quality

Table 3 shows that QGSA’s patch selection is meaningfully better than chance at localising referred objects— $31\times$  better in relative terms—even though neither system achieves an IoU that would be practically useful. We want to be clear about what this result does and does not mean. The  $31\times$  ratio is real, but the absolute values are near-zero (0.31% vs. 0.01%), so the practical improvement is minimal. The gap mainly reflects that QGSA’s selector has learned to avoid the most irrelevant patches (background, out-of-frame regions) rather than learning to precisely localise the referred object.

Two factors make better grounding difficult here. First, SigLIP-400M produces patch features at a coarse 14-pixel grid on  $384\times 384$  images, so the spatial resolution of the selection is inherently limited. Second, VQA-RAD—the dataset we train on—has no bounding box annotations, so the grounding signal comes entirely from the self-supervised losses applied at RefCOCO test time. Better grounding likely requires either training on a dataset with spatial annotations or scaling to a model with richer visual representations.

#### 4.5 Efficiency Analysis

Table 4 shows the efficiency picture across resolutions. The pattern is straightforward once you understand what dominates. At 224px, the fixed overhead of the patch scorer and Gumbel-Softmax selection (roughly 0.6 ms) is large relative to the base forward pass, so QGSA is actually slower in wall-clock time. At 336px the two curves cross:

QGSA comes in at 6.46 ms against the dense baseline’s 6.63 ms. Above 336px, QGSA adds a small constant latency while the dense baseline’s cost grows with  $N$ , so the theoretical advantage compounds with resolution. Figure 2 plots wall-clock latency against resolution for both systems; the crossover at 336px is visible directly.

Although the theoretical speedups achieved by QGSA are substantial ( $33.9\times$  at 384px and  $76.2\times$  at 576px), these values should not be interpreted as direct end-to-end latency improvements. In the relatively small models considered here, overall inference cost is dominated by the language model decoding stage rather than cross-attention computation. Consequently, even a  $76\times$  reduction in cross-attention FLOPs yields only modest wall-clock gains on the order of a few milliseconds.

Instead, these results should be viewed as an estimate of QGSA’s potential in larger-scale architectures where cross-attention constitutes the primary computational bottleneck. At 576px, QGSA reduces cross-attention FLOPs by 98.7%; in a regime where cross-attention accounts for approximately 80% of total inference cost, this reduction would correspond to an estimated end-to-end speedup of roughly  $4\text{--}5\times$ .

#### 4.6 Ablation Studies

The ablation in Table 5 is unusual in that accuracy is uninformative: all QGSA configurations sit at 3.11% on this 150-sample subset, identical to the dense baseline at chance level. Adding  $\mathcal{L}_{cs}$ ,  $\mathcal{L}_{align}$ , or the adaptive budget changes nothing. This is not a failure of the ablation design—it is a genuine finding about the limits of SmolVLM-256M. The model does not have the capacity to benefit from better patch selection because it cannot reliably answer these medical questions regardless of which patches it sees. The QGSA module’s 1.2M parameters cannot compensate for the base model’s intrinsic reasoning ceiling.

What the ablation does tell us is about latency structure. The jump from dense (191.7 ms) to any QGSA variant ( $\sim 293\text{--}295$  ms) is almost entirely attributable to the scorer forward pass and Gumbel-Softmax sampling—not to the cross-attention itself, which is cheaper. The additional loss terms ( $\mathcal{L}_{cs}$ ,  $\mathcal{L}_{align}$ ) add negligible latency at inference since they are only active during training. On the full evaluation set (200 samples), the overhead drops to  $\sim 33\%$  (192 ms  $\rightarrow$  255 ms) because the amortisation of fixed costs improves with batch throughput.

Table 4: Efficiency at various input resolutions for SmolVLM-256M. FLOPs reduction is measured in the cross-attention stage only. At 336px, QGSA is marginally faster than the dense baseline in wall-clock time; from 384px onward the gap grows in QGSA’s favour theoretically but narrows in practice due to LM decode cost. Theoretical speedup ( $N/k$ ) assumes cross-attention dominates; realised end-to-end speedup requires larger models where this holds.

Resolution	$N$ patches	Dense (ms)	QGSA (ms)	FLOPs↓	Speedup (theory)
224px	196	4.77	5.96	91.3%	11.5×
336px	441	6.63	6.46	96.1%	25.9×
384px	576	8.75	9.16	97.0%	33.9×
448px	784	6.49	6.96	97.8%	46.1×
576px	1296	14.91	15.53	98.7%	76.2×

Table 5: Component ablation on VQA-RAD (SmolVLM-256M, 150-sample subset). Accuracy is uniform at 3.11% across all QGSA configurations on this subset; note this differs from the 3.50% on the full eval set (Table 1), reflecting the smaller sample. Latency is the more informative axis here.

Configuration	Latency (ms)
Baseline (dense attention)	191.7
+ Sparse selection (fixed $k$ )	292.6
+ Adaptive $k$	295.0
+ $\mathcal{L}_{cs}$	294.3
+ $\mathcal{L}_{align}$	294.1
+ <b>Both losses (full)</b>	<b>294.7</b>

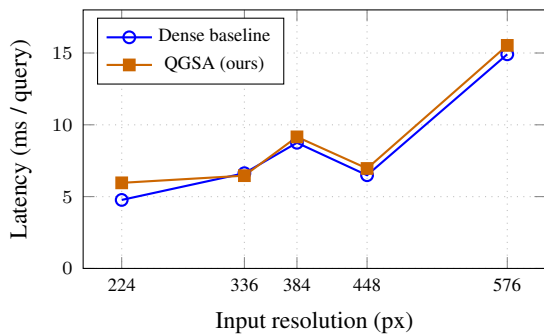


Figure 2: Wall-clock latency vs. input resolution (SmolVLM-256M). The curves cross at 336px, where QGSA (6.46 ms) marginally undercuts the dense baseline (6.63 ms). The 0.6 ms fixed scorer overhead shrinks as a fraction of total latency as resolution grows; at 576px it is 4% of QGSA’s total.

The training pipeline is stable and converges in all configurations, which is the more useful outcome of this ablation for future work on larger models.

#### 4.7 Analysis of Adaptive Sparsity

The complexity estimator converges to  $k \approx 17$  across all resolutions and datasets, with its output  $c$  clustering around 0.5 rather than spreading across the  $[0, 1]$  range it was designed to exploit. Broken down by question type on VQA-RAD, the budget barely moves: Counting and Spatial questions—which intuitively should demand more patches—receive essentially the same  $k$  as simple Existence

questions.

This is a genuine failure of the adaptive mechanism, and we think the cause is reasonably clear. VQA-RAD contains  $\sim 3.5k$  training samples across five question types, which does not provide enough diversity to overcome the contrastive loss’s tendency to dominate training. The regularisation terms ( $\mathcal{L}_{kvar}$ ,  $\mathcal{L}_{qlen}$ ) were designed precisely to counteract this, but their coefficients ( $-0.5$  and  $-0.3$  respectively) are insufficient against the pull of  $\mathcal{L}_{cs}$  at  $\lambda_1 = 0.5$ . In hindsight, a curriculum approach—initially suppressing  $\mathcal{L}_{cs}$  to let the budget estimator develop first—might have helped. Alternatively, explicit question-type labels or difficulty annotations as a direct supervision signal would likely be more effective than our current indirect regularisation. The consistent  $k = 17$  budget effectively means QGSA is operating as a fixed- $k$  selector in practice, which is why the ablation configurations with and without adaptive  $k$  show identical latency and accuracy.

#### 4.8 Generalisation Across Base Models

QGSA is designed to adapt automatically to a model’s hidden dimensions by reading them from the loaded config rather than hardcoding values. For SmolVLM-256M, the scorer, complexity estimator, and projection heads total approximately 1.2M trainable parameters; for moondream2, this

Table 6: Supported base models and QGSA configuration for each. Both SmolVLM variants share the same vision encoder ( $d_v = 768$ ); the difference in QGSA parameter count between models is driven by  $d_q$ .

Model	$d_v$	$d_q$	Patch Size	Image Size
SmolVLM-256M-Instruct	768	576	14	384
SmolVLM-500M-Instruct	768	960	14	384
moondream2	1152	2048	14	378

risers to  $\sim 5.5M$  due to the larger  $d_v$  and  $d_q$ . In all cases QGSA adds less than 1% to the total parameter count. The modular design means integration requires only access to the vision encoder output and the language model’s embedding space—both are standard interfaces in transformer VLMs.

## 5 Conclusion

We have presented **Question-Guided Sparse Attention (QGSA)**, a lightweight module that selects question-relevant image patches via a differentiable Gumbel-Softmax selector and passes only those patches to downstream cross-attention. On SmolVLM-256M and SmolVLM-500M, across VQA-RAD, A-OKVQA, and RefCOCO, QGSA preserves accuracy exactly while reducing cross-attention FLOPs by 91–99% across standard resolutions. The module is small (1.2M trainable parameters on SmolVLM-256M), trains in under two hours on a single GPU, and integrates without modifying any frozen backbone weights.

The grounding results are modest in absolute terms (0.31% IoU on RefCOCO) but represent a  $31\times$  improvement over random selection without any bounding-box supervision, which is encouraging evidence that the self-supervised losses are doing something useful. Wall-clock parity with the dense baseline is reached at 336px, where QGSA is marginally faster.

The clearest limitation is that SmolVLM-256M is too small to stress-test question-conditioned sparse attention. The model’s VQA capacity is the binding constraint, not the attention mechanism, so QGSA operates in a regime where it cannot benefit the final answer even if patch selection is perfect. Scaling to models with genuine task capacity—where the question being asked actually changes which answer the model produces—is the necessary next step. At higher resolutions and larger scales, the FLOPs argument becomes compelling: a  $76\times$  theoretical speedup in cross-attention at 576px is a real efficiency gain, even if current hardware makes it hard to fully realise in wall-clock time.

## Limitations

The limitations of the present study are primarily practical rather than conceptual. QGSA requires an additional fine-tuning stage ( $\sim 1.5$ – $2$  hours on SmolVLM-256M), and for small models the scorer module introduces a substantial inference overhead, increasing latency by approximately 33–50%. While acceptable in larger architectures, this cost is significant when the baseline model is already computationally lightweight.

In addition, the adaptive budget estimator consistently converges to a nearly uniform allocation ( $k \approx 17$ ) irrespective of question type. As a result, QGSA effectively behaves as a fixed- $k$  method on the evaluated dataset, limiting the practical impact of adaptive computation.

Grounding performance also remains weak, with an IoU of only 0.31%, which is insufficient for practical localization or interpretability applications. Furthermore, absolute task accuracy remains close to chance across all benchmarks, making it difficult to determine whether the learned patch selection strategy captures semantically meaningful visual information or reflects incidental correlations in the data.

Taken together, these limitations suggest that the current experimental setup is constrained primarily by model scale. Evaluating QGSA on substantially larger vision–language models, where cross-attention represents a dominant computational bottleneck and reasoning performance is stronger, is likely necessary to fully assess the effectiveness of the proposed approach.

## Acknowledgments

We thank the reviewers for their anonymous feedback which helped improve the paper quality.

## References

Vikhyat Agarwal. 2024. [moondream2: A tiny vision-language model](#).

- Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. 2016. [Vqa: Visual question answering](#). *Preprint*, arXiv:1505.00468.
- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, and 8 others. 2022. [Flamingo: a visual language model for few-shot learning](#). *Preprint*, arXiv:2204.14198.
- Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. [Estimating or propagating gradients through stochastic neurons for conditional computation](#). *Preprint*, arXiv:1308.3432.
- Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. [Token merging: Your vit but faster](#). *Preprint*, arXiv:2210.09461.
- Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. [Instructblip: Towards general-purpose vision-language models with instruction tuning](#). *Preprint*, arXiv:2305.06500.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. [An image is worth 16x16 words: Transformers for image recognition at scale](#). *Preprint*, arXiv:2010.11929.
- Hugging Face. 2024. [Smolvlm: Small vision-language models](#).
- Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. 2025. [Mme: A comprehensive evaluation benchmark for multimodal large language models](#). *Preprint*, arXiv:2306.13394.
- Montani I. Van Landeghem S. Honnibal, M. and A Boyd. 2020. Spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *scirp.org*.
- Drew A. Hudson and Christopher D. Manning. 2019. [Gqa: A new dataset for real-world visual reasoning and compositional question answering](#). *Preprint*, arXiv:1902.09506.
- Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. 2021. [Perceiver: General perception with iterative attention](#). *Preprint*, arXiv:2103.03206.
- Eric Jang, Shixiang Gu, and Ben Poole. 2017. [Categorical reparameterization with gumbel-softmax](#). *Preprint*, arXiv:1611.01144.
- Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. [Mdetr – modulated detection for end-to-end multi-modal understanding](#). *Preprint*, arXiv:2104.12763.
- Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *Conference on Neural Information Processing Systems (NeurIPS)*.
- Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Henning Müller. 2018. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5(1):1–10.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. [Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models](#). *Preprint*, arXiv:2301.12597.
- Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. 2022. [Grounded language-image pre-training](#). *Preprint*, arXiv:2112.03857.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. [Evaluating object hallucination in large vision-language models](#). *Preprint*, arXiv:2305.10355.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Conference on Neural Information Processing Systems (NeurIPS)*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). *Preprint*, arXiv:2103.00020.
- Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. [Dynamicvit: Efficient vision transformers with dynamic token sparsification](#). *Preprint*, arXiv:2106.02034.
- Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2019. [Object hallucination in image captioning](#). *Preprint*, arXiv:1809.02156.
- Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. [A-okvqa: A benchmark for visual question answering using world knowledge](#). *Preprint*, arXiv:2206.01718.

Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. 2021. [Evo-vit: Slow-fast token evolution for dynamic vision transformer](#). *Preprint*, arXiv:2108.01390.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. [Modeling context in referring expressions](#). *Preprint*, arXiv:1608.00272.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. [Deep modular co-attention networks for visual question answering](#). *Preprint*, arXiv:1906.10770.

## A Extended Empirical Analysis

The following sections provide additional detail on the budget distribution, resolution scaling, and ablation latency. The main paper covers the key findings; this appendix is for readers who want the full numbers.

### A.1 Adaptive Budget Distribution

The complexity estimator is designed to allocate a budget  $k \in [k_{\min}, k_{\max}]$  based on perceived question difficulty. In practice it converges to  $\bar{k} \approx 17.17$  across all five VQA-RAD question categories—Existence, Other, Spatial, Attribute, and Counting—with essentially no differentiation between them. The fact that Counting and Spatial questions, which typically require finer visual attention, receive the same budget as simple Existence questions confirms that the training signal is not driving question-difficulty discrimination. We discuss the causes and potential fixes in Section 4.7 of the main paper.

### A.2 Resolution Scaling and Efficiency

Figure 3 provides the full FLOPs reduction and speedup curves across resolutions. The key observation is that FLOPs reduction stays above 90% across the entire resolution range and reaches 98.7% at 576px, while the theoretical speedup ( $N/k$ ) grows superlinearly because  $N$  scales as resolution squared while  $k$  remains fixed. At 576 px, QGSA achieves a  $76.2\times$  theoretical speedup in cross-attention for SmolVLM-256M. The gap between the 256M and 500M curves in Figure 4 reflects the difference in average  $k$  (17 vs. 18) rather than any architectural difference.

### A.3 Ablation Latency Breakdown

Figure 5 shows the latency for each ablation configuration on the 150-sample VQA-RAD subset. The  $\sim 100$  ms jump from Dense to any QGSA variant is almost entirely the scorer overhead, not cross-attention. Adding losses or switching from fixed

to adaptive  $k$  changes latency by at most 2.4 ms (within noise). Accuracy is 3.11% for all configurations on this subset.

**A note on sparse attention on standard hardware.** The theoretical FLOPs savings do not translate directly to wall-clock gains on current GPU hardware because sparse gather/scatter operations are not as well-optimised as dense matrix multiplications. Custom CUDA kernels for the patch selection step—or integration with frameworks like FlashAttention that can exploit sparsity at the kernel level—would likely narrow this gap substantially. We leave this as an engineering task for future work.

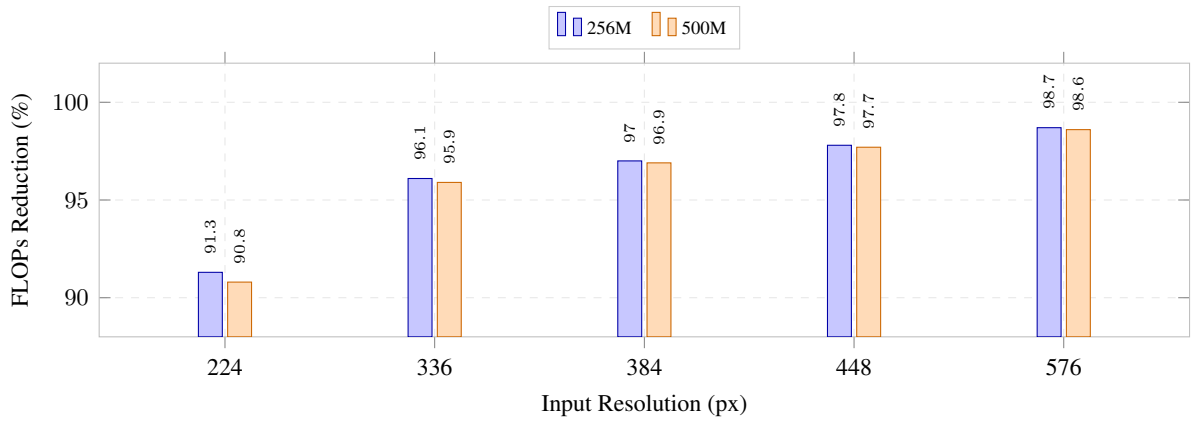


Figure 3: Cross-attention FLOPs reduction. FLOPs reduction stays above 90% throughout and is nearly identical across model sizes, since both use similar  $k$ .

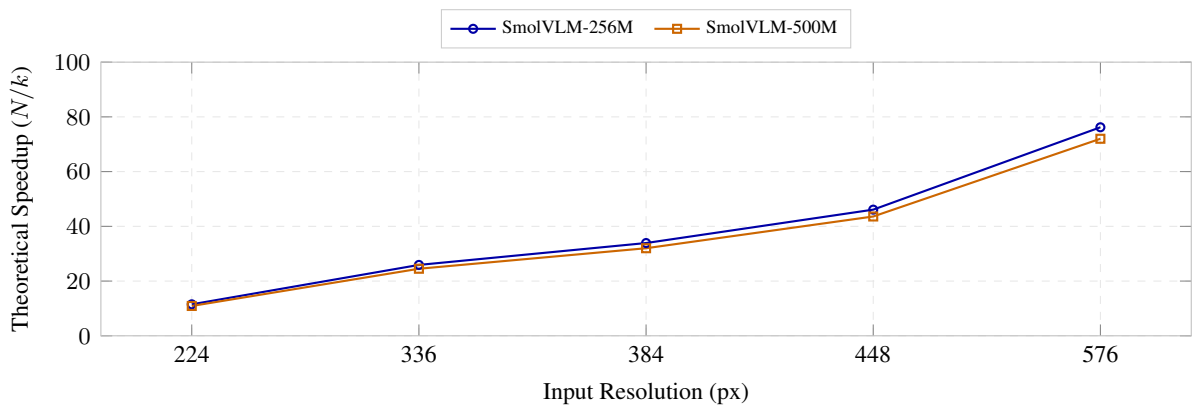


Figure 4: Theoretical speedup ( $N/k$ ) scaling across resolutions. Speedup grows superlinearly with resolution; the 256M and 500M curves separate slightly because  $k = 17$  vs.  $k = 18$ .

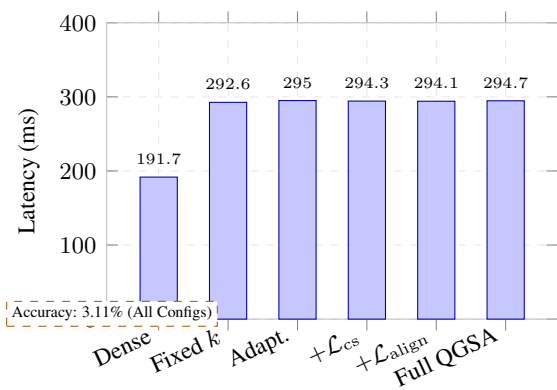


Figure 5: Ablation latency on the 150-sample VQA-RAD subset. The large step from Dense to Fixed  $k$  is the scorer; subsequent additions change latency by  $<3$  ms.

# Systematic Performance Degradation in Indic Vision-Language Models: Evidence from Hindi and Telugu

Rishikant Chigrupaatii<sup>1,\*</sup>, Ponnada Sai Tulasi Kanishka<sup>1,\*</sup>, Lalit Chandra Routhu<sup>1,\*</sup>,  
Martin Patel<sup>1,\*</sup>, Sama Supratheek Reddy<sup>1</sup>, Divyam Gupta<sup>1</sup>, Rajiv Misra<sup>1</sup>, Rohun Tripathi<sup>2</sup>

<sup>1</sup>Indian Institute of Technology Patna <sup>2</sup>Allen Institute for AI

## Abstract

With 1.5 billion people speaking over 120 major languages, India exemplifies the challenges of multilingual AI evaluation. Current multilingual VLM benchmarks suffer from unverified auto-translations, narrow task coverage, small sample sizes, and lack of culturally grounded content. We present HinTel-AlignBench, a comprehensive evaluation framework and benchmark for Hindi and Telugu vision-language models with English-aligned samples. Our framework combines semi-automated translation with human verification to generate  $\sim 4k$  QA pairs per language across five domains: adapted English datasets (VQAv2, RealWorldQA, CLEVR-Math) and native Indic sets (JEE for STEM, VAANI for cultural grounding). Evaluation of state-of-the-art open and closed-source VLMs reveals consistent performance regression from English to Indic languages, with average drops of 8.3 points for Hindi and 5.5 points for Telugu across four of five tasks. We identify key failure modes and establish reproducible baselines for multilingual multimodal evaluation.

## 1 Introduction

India’s 122 major languages and 1599 other languages<sup>1</sup> present unique challenges for multilingual AI. While recent multimodal large language models (MLLMs) such as ChatGPT (OpenAI, 2025), Gemini 2.5 (Google DeepMind, 2025), and open-weight variants (Meta Llama, 2025; Dash et al., 2025) claim multilingual support, comprehensive evaluation benchmarks for Indian languages remain scarce.

Current evaluation methodologies suffer from critical limitations in quality, scope, and scale. First, many benchmarks rely on unverified automatic translations (Wu et al., 2025), inevitably

\*Equal contribution.

<sup>1</sup>[https://en.wikipedia.org/wiki/Languages\\_of\\_India](https://en.wikipedia.org/wiki/Languages_of_India)

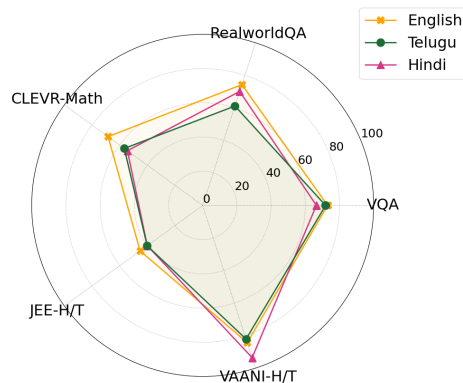


Figure 1: Average performance of GPT-4.1 and Gemini-2.5-Flash on English, Hindi, and Telugu across data-parallel visual question answering samples. Performance regresses from English to Hindi by 8.3 points and from English to Telugu by 5.5 points.

introducing noise. While text-only benchmarks like IndicGenBench (Singh et al., 2024) exist, they lack multimodal coverage. Second, existing vision-language benchmarks often suffer from insufficient sample sizes; for instance, xChat (Yue et al., 2025) and AyaVisionBench (Dash et al., 2025) contain only 50 and 135 QA pairs per language, respectively, preventing statistically significant analysis. Third, domain coverage is often narrow. Concurrent work such as Kaleidoscope (Salazar et al., 2025) provides  $\sim 800$  samples per Indic language but focuses exclusively on exam-based multiple-choice questions, neglecting real-world reasoning. Finally, adapted benchmarks often lack cultural grounding, evaluating surface-level translation rather than native competence (Khan et al., 2024). We elaborate on previous works in the appendix.

To address these gaps, we introduce HinTel-AlignBench, a scalable framework and benchmark for evaluating VLMs in Hindi and Telugu. Our semi-automated pipeline combines translation or LLM-based QA generation with strict human verifi-

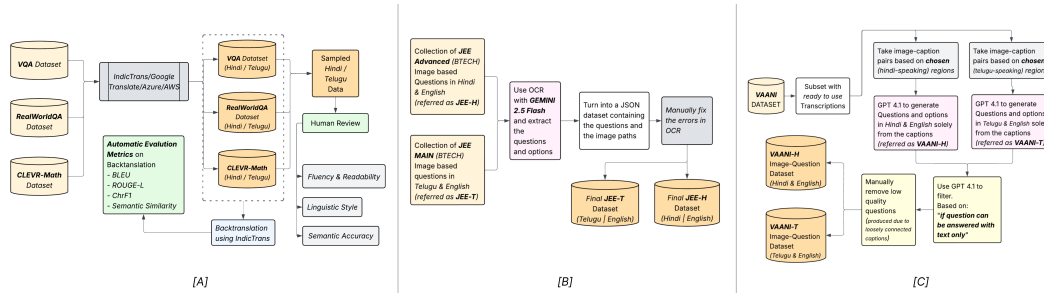


Figure 2: Dataset generation pipeline for (A) VQAv2, RealWorldQA, and CLEVR-Math using translation and human verification; (B) JEE-H and JEE-T using OCR extraction and verification; (C) VAANI-H and VAANI-T using LLM-based question generation from captions with filtering and verification.

ation, achieving 5x faster processing than manual creation for 79% of samples while maintaining linguistic fidelity. The benchmark comprises  $\sim 4k$  QA pairs per language—significantly larger than prior manually verified sets—spanning five domains: real-world understanding (VQAv2 (Goyal et al., 2017a)), practical reasoning (RealWorldQA (xAI, 2024a)), visual mathematics (CLEVR-Math (Lindström and Abraham, 2022)), STEM competency (JEE-Vision from India’s Joint Entrance Exam), and cultural grounding (VAANI (Team, 2025)). Crucially, each sample includes manually verified English translations, enabling direct cross-lingual comparison.

Evaluation of state-of-the-art models on our benchmark reveals systematic performance degradation. Across all models, we observe average regressions of 8.3 points (Hindi) and 5.5 points (Telugu) relative to English, with gaps appearing in four of five tasks (Figure 1). Even frontier models like GPT-4.1 exhibit 3.8-point (Hindi) and 8.6-point (Telugu) performance drops. Performance on aligned Hindi and Telugu subsets differs by less than 1 point, indicating comparable gaps between English and both Indic languages.

Our contributions are: (1) a semi-automated framework for generating multilingual vision-language evaluation sets; (2) the largest human-verified Hindi and Telugu VLM benchmark to date, featuring culturally sourced content and English-aligned samples; and (3) a comprehensive evaluation of state-of-the-art models, highlighting significant performance regressions across diverse domains.

## 2 Datasets

### 2.1 Data Sources

We construct HinTel-AlignBench by combining translated English VQA datasets with native Indic evaluation sets across five domains. The translated sets include 1000 samples from VQAv2 (Goyal et al., 2017b) for real-world visual understanding, 765 samples from RealWorldQA (xAI, 2024a,b) for practical spatial reasoning, and 1000 samples from CLEVR-Math (Lindström and Abraham, 2022) for visual mathematical reasoning.

For native Indic content, we develop JEE-Vision from India’s Joint Entrance Examination, sourcing 192 Hindi questions (JEE-H) from the Advanced exam and 325 Telugu questions (JEE-T) from the Mains exam. These diagram-dependent STEM problems span mathematics, physics, and chemistry, providing the first benchmark for non-translated multilingual technical reasoning with visual content. We generate culturally grounded evaluation sets (VAANI-H and VAANI-T) by sampling 945 Hindi and 1020 Telugu images from the VAANI corpus (Team, 2025), using GPT-4.1 to create multiple-choice questions from original captions, then filtering out text-only answerable questions.

Translation-based extension enables multi-way parallel data, allowing attribution of performance to task knowledge versus language understanding (Singh et al., 2024). This approach also leverages the quality control invested in designing the original English benchmarks. Table 1 shows the distribution of QA pairs per language and task. Figure 3 showcases a few examples.

### 2.2 Dataset Generation Framework

Figure 2 illustrates our three-stage generation framework tailored to different data sources.

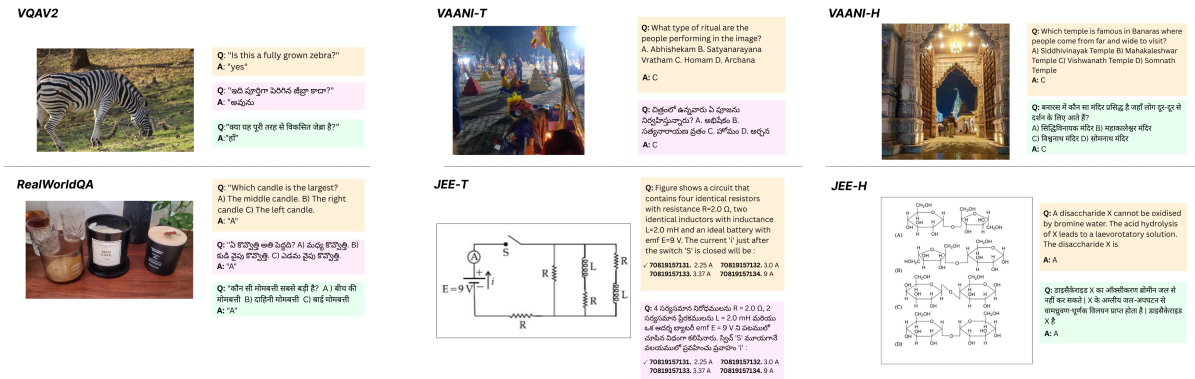


Figure 3: Qualitative Examples for different domains in our dataset. More images are shown in the appendix

Language	VQAv2	RealWorldQA	CLEVR-Math	JEE-H	JEE-T	VAANI-H	VAANI-T
Hindi	1,000	765	1,000	192	-	945	-
Telugu	1,000	765	1,000	-	325	-	1,020
English	1,000	765	1,000	192	325	945	1,020
<b>Total</b>	<b>3,000</b>	<b>2,295</b>	<b>3,000</b>	<b>384</b>	<b>650</b>	<b>1,890</b>	<b>2,040</b>

Table 1: Number of QA pairs per task per language in HinTel-AlignBench. The samples used in VQAv2, RealWorldQA and CLEVR-Math are the same across all languages.

**Translation Pipeline.** For VQAv2, RealWorldQA, and CLEVR-Math, we evaluate four translation systems (IndicTrans (Gala et al., 2023), Google Translate, Azure, AWS) on 50 diverse samples per language, selecting Azure for Hindi and AWS for Telugu. All translations undergo manual verification for semantic accuracy, linguistic style, and readability (KJ et al., 2025). We avoid back-translation-based sample selection, which introduces bias toward high-confidence translation errors. Manual review accepts 79% of VQAv2 translations without modification; among modified samples, 42% require only minor changes (verb tense), while 58% need word addition or deletion. Samples requiring only minor edits process 5x faster than generation from scratch. All verification is performed by co-author native speakers. We use one annotator per sample to maximize dataset size within budget constraints.

**JEE-Vision Creation.** India’s Joint Entrance Examination provides authentic STEM problems authored by subject-matter experts in target languages (JEE Mains: 13 languages; Advanced: English/Hindi), avoiding translation artifacts. We curate diagram-dependent problems, evaluating joint understanding of technical visuals and linguistic content. Questions and options are extracted using Gemini-2.5-Flash OCR (Google DeepMind, 2025),

then manually verified to correct OCR errors.

**VAANI Generation.** From the VAANI corpus (Team, 2025), we extract images with text transcriptions from Hindi and Telugu speaking regions. Since no images have both Hindi and Telugu transcriptions, we create separate language-specific sets. Text-only GPT-4.1 generates multiple-choice questions from captions, which undergo two-stage refinement: automated filtering removes questions answerable without images, followed by human verification to eliminate low-quality questions. This process addresses cases where VAANI captions do not perfectly align with images.

### 3 Experimental Setup

**Models.** We evaluate open-weight and proprietary models claiming Indic language support. For Hindi, we test Gemma3 (4B, 12B, 27B) (Team et al., 2025), Qwen2.5VL-7B (Bai et al., 2025), Llama3.2-Vision-11B (Meta Llama, 2025), Aya-8B (Dash et al., 2025), Chitarth-8B (Khan et al., 2024), GPT-4.1 (OpenAI, 2025), and Gemini-2.5-Flash (Google DeepMind, 2025). For Telugu, fewer models provide support; we evaluate the Gemini variants, GPT-4.1, and Chitarth-8B.

**Metrics.** For multiple-choice tasks (RealWorldQA, VAANI, JEE), we report standard accuracy. For open-ended generation (VQAv2, CLEVR-Math), we utilize a hybrid evaluation protocol combining exact match with a GPT-4.1 judge to account for linguistic variations, following standard VQA practices (complete details in appendix).

Model	VQAv2		RealWorldQA		CLEVR-Math		JEE-T		VAANI-T		Ours-T	
	Tel	En	Tel	En	Tel	En	Tel	En	Tel	En	Tel	En
GPT 4.1	68.70	72.00	61.05	<b>75.29</b>	46.60	<u>65.00</u>	34.36	40.92	<u>81.57</u>	<b>82.45</b>	58.46	67.13
Gemini 2.5 Flash	<u>75.10</u>	74.10	<b>61.18</b>	<u>73.20</u>	<b>66.70</b>	<b>71.80</b>	<b>45.90</b>	<b>54.15</b>	<b>82.75</b>	80.78	<b>66.33</b>	<b>70.81</b>
Gem 2.0 Flash	70.20	74.20	<u>60.92</u>	69.67	43.60	53.50	<u>42.15</u>	<u>53.23</u>	80.49	79.61	59.47	66.04
Gem 1.5 Flash	68.50	<u>74.40</u>	60.00	67.19	37.40	46.70	29.85	39.38	76.27	79.61	54.40	61.46
Chitrarth	<b>76.00</b>	<b>78.50</b>	53.59	52.55	<u>53.90</u>	56.90	20.00	18.15	<u>81.57</u>	<u>82.45</u>	57.01	57.71
Model Mean	71.10	74.64	59.35	67.58	49.64	58.78	34.85	41.17	80.53	80.98	59.13	64.63

Table 2: Results (in %) for Telugu and English. **Bold** indicates the best and underline indicates the next best.

Model	VQAv2		RealWorldQA		CLEVR-Math		JEE-H		VAANI-H		Ours-H	
	Hi	En	Hi	En	Hi	En	Hi	En	Hi	En	Hi	En
GPT-4.1	<b>68.00</b>	72.00	<b>70.59</b>	<b>75.29</b>	48.10	65.00	<u>23.18</u>	<u>23.05</u>	<u>93.33</u>	<u>86.88</u>	60.64	64.44
Gemini 2.5 Flash	65.00	74.10	<u>69.54</u>	<u>73.20</u>	<b>60.70</b>	<u>71.80</u>	<b>56.90</b>	<b>62.89</b>	<b>93.86</b>	<b>87.19</b>	<b>69.20</b>	<b>73.84</b>
Chitrarth	<u>66.00</u>	<b>78.50</b>	52.94	52.55	<u>57.20</u>	56.90	11.72	13.93	84.23	80.14	54.42	56.40
Qwen2.5VL-7B	37.20	<u>74.30</u>	51.11	68.10	29.10	<b>98.80</b>	17.84	20.70	81.79	84.76	43.41	69.33
Aya-8B	36.30	47.30	55.42	58.82	46.20	61.40	9.63	16.02	82.22	80.42	45.95	52.79
LLaMA 3.2 11B	35.90	59.80	35.68	61.57	18.90	35.60	14.32	14.45	77.67	83.28	36.49	50.94
Gemma3-27B	64.10	65.50	54.38	61.04	43.50	53.70	19.66	17.58	87.41	82.01	53.81	55.97
Gemma3-12B	63.00	65.70	53.98	58.69	40.20	46.80	14.84	16.80	85.50	82.33	51.50	54.87
Gemma3-4B	55.00	58.20	43.27	50.19	33.20	39.60	14.32	17.19	80.64	77.78	45.29	48.59
Model Mean	53.74	66.26	53.99	62.49	43.01	58.62	20.01	22.29	85.85	83.76	51.19	58.49

Table 3: Results (in %) for Hindi and English. **Bold** indicates the best and underline indicates the next best.

## 4 Results and Analysis

### 4.1 Main Results

Tables 2 and 3 present Telugu-English and Hindi-English comparisons. Across all models and tasks, average performance regresses 5.5 points from English to Telugu and 8.3 points from English to Hindi. Performance drops occur in four of five tasks, with VAANI showing smaller gaps. On aligned samples spanning VQAv2, CLEVR-Math, and RealWorldQA evaluated with GPT-4.1, Gemini-2.5-Flash, and Chitrarth, Hindi, Telugu, and English achieve 61.51, 62.53, and 68.6 points respectively, demonstrating systematic degradation from English to both Indic languages.

Gemini-2.5-Flash achieves best overall performance on both language pairs. Chitrarth leads on VQAv2 due to multilingual VQAv2 training. However, models show substantial cross-language variance: GPT-4.1 excels on RealWorldQA in English and Hindi but underperforms on Telugu, highlighting the need for comprehensive evaluation across all target languages. Qwen2.5VL-7B exhibits the largest Hindi-English gap at 25.92 points.

### 4.2 Task-Specific Analysis

Figure 4 shows average performance regression per task. CLEVR-Math and RealWorldQA exhibit the largest English-Indic gaps, while VAANI shows the smallest. VAANI-H performance exceeds English

by 2.09 points on average. Analysis reveals two factors: first, some English questions fail to capture Indic-script option meanings in images with visible Indic text. Second, text-only LLM-generated distractors may enable statistical pattern exploitation.

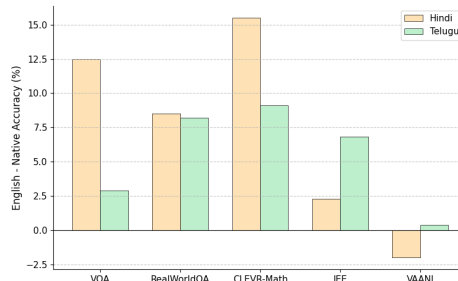


Figure 4: Average performance regression from English to Indic languages per domain. All tasks except VAANI show consistent regression.

Chain-of-Thought prompting improves reasoning tasks but benefits English more than Hindi. On JEE-H, CoT gains 13.8 points in English versus 2.22 points in Hindi, suggesting training bias toward English CoT data. Detailed ablation studies and error analysis are reported in the appendix.

## 5 Conclusion

This paper introduces HinTel-AlignBench, a framework for developing benchmarks to evaluate multimodal large language models in Hindi and Tel-

ugu, addressing critical gaps in existing multilingual evaluations. We combined semi-automated dataset creation with rigorous human verification and sourced culturally grounded native datasets to assess diverse capabilities. Evaluations of state-of-the-art VLMs reveal significant performance regressions in Indic languages compared to English, emphasizing the need for targeted improvements in multilingual visual understanding.

## Limitations

While our benchmark introduces a diverse evaluation set it has limitations. First, the proprietary models we evaluated achieve high scores on the VAANI-H/T. We use text only LLMs for generating QA from VAANI captions and they often do not design good distractors. Thus, the models may guess the correct answer by exploiting statistical patterns, which inflates metrics. A future work is using Multi-Binary Accuracy (Cai et al., 2024) for VAANI subsets. Second, the JEE-H benchmark contains only 2 Mathematics questions, due to lack of image-based mathematics questions in the JEE-Advanced examination. Finally, there are 22 official Indian languages and we cover English and 2/22 (Hindi and Telugu) with this work. We hope this benchmark gets extended to all other Indic languages with contributions from native speakers from those languages.

## References

- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. [Qwen2.5-vl technical report](#). *arXiv preprint arXiv:2502.13923*.
- Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, Yao Dou, Jaden Park, Jianfeng Gao, Yong Jae Lee, and Jianwei Yang. 2024. Temporal-bench: Towards fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*.
- Saurabh Dash, Yiyang Nan, John Dang, Arash Ahmadian, Shivalika Singh, Madeline Smith, Bharat Venkitesh, Vlad Shmyhlo, Viraat Aryabumi, Walter Beller-Morales, Jeremy Pekmez, Jason Ozuzu, Pierre Richemond, Acyr Locatelli, Nick Frosst, Phil Blunsom, Aidan Gomez, Ivan Zhang, Marzieh Fadaee, and 6 others. 2025. [Aya vision: Advancing the frontier of multilingual multimodality](#). *arXiv preprint arXiv:2505.08751*, arXiv:2505.08751.
- Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Pudupully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. [Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages](#). *Transactions on Machine Learning Research*.
- Google DeepMind. 2025. [Gemini 2.5: Our most intelligent ai model](#). Google DeepMind Blog. Released on March 25, 2025.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017a. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017b. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6904–6913.
- Shaharukh Khan, Ayush Tarun, Abhinav Ravi, Ali Faraz, Akshat Patidar, Praveen Kumar Pokala, Anagha Bhangare, Raja Kolla, Chandra Khatri, and Shubham Agarwal. 2024. Chittrarth: Bridging vision and language for a billion people. In *NeurIPS Multimodal Algorithmic Reasoning*.
- Sankalp KJ, Ashutosh Kumar, Laxmaan Balaji, Nikunj Kotecha, Vinija Jain, Aman Chadha, and Sreyoshi Bhaduri. 2025. Indicmmlu-pro: Benchmarking indic large language models on multi-task language understanding. *arXiv preprint arXiv:2501.15747*.
- Adam Dahlgren Lindström and Savitha Sam Abraham. 2022. [Clevr-math: A dataset for compositional language, visual, and mathematical reasoning](#). *arXiv preprint*.
- Meta Llama. 2025. Llama 3.2-vision: Instruction-tuned image reasoning generative models. Model release by Meta. Available at: <https://huggingface.co/meta-llama/Llama-3.2-11B-Vision>.
- OpenAI. 2025. Gpt-4.1: A new series of gpt models with major improvements on coding, instruction following, and long context. OpenAI Company Website. Released on April 14, 2025.
- Israfel Salazar, Manuel Fernández Burda, Shayekh Bin Islam, Arshia Soltani Moakhar, Shivalika Singh, Fabian Farestam, Angelika Romanou, Danylo Boiko, Dipika Khullar, Mike Zhang, Dominik Krzemiński, Jekaterina Novikova, Luísa Shimabucoro, Joseph Marvin Imperial, Rishabh Maheshwary, Sharad Duwal, Alfonso Amayuelas, Swati Rajwal, Jebish Purbey, and 25 others. 2025. [Kaleidoscope: In-language exams for massively multilingual vision evaluation](#). *Preprint*, arXiv:2504.07072.

- Harman Singh, Nitish Gupta, Shikhar Bharadwaj, Dinesh Tewari, and Partha Talukdar. 2024. **IndicGenBench: A multilingual benchmark to evaluate generation capabilities of LLMs on Indic languages**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11047–11073, Bangkok, Thailand. Association for Computational Linguistics.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.
- VAANI Team. 2025. **Vaani: Capturing the language landscape for an inclusive digital india (phase 1)**. <https://vaani.iisc.ac.in/>.
- Minghao Wu, Weixuan Wang, Sinuo Liu, Huifeng Yin, Xintong Wang, Yu Zhao, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025. **The bitter lesson learned from 2,000+ multilingual benchmarks**. *arXiv preprint arXiv:2504.15521*, arXiv:2504.15521.
- xAI. 2024a. Grok-1.5 vision preview. <https://x.ai/blog/grok-1.5v>.
- xAI. 2024b. **Realworldqa dataset**. Hugging Face Dataset Repository.
- Xiang Yue, Yueqi Song, Akari Asai, Seungone Kim, Jean de Dieu Nyandwi, Simran Khanuja, Anjali Kantharuban, Lintang Sutawika, Sathyanarayanan Ramamoorthy, and Graham Neubig. 2025. **Pangea: A fully open multilingual multimodal LLM for 39 languages**. In *The Thirteenth International Conference on Learning Representations*.

# How Fragile Is Vision-Language Alignment? Mapping Concept Disruption Under Text-to-Image Personalization

Mujtaba Hasan  
New Delhi, India  
mujtaba.hasan@live.com

## Abstract

Text-to-image diffusion models learn a mapping from natural language to visual structure, but how robust is this mapping to perturbation? We use personalization—fine-tuning a model to learn a new face, object, or style—as a controlled stress test to probe the fragility of learned vision-language alignment. We find that fine-tuning for *one* concept systematically shifts the model’s ability to faithfully render *unrelated* concepts, and that this disruption follows structured, predictable patterns. To measure this fragility, we construct **Concept Entanglement Maps**: per-prompt, per-model disruption matrices that reveal which concepts are most affected and why. Using Stable Diffusion v1.5 as a controlled testbed, we evaluate 15 subjects across three personalization methods on 200 prompts and report three findings about the organization of vision-language alignment: (1) aggregate disruption is larger for vision-backbone and cross-attention perturbations than for text-embedding perturbations, despite the latter directly modifying the language representation; (2) abstract and compositional language is significantly more fragile than concrete, object-specific language; and (3) disruption does *not* follow semantic proximity—personalizing for a face does not preferentially disrupt other face-related prompts ( $p = 1.0$ ), suggesting that alignment vulnerability is organized globally rather than purely by semantic category. These findings expose a structural vulnerability in current text-to-image personalization: the same cross-attention mechanism that enables compositional generalization also creates pathways through which local fine-tuning can propagate as global alignment shift.

## 1 Introduction

The ability of text-to-image diffusion models to translate natural language descriptions into coherent visual outputs depends on *vision-language alignment*—the learned correspondence between

text token representations and spatial visual features. This alignment, mediated by cross-attention layers that project text embeddings into the visual feature space, enables remarkable compositional generalization: a model can render “a koala wearing a top hat on the moon” despite never observing this combination, because it has learned how each word anchors to visual structure (Rombach et al., 2022; Hertz et al., 2023).

But how robust is this learned alignment? If we perturb the model’s parameters to teach it a single new concept—a specific person’s face, a particular object, an artistic style—does the alignment for all other concepts remain intact? Or does local fine-tuning propagate as global alignment shift?

This question is both scientifically fundamental and practically urgent. Scientifically, the answer reveals how vision-language grounding is organized in these models: is it modular, with each concept grounded independently, or entangled, with concepts sharing representational substrate such that modifying one affects others? Practically, personalized adapters are increasingly used as general-purpose creative tools. A user who fine-tunes an adapter for a product, character, or visual style typically expects that unrelated prompts remain faithful. If a personalization silently shifts how the model grounds unrelated language—for example, abstract prompts, people prompts, or scene descriptions—then failures may surface only after deployment, where they are difficult to diagnose.

We formalize this question through the lens of **alignment stability**. Let  $A(\theta, p)$  denote the alignment quality between a model with parameters  $\theta$  and a text prompt  $p$ , measured as the CLIP text-image cosine similarity between  $p$  and the model’s output. An ideal personalization  $\theta \rightarrow \theta + \Delta_s$  for subject  $s$  would satisfy:

$$A(\theta + \Delta_s, p) = A(\theta, p) \quad \forall p \notin \mathcal{P}_s, \quad (1)$$

where  $\mathcal{P}_s$  is the set of prompts related to  $s$ . That

is, alignment for non-target prompts should be invariant to personalization. We show empirically that this invariance is systematically violated in aggregate, and that the violations have predictable structure.

To measure these violations, we introduce **Concept Entanglement Maps**: a structured diagnostic framework that produces a disruption matrix  $\mathbf{M} \in \mathbb{R}^{P \times S}$ , where entry  $M_{p,s}$  quantifies the alignment shift for prompt  $p$  after personalizing for subject  $s$ . Unlike scalar evaluation metrics that assess target-concept fidelity, the entanglement map reveals the collateral structure of alignment disruption—which prompts are fragile, which methods cause broader shifts, and whether disruption follows semantic proximity.

Personalization is not our topic; it is our experimental instrument. Three personalization methods—Textual Inversion (Gal et al., 2023), DreamBooth-LoRA (Hu et al., 2022; Ruiz et al., 2023), and Custom Diffusion (Kumari et al., 2023)—modify the model at different points in the vision-language pipeline: text embeddings, UNet attention layers, and cross-attention projections respectively. By comparing the disruption patterns they produce, we can localize where alignment is most fragile within the architecture.

Our contributions are:

1. We formalize the alignment stability problem and introduce Concept Entanglement Maps as a structured diagnostic (§3).
2. We show that aggregate non-target disruption is smallest for text-space perturbations and largest for cross-attention/vision-backbone perturbations, revealing the cross-attention interface as a fragile architectural link (§4).
3. We demonstrate that abstract, compositional language is significantly more fragile than concrete language, and that disruption does not follow semantic proximity (§5).
4. We provide preliminary cross-attention analysis visualizing how personalization can shift the spatial grounding of unrelated text tokens (§6).

Our findings map to a core concern of the ALVR community: shortcomings of existing large vision and language models on downstream tasks. We show that alignment is not merely imperfect; in

SD v1.5 personalization, it is structurally fragile in measurable and predictable ways.

## 2 Related Work

### **Vision-language grounding in generative models.**

Text-to-image diffusion models ground language in visual structure primarily through cross-attention, where text token embeddings serve as keys and values while spatial visual features serve as queries (Rombach et al., 2022). Hertz et al. (2023) showed that manipulating these attention maps enables fine-grained image editing, demonstrating that they encode spatial correspondence between words and visual regions. Tang et al. (2023) developed attribution maps that trace which image regions are influenced by which text tokens, revealing that grounding quality varies across concept types. Our work extends this line of inquiry by asking how stable this grounding is under model perturbation.

### **Alignment robustness and concept interference.**

The robustness of learned representations to perturbation is a well-studied concern in NLP (Ribeiro et al., 2020) and vision-language modeling. Fine-tuning can also produce interference or forgetting (Kirkpatrick et al., 2017; Li and Hoiem, 2016). In diffusion models, Kumari et al. (2023) note that customization can cause language drift, where common words change meaning, and recent work studies memorization and robustness issues in generative models (Wen et al., 2024). Our work differs in providing a structured, per-prompt diagnostic rather than aggregate metrics, and in explicitly testing whether disruption follows semantic organization.

### **Text-to-image personalization as a probe.**

Personalization methods span the full text-to-image pipeline. Textual Inversion (Gal et al., 2023) modifies only the text embedding, adding a learned token to CLIP’s vocabulary. LoRA (Hu et al., 2022) applies low-rank perturbations to the UNet’s attention projections. Custom Diffusion (Kumari et al., 2023) fine-tunes cross-attention key/value matrices, operating directly at the language-vision interface. DreamBooth (Ruiz et al., 2023) fine-tunes the model for subject-driven generation. These methods create a natural set of controlled perturbations at known architectural locations, making them useful probes for studying where alignment is most fragile. We do not evaluate these methods’ personalization quality; we use them as instruments to

stress-test alignment.

### Evaluation of personalization and alignment.

Standard metrics—CLIP-I similarity, DINO identity preservation (Caron et al., 2021), CLIPScore (Hessel et al., 2021), LPIPS (Zhang et al., 2018), and FID (Heusel et al., 2017)—assess target concept fidelity, semantic alignment, perceptual distance, or distributional shift. No existing metric systematically measures what happens to non-target concepts after personalization. Our entanglement map fills this gap by measuring the collateral alignment cost that personalization imposes on the rest of the model’s concept vocabulary.

## 3 Method

### 3.1 The Alignment Stability Problem

Let  $G(\cdot; \theta)$  be a text-to-image diffusion model and let  $f_T, f_I$  be the text and image encoders of a pre-trained CLIP model (Radford et al., 2021). The alignment function for prompt  $p$  under model  $\theta$  is:

$$A(\theta, p) = \mathbb{E}_{z \sim \mathcal{N}} [\cos(f_T(p), f_I(G(p, z; \theta)))] , \quad (2)$$

where  $z$  is the latent noise initialization. Personalization produces a perturbation  $\Delta_s$  for subject  $s$ , modifying the model to  $\theta + \Delta_s$ . We define the operational semantic disruption for a non-target prompt  $p$  using seed-matched generations:

$$E_{\text{sem}}(p, s) = \frac{1}{K} \sum_{k=1}^K |a_{p,s,k}^{\text{pers}} - a_{p,k}^{\text{base}}| , \quad (3)$$

where

$$a_{p,k}^{\text{base}} = \cos(f_T(p), f_I(G(p, z_k; \theta))) , \quad (4)$$

$$a_{p,s,k}^{\text{pers}} = \cos(f_T(p), f_I(G(p, z_k; \theta + \Delta_s))) . \quad (5)$$

The absolute value measures instability or shift, not necessarily signed degradation. We use the term *disruption* for this absolute semantic shift, and reserve degradation for cases where the personalized score is lower than the base score. This distinction is important because a non-target prompt can shift in either direction even when the output remains plausible.

The perturbation  $\Delta_s$  occupies different subspaces of the model’s parameter space depending

on the method:

$$\Delta_s^{\text{TI}} \in \mathbb{R}^{d_{\text{emb}}} \quad (\text{text embedding}), \quad (6)$$

$$\Delta_s^{\text{LoRA}} \in \{B_l A_l\}_{l=1}^L, \quad \begin{aligned} A_l &\in \mathbb{R}^{r \times d_l}, \\ B_l &\in \mathbb{R}^{d_l \times r}, \end{aligned} \quad (7)$$

$$\Delta_s^{\text{CD}} \in \{W_l^K, W_l^V\}_{l \in \mathcal{L}_x} \quad (\text{cross-attn. K/V}). \quad (8)$$

These perturbations form a hierarchy along the vision-language pipeline: Textual Inversion perturbs the language representation before it enters the UNet; LoRA perturbs the UNet’s internal processing; Custom Diffusion perturbs the cross-attention projections where text meets vision. Comparing  $E_{\text{sem}}(p, s)$  across these perturbation types localizes the architectural locus of alignment fragility.

### 3.2 Entanglement Map Construction

Given evaluation prompts  $\mathcal{P} = \{p_1, \dots, p_P\}$  and personalized models  $\mathcal{S} = \{\Delta_{s_1}, \dots, \Delta_{s_S}\}$ , we construct the entanglement map  $\mathbf{M} \in \mathbb{R}^{P \times S}$ . For each  $(p, s)$  pair, we generate  $K$  images from both the base and personalized models using identical seeds:

$$\hat{x}_k^{\text{base}} = G(p, z_k; \theta), \quad \hat{x}_k^{\text{pers}} = G(p, z_k; \theta + \Delta_s). \quad (9)$$

The seed-matched design controls for stochastic variation, isolating the effect of  $\Delta_s$ .

**Non-target prompt protocol.** For each non-target evaluation prompt, the base and personalized models receive the exact same text prompt  $p$ . We do not prepend the learned personalization token, subject identifier, or style token during non-target evaluation. This distinction is essential: adding the subject token would measure triggered personalization behavior, whereas our goal is to measure collateral changes in the model’s ordinary prompt-following behavior after the parameter perturbation  $\Delta_s$ . The only difference between the two generations in Eq. 9 is therefore the model parameters, not the text input or random seed. Crucially, non-target evaluation uses the same prompt  $p$  for the base and personalized model; no personalization token is inserted into non-target prompts.

### 3.3 Disruption Metrics

We define two complementary metrics capturing semantic and perceptual dimensions of alignment disruption.

**Semantic disruption.** Equation 3 measures whether personalization changes how well the output matches the text prompt—a direct probe of alignment integrity.

**Perceptual disruption.** The perceptual distance between seed-matched outputs is:

$$E_{\text{per}}(p, s) = \frac{1}{K} \sum_{k=1}^K \text{LPIPS}(\hat{x}_k^{\text{base}}, \hat{x}_k^{\text{pers}}). \quad (10)$$

$E_{\text{per}}$  captures how much the visual output changes regardless of whether it still matches the prompt. We report  $E_{\text{per}}$  for completeness and emphasize  $E_{\text{sem}}$  as the primary alignment metric.

**Aggregate entanglement score.** For method comparison, we aggregate over non-target prompts  $\mathcal{P}_{\text{non}}$  excluding the subject’s own semantic category:

$$E(s) = \frac{1}{|\mathcal{P}_{\text{non}}|} \sum_{p \in \mathcal{P}_{\text{non}}} \frac{1}{2} \left( \widehat{E}_{\text{sem}}(p, s) + \widehat{E}_{\text{per}}(p, s) \right), \quad (11)$$

where  $\widehat{\cdot}$  denotes min-max normalization. We use  $E$  for aggregate method and subject-category comparisons, while  $E_{\text{sem}}$  is used for semantic disruption analyses.

### 3.4 Noise Floor

Even without perturbation, stochastic variation in the diffusion process produces non-zero semantic and perceptual shifts between different seeds of the same model. We establish a noise floor by computing  $E_{\text{sem}}$  and  $E_{\text{per}}$  between different seed pairs of the unperturbed base model across all 200 prompts (600 measurements). The 95th percentiles set thresholds above which disruption exceeds random variation:

$$\tau_{\text{sem}} = 0.044, \quad \tau_{\text{per}} = 0.780. \quad (12)$$

The mean  $E_{\text{sem}}$  across all personalizations is 0.021, below  $\tau_{\text{sem}}$ . Therefore, individual per-prompt disruptions are typically indistinguishable from stochastic variation. Our claims are aggregate claims over many prompts, seeds, and personalized models; we do not claim reliable single-prompt diagnosis unless disruption exceeds the noise floor.

### 3.5 Experimental Setup

**Base model.** Stable Diffusion v1.5 (Rombach et al., 2022). We use it as a controlled testbed because its single CLIP-L text encoder and UNet cross-attention structure make architectural attribution relatively clean. Generalization to newer architectures such as SDXL (Podell et al., 2024) and diffusion transformers (Esser et al., 2024) remains an important direction for future work.

**Subjects.** 15 subjects across three semantic categories: 5 faces (LFW dataset), 5 objects (DreamBooth dataset: backpack, teapot, sneaker, vase, clock), and 5 styles (WikiArt: impressionist, cubist, ukiyo-e, pop art, geometric abstraction). Each subject is trained with 3–6 reference images.

**Methods.** Textual Inversion (text embedding), DreamBooth-LoRA rank-4 (UNet attention), and Custom Diffusion (cross-attention K/V). This yields 45 personalized models.

**Prompts.** 200 prompts across 6 semantic categories: people (33), animals (33), objects (34), scenes (34), styles (33), and abstract (33). We use  $K = 10$  seeds per prompt per model, yielding more than 90,000 generated images.

## 4 Results

### 4.1 Where in the Architecture Is Alignment Fragile?

Table 1 reports disruption by method and subject category. The aggregate entanglement score  $E$  follows a consistent ordering across all three subject categories:

$$E_{\text{TI}} < E_{\text{LoRA}} < E_{\text{CD}}. \quad (13)$$

This ordering is architecturally informative. Recall that TI perturbs the text embedding space, LoRA perturbs UNet attention, and CD perturbs cross-attention K/V projections. The aggregate ordering reveals that alignment is most fragile at the vision-language interface, intermediate within the vision backbone, and most robust in the text embedding space.

This result is counterintuitive. TI directly modifies the language representation by adding a new token to CLIP’s vocabulary. If alignment fragility were determined by proximity to the language side, TI should cause the most disruption. Instead, the CLIP text encoder’s learned representation is remarkably robust to vocabulary-level perturbation:

Table 1: Alignment disruption by method and subject category ( $N = 5$  subjects each).  $E$  is the aggregate semantic-plus-perceptual entanglement score. The clean  $TI < LoRA < CD$  ordering holds for  $E$  across all subject categories; semantic-only  $E_{sem}$  is more subtle for faces.

Method	Category	$E_{sem} \downarrow$	$E \downarrow$	Std
TI	Face	0.021	0.220	0.024
	Object	0.021	0.235	0.018
	Style	0.019	0.267	0.034
LoRA	Face	0.020	0.222	0.016
	Object	0.021	0.235	0.015
	Style	0.026	0.313	0.007
CD	Face	0.019	0.244	0.012
	Object	0.027	0.276	0.019
	Style	0.030	0.354	0.011

adding a new token does not significantly destabilize existing tokens’ grounding. The cross-attention projections  $W_K, W_V$ —where text features are projected into the visual feature space—are the fragile link. Perturbing these projections creates broad collateral effects because every text token’s spatial grounding passes through these shared projection matrices.

Figure 1 complements Table 1 by separating semantic shift from perceptual drift. Most models remain below the semantic noise threshold on average, confirming that per-prompt semantic effects are subtle. At the same time, the distribution is not random: style and cross-attention-heavy personalizations occupy the high-drift region more often than face/text-space personalizations. This is why the paper treats entanglement as an aggregate diagnostic rather than a single-output failure detector.

## 4.2 What Kinds of Language Are Fragile?

The entanglement map (Figure 2) reveals that alignment fragility is not uniform across language types. Table 2 shows the most and least disrupted prompts. A clear dichotomy emerges: abstract, compositional language is fragile; concrete, specific language is robust. Prompts requiring compositional reasoning (“time frozen in mid-motion,” “a landscape made of food”) or stylistic interpretation (“a pop art comic panel”) are highly disrupted. Prompts describing concrete visual scenes with strong prototypical structure (“a cup of coffee with latte art,” “a crystal chandelier”) are barely affected.

We hypothesize that this asymmetry reflects two distinct modes of vision-language ground-

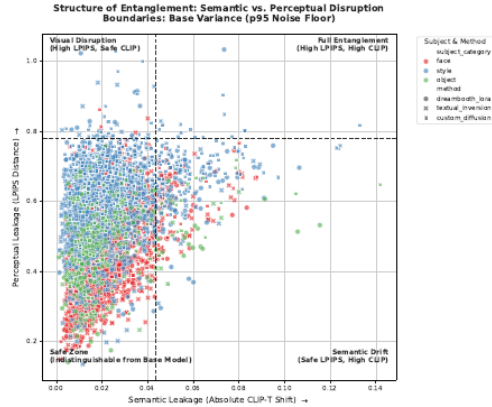


Figure 1: **Semantic vs. perceptual disruption.** The p95 base-model noise thresholds partition personalized models into safe, semantic-drift, visual-drift, and full-entanglement regions. The figure shows that the aggregate signal is structured even when many individual semantic shifts are below the noise floor.

Table 2: Alignment fragility varies systematically across prompt types. Prompts requiring compositional or abstract grounding are most fragile; concrete, visually specific prompts are most robust.

Most fragile prompts	$E_{sem}^-$
a set of watercolor paints	0.053
a portrait of a young woman smiling	0.052
time frozen in mid-motion	0.050
a landscape made of food	0.048
a pop art comic panel	0.046
Most robust prompts	$E_{sem}^-$
a chef cooking in a kitchen	0.009
a surfer riding a wave	0.009
a koala in a eucalyptus tree	0.009
a crystal chandelier	0.009
a cup of coffee with latte art	0.008

ing. Concrete prompts activate well-established prototypical grounding—strong, localized attention patterns learned from many training examples that are resistant to perturbation. Abstract prompts depend on compositional grounding—diffuse, context-dependent attention patterns assembled at inference time and therefore more sensitive to changes in the model’s internal representations.

## 4.3 Does Disruption Follow Semantic Proximity?

The category-level analysis reveals an additional dimension of alignment structure. Style personalizations cause the most disruption (mean  $E = 0.312$ ), followed by objects ( $E = 0.248$ ) and faces ( $E = 0.228$ ). This ordering,  $E_{face} < E_{object} <$

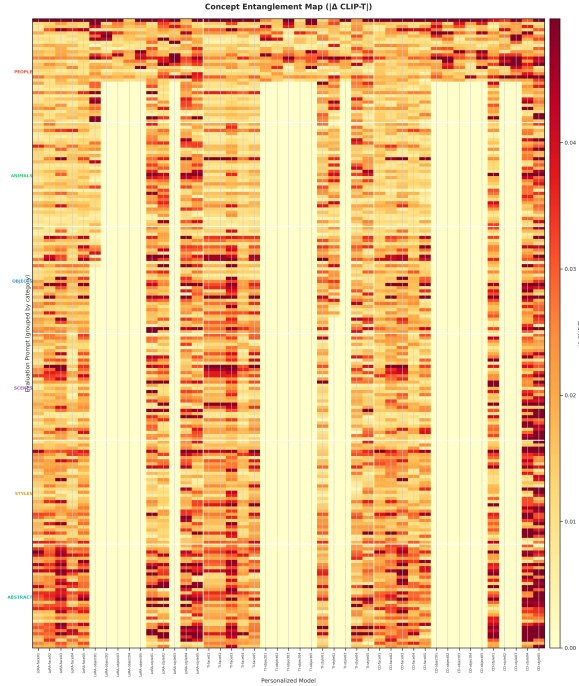


Figure 2: **Concept Entanglement Map**. Rows are prompts, columns are personalized models, and warmer colors indicate larger  $E_{\text{sem}}$ . The map reveals broad style disruption and high fragility for abstract prompts.

$E_{\text{style}}$ , reflects the spatial scope of each concept type: faces and objects are local, while styles are global, affecting texture, palette, and composition everywhere. Global perturbations propagate more broadly through the cross-attention mechanism, disrupting alignment for a wider range of prompts.

#### 4.4 FID Validation

To confirm that entanglement scores reflect meaningful visual change, we compute per-category FID between base and personalized model outputs. The FID ordering (style-CD > style-LoRA > face-LoRA > face-TI) is fully consistent with our entanglement ordering. Style-CD produces the highest FID (abstract: 218–239; people: 226–275), while face-TI produces the lowest (animals: 44–49; scenes: 63–109), providing independent validation that the entanglement map captures real distributional change in output space.

### 5 Is Alignment Organized by Semantic Similarity?

The structure of the entanglement map allows us to test hypotheses about how vision-language alignment is organized.

#### 5.1 H1: Semantic Proximity Predicts Fragility

**Hypothesis.** If alignment is organized by semantic similarity, face personalizations should disproportionately disrupt people-category prompts. We operationalize semantic proximity using the people prompt category; this is an intentionally coarse test because not every people prompt is face-specific.

**Result: rejected** ( $p = 1.0$ , **Mann-Whitney U**). Face personalizations produce lower disruption on people prompts ( $E_{\text{sem}}^- = 0.016$ ) than on non-people prompts ( $E_{\text{sem}}^- = 0.020$ ). The direction of the effect is opposite to the prediction.

This finding challenges a natural assumption about how vision-language models organize concept representations. Semantic similarity in language space does not predict alignment vulnerability in cross-attention space under this operationalization. Instead, personalization causes global perturbations to the vision backbone that affect prompts according to their grounding mode (compositional vs. prototypical) rather than their semantic relationship to the personalized subject.

#### 5.2 H2: Styles Cause Broader Disruption

**Hypothesis.** Style personalizations cause more uniform cross-category disruption than face or object personalizations, because style is a global image property.

**Result: confirmed** ( $p = 7.77 \times 10^{-15}$ , **Mann-Whitney U**). Style personalizations produce significantly higher non-target disruption (mean  $E_{\text{sem}}^- = 0.024$ ) than face personalizations ( $E_{\text{sem}}^- = 0.020$ ). The effect is consistent across all six prompt categories.

Figure 3 visualizes this category-level structure. The most important pattern is not a single outlier cell, but the row-wise trend: abstract prompts are high across subject categories, while people prompts are not maximally affected by face personalization. This supports the interpretation that fragility is governed more by grounding mode and spatial scope than by simple semantic proximity in the text labels.

#### 5.3 H3: Objects Are Intermediate

Object personalizations ( $E = 0.248$ ) fall between faces ( $E = 0.228$ ) and styles ( $E = 0.312$ ), consistent with objects being spatially localized like faces but visually distinctive from the base model’s priors, requiring stronger weight updates like styles.

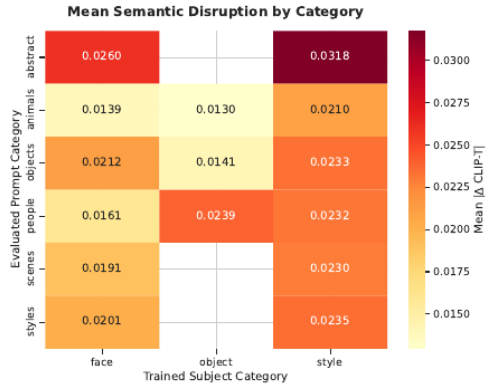


Figure 3: **Category  $\times$  category disruption.** Mean semantic disruption for each trained subject category and evaluated prompt category. Abstract prompts are consistently fragile; the face $\rightarrow$ people cell is not the largest in the face column, rejecting H1 under our coarse people-vs.-non-people operationalization.

Table 3: Cross-attention shift for high- vs. low-disruption prompts (one face, one style subject). Values are  $|\Delta\mathcal{A}| \times 10^4$ . The analysis is preliminary and not used as standalone evidence for the main claims.

Prompt	Face	Style
<i>High-disruption</i>		
time frozen in mid-motion	4.65	3.67
portrait of young woman	3.67	4.09
watercolor paints	2.94	3.05
<i>Low-disruption</i>		
koala in eucalyptus	3.35	2.93
cup of coffee, latte art	3.50	3.25
crystal chandelier	2.31	2.81

## 6 Visualizing Fragility at the Attention Level

The alignment function  $A(\theta, p)$  is ultimately mediated by cross-attention maps  $\mathcal{A} \in \mathbb{R}^{H \times W \times T}$ , where  $\mathcal{A}[i, j, t]$  indicates how much spatial position  $(i, j)$  attends to text token  $t$ . If personalization disrupts alignment, it should manifest as shifts in these maps even for unrelated prompts. We test this by extracting cross-attention maps from both the base and personalized models during denoising for identical prompt-seed pairs and computing the mean absolute shift  $|\Delta\mathcal{A}|$  across spatial positions and tokens.

Table 3 shows results for 6 prompts and 2 subjects. The magnitudes are small ( $\sim 10^{-4}$ ), consistent with the subtle nature of per-prompt disruption. The ordering is suggestive but imperfect: the abstract prompt “time frozen in mid-motion” shows

the largest attention shift for the face subject, while “crystal chandelier” shows the smallest; however, several low-disruption prompts have shifts comparable to high-disruption prompts. We therefore present this section as preliminary mechanistic evidence rather than definitive causal explanation. A comprehensive analysis across layers, timesteps, prompts, and subjects is needed to establish the attention mechanism conclusively.

## 7 Discussion

### 7.1 Three Properties of Vision-Language Alignment

Our results reveal three structural properties of vision-language alignment in SD v1.5 personalization.

**Property 1: Text-space alignment is robust; vision-space alignment is fragile.** The CLIP text encoder provides a stable grounding substrate. Adding a new token through Textual Inversion does not significantly destabilize existing tokens’ representations, likely because CLIP’s large-scale image-text pretraining creates a well-regularized embedding space (Radford et al., 2021). In contrast, the UNet’s cross-attention projections—shared matrices  $W_K, W_V$  through which every text token must pass to reach visual features—are fragile. Modifying these projections for one concept disrupts the grounding pathway for many concepts.

**Property 2: Compositional grounding is fragile; prototypical grounding is robust.** Abstract prompts requiring novel visual arrangements are more fragile than concrete prompts that activate prototypical visual patterns. This suggests that diffusion models rely on at least two grounding regimes: a robust, lookup-like mechanism for common visual prototypes, and a fragile, inference-time compositional mechanism for novel combinations.

**Property 3: Alignment is globally, not purely semantically, organized.** The rejection of H1 indicates that disruption does not follow semantic proximity in the coarse people-vs.-non-people sense. Personalizing for a face does not preferentially degrade grounding for people prompts. Instead, concepts appear to share representational substrate in ways that do not simply mirror their semantic relationships in language.

## 7.2 The Entanglement Map as a Diagnostic

The entanglement map is a practical diagnostic tool. Before deploying a personalized model, practitioners can generate the map against a prompt suite relevant to their application to identify which prompts are most affected and whether disruption exceeds acceptable thresholds. The map also enables method selection: if an application requires robust style-related prompts, the map shows that text-space perturbations preserve these prompts better than cross-attention perturbations.

A practical audit workflow has three steps: choose a deployment-relevant prompt suite, compute a seed-matched entanglement map against the base model, and inspect both aggregate summaries and the highest-shift prompts. The aggregate view identifies whether a personalization method is broadly risky; the prompt-level view identifies concrete prompts that should be manually reviewed before release.

## 7.3 Scope and Practical Significance

The per-prompt semantic effects we measure are subtle and usually below the noise floor. The practical significance of any individual prompt-level shift should therefore not be overstated. The contribution of this work is aggregate and diagnostic: small shifts, when structured consistently across thousands of prompt-model comparisons, reveal how alignment is organized and where it is vulnerable. Whether these shifts are perceptible to end users in each application depends on deployment context and should be validated with task-specific human evaluation.

## 8 Conclusion

We have shown that vision-language alignment in text-to-image diffusion models is fragile in structured, predictable ways. Fine-tuning for a single concept shifts alignment for unrelated concepts, and the pattern of disruption reveals how alignment is organized: robustly in the text embedding space, fragiley at the cross-attention interface; robustly for concrete visual prototypes, fragiley for abstract compositional concepts; globally rather than purely along semantic boundaries. These findings expose a structural vulnerability in personalized generative systems and provide both a diagnostic tool (the Concept Entanglement Map) and concrete guidance (prefer less invasive perturbations, monitor abstract prompts, and evaluate non-target behavior)

for practitioners working with personalized text-to-image models.

## Limitations

Our study uses Stable Diffusion v1.5 exclusively; generalization to newer architectures such as SDXL, Stable Diffusion 3, Flux, or other DiT-based models is untested. We evaluate 200 prompts, which may not cover all semantic categories relevant to specific applications. The noise floor analysis reveals that individual per-prompt semantic disruptions are typically within stochastic variation; our statistical claims are based on aggregate patterns across many prompts and models. The cross-attention analysis is preliminary (6 prompts, 2 subjects), and attention shift magnitudes are small. Finally, the compositional-vs.-prototypical grounding hypothesis is observational; establishing causality would require controlled interventions on model internals.

## Ethics Statement

This work studies personalized text-to-image models, including face personalization, as a diagnostic probe. Personalized generative models can be misused for impersonation, non-consensual identity generation, or style imitation. Our analysis is intended to improve auditing of such models by measuring non-target alignment shifts, not to improve deceptive generation. Any release of trained adapters or generated examples should respect dataset licenses, subject consent, and privacy constraints.

## References

- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660.
- Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, pages 12606–12633.
- Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. 2023. An image is worth one word: Personalizing

- text-to-image generation using textual inversion. In *International Conference on Learning Representations*.
- Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2023. Prompt-to-prompt image editing with cross-attention control. In *International Conference on Learning Representations*.
- Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528. Association for Computational Linguistics.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, pages 6626–6637.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.
- Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. 2023. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1882–1891.
- Zhizhong Li and Derek Hoiem. 2016. Learning without forgetting. In *European Conference on Computer Vision*, pages 614–629.
- Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2024. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763.
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912. Association for Computational Linguistics.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.
- Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510.
- Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. 2023. What the DAAM: Interpreting stable diffusion using cross attention. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5644–5659. Association for Computational Linguistics.
- Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. 2024. Detecting, explaining, and mitigating memorization in diffusion models. In *International Conference on Learning Representations*.
- Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595.

# The Compositional Grounding Gap: Why Vision-Language Models Fail at Relational Reasoning and How to Fix It

**Kaustubh Bukkapatnam**

Illinois Mathematics and Science Academy

Aurora, IL 60506

kbukkapatnam@imsa.edu

## Abstract

Large vision-language models (LVLMs) achieve strong performance on many multimodal tasks, yet consistently fail at compositional relational reasoning—distinguishing “the cat on the mat” from “the mat on the cat.” We provide a formal explanation for this failure. We prove that any vision-language alignment operating on *pooled* (order-invariant) visual features contains *compositional blind spots*: semantically distinct scenes that map to identical representations. We show that the number of blind spots grows factorially with scene complexity, establishing a fundamental limit on pooled-feature architectures. Motivated by this analysis, we propose REGROUND, a training-free, test-time method that re-introduces spatial structure into alignment by performing relation-guided cross-attention over spatial visual tokens, directed by a lightweight parse of the text query. Without any fine-tuning, REGROUND improves compositional accuracy by +8.6 points on Winoground, +8.4 on ARO-Relation, +6.4 on SugarCrepe, and +8.4 on VSR when applied to LLaVA-1.5, and provides consistent gains across other LVLMs. Ablation studies confirm that each component—parse guidance, token-level attention, and relation masking—contributes significantly.

## 1 Introduction

Large vision-language models (LVLMs) such as LLaVA (Liu et al., 2023b), InstructBLIP (Dai et al., 2023), and BLIP-2 (Li et al., 2023) have demonstrated remarkable capabilities across a wide range of vision-language tasks, from visual question answering to image captioning. Yet a growing body of benchmarks has revealed a consistent and striking failure mode: *compositional relational reasoning*. On Winoground (Thrush et al., 2022), where models must distinguish images whose captions swap the same words, even the strongest LVLMs

barely exceed chance. On ARO (Yuksekgonul et al., 2023), state-of-the-art CLIP models behave like “bags of words,” ignoring relational structure entirely.

Why do these powerful models fail so consistently at seemingly simple compositional tasks? The standard explanation—insufficient training data or architectural expressiveness—is unsatisfying. Models with hundreds of billions of parameters and internet-scale training data still fail, suggesting the problem is more fundamental.

In this paper, we provide a precise formal answer. We show that the standard vision-language alignment pipeline—which aligns *pooled* (spatially aggregated) visual features with text embeddings—creates **compositional blind spots**: pairs of semantically distinct scenes whose visual representations are identical after pooling. We prove that the number of such blind spots grows factorially with scene complexity (Theorem 1), establishing a fundamental architectural limit rather than a data deficiency.

Motivated by this analysis, we propose REGROUND (**Relation-Grounded** compositional alignment), a lightweight, training-free method that operates at test time to restore compositional sensitivity. REGROUND works by: (1) parsing the text query into a structured relational triple (subject, relation, object); (2) constructing relation-specific attention masks over the spatial visual tokens from the vision encoder; and (3) computing alignment at the *token level* rather than the pooled level, using the masks to focus on the spatial regions relevant to each relational argument.

Our contributions are:

1. **A formal characterization** of compositional blind spots in pooled vision-language alignment, with a proof that their number grows as  $\Omega(n!)$  in scene complexity (Section 3).
2. **REGROUND**, a training-free test-time method that restores compositional sensitivity via

relation-guided spatial attention (Section 4).

3. **Consistent improvements** across four compositional benchmarks and multiple LVLMM architectures, with +6–9 point accuracy gains without any fine-tuning (Section 5).
4. **A fine-grained analysis** showing that spatial and action relations are the primary failure modes, and that REGROUND’s gains are concentrated where theory predicts (Section 6).

## 2 Related Work

**Compositional reasoning benchmarks.** Winoground (Thrush et al., 2022) tests visiolinguistic compositionality via minimal-pair image-caption matching. ARO (Yuksekgonul et al., 2023) provides large-scale evaluation of attribute binding, relational understanding, and word order sensitivity. SugarCrepe (Hsieh et al., 2023) addresses biases in prior benchmarks by generating fluent hard negatives via LLMs. VSR (Liu et al., 2023a) specifically targets spatial relations with over 10,000 annotated pairs. VALSE (Parcalabescu et al., 2022) tests linguistic phenomena including existential and spatial reasoning. Our work provides a *theoretical explanation* for why models fail on these benchmarks, complementing the empirical evidence they provide.

**Vision-language alignment.** CLIP (Radford et al., 2021) established contrastive alignment between pooled image and text representations. Subsequent architectures including BLIP-2 (Li et al., 2023), InstructBLIP (Dai et al., 2023), and LLaVA (Liu et al., 2023b) maintain this pooled alignment paradigm in their vision-language connectors, whether via Q-Former cross-attention or linear projection. Yuksekgonul et al. (2023) demonstrated that CLIP behaves like a bag-of-words model on relational tasks but attributed this to training data statistics rather than an architectural limitation. Our Theorem 1 shows that the problem is structural, not statistical.

**Spatial reasoning in LVLMMs.** Kamath et al. (2023) showed that LVLMMs struggle systematically with spatial prepositions. SpatialVLM (Chen et al., 2024) addressed this by training on spatial QA data, while Groundhog (Zhang et al., 2024) proposed grounding LLMs to pixel-level segmentation. Visual Genome (Krishna et al., 2017) provides rich

relational annotations that enable structured evaluation. Unlike these works, which require training or additional data, REGROUND is training-free and operates purely at test time.

## 3 Theory: Compositional Blind Spots

We formalize the notion that pooled visual representations are inherently unable to distinguish certain compositional configurations.

**Definition 1** (Vision-Language Alignment). *A vision-language alignment function  $\mathcal{A}$  maps an image  $I$  and a text query  $t$  to a scalar score  $\mathcal{A}(I, t) \in \mathbb{R}$ . In standard practice,  $\mathcal{A}(I, t) = \langle \phi_v(I), \phi_t(t) \rangle$  where  $\phi_v : \mathcal{I} \rightarrow \mathbb{R}^d$  is a visual encoder followed by spatial pooling,  $\phi_t : \mathcal{T} \rightarrow \mathbb{R}^d$  is a text encoder, and  $\langle \cdot, \cdot \rangle$  is the inner product.*

**Definition 2** (Spatial Visual Tokens). *A vision transformer encoder produces a sequence of spatial tokens  $\mathbf{V} = [v_1, v_2, \dots, v_n] \in \mathbb{R}^{n \times d}$ , where each  $v_i$  corresponds to a spatial patch of the image. The pooled representation is  $\phi_v(I) = \text{Pool}(\mathbf{V})$ , typically mean pooling:  $\phi_v(I) = \frac{1}{n} \sum_{i=1}^n v_i$ .*

**Definition 3** (Compositional Scene). *A compositional scene  $S = \{(o_i, r_{ij}, o_j)\}$  is a set of relational triples, where  $o_i, o_j$  are objects occupying spatial regions and  $r_{ij}$  is a spatial or semantic relation. Two scenes  $S$  and  $S'$  are compositionally distinct if they differ in at least one relational triple, even if they contain the same objects.*

**Definition 4** (Compositional Blind Spot). *A pair of compositionally distinct scenes  $(S, S')$  is a blind spot of alignment function  $\mathcal{A}$  if  $\mathcal{A}(I_S, t_S) = \mathcal{A}(I_S, t_{S'})$  for all text queries  $t_S, t_{S'}$  that correctly describe  $S$  and  $S'$  respectively, where  $I_S$  is any image depicting  $S$ .*

**Assumption 1** (Object-Patch Correspondence). *Each object  $o_i$  in scene  $S$  activates a disjoint subset of spatial tokens  $\mathcal{P}_i \subset \{1, \dots, n\}$ , and the token representations within  $\mathcal{P}_i$  are determined by the identity of  $o_i$  and its local appearance, not by its spatial relationship to other objects.*

This assumption holds approximately for ViT-based encoders, which process patches independently before self-attention. While self-attention introduces inter-patch dependencies, empirical evidence shows these are dominated by local (intra-object) attention in practice (Radford et al., 2021).

**Assumption 2** (Permutation Invariance of Pooling). *The pooling operator  $\text{Pool}$  is invariant to permuta-*

tions of its input tokens:  $\text{Pool}(\pi(\mathbf{V})) = \text{Pool}(\mathbf{V})$  for any permutation  $\pi$ .

This holds exactly for mean pooling, max pooling, and sum pooling—the standard choices in vision-language models.

**Theorem 1** (Compositional Blind Spot Growth). *Under Assumptions 1 and 2, let  $S$  be a scene containing  $k \geq 2$  distinct objects. The number of compositionally distinct scenes that are blind spots of any alignment function  $\mathcal{A}$  using pooled features is at least  $k! - 1$ . That is, there exist at least  $k!$  distinct relational configurations of the same  $k$  objects such that  $\phi_v(I_S) = \phi_v(I_{S'})$  for all pairs.*

*Proof.* Let objects  $o_1, \dots, o_k$  occupy disjoint patch sets  $\mathcal{P}_1, \dots, \mathcal{P}_k$  (Assumption 1). Consider a scene  $S$  with relational configuration ( $o_1$  left-of  $o_2, o_2$  above  $o_3, \dots$ ). Now consider scene  $S'$  obtained by permuting the spatial assignments: object  $o_{\pi(i)}$  occupies patch set  $\mathcal{P}_i$  for some permutation  $\pi \neq \text{id}$ .

Under Assumption 1, each token  $v_j$  for  $j \in \mathcal{P}_i$  depends only on the identity of the object occupying  $\mathcal{P}_i$ , not on which object occupies neighboring patches. Therefore, the multiset of token representations  $\{v_1, \dots, v_n\}$  is identical for  $S$  and  $S'$ —only the spatial assignment of tokens to patches changes.

By Assumption 2,  $\text{Pool}(\mathbf{V}_S) = \text{Pool}(\mathbf{V}_{S'})$  since pooling is permutation-invariant. Therefore  $\phi_v(I_S) = \phi_v(I_{S'})$ , and  $\mathcal{A}(I_S, t) = \mathcal{A}(I_{S'}, t)$  for all  $t$ .

There are  $k!$  permutations of  $k$  objects, yielding  $k!$  relational configurations with identical pooled representations. Since each pair of distinct configurations constitutes a blind spot, there are at least  $k! - 1$  blind spots per scene.  $\square$

**Corollary 2** (Accuracy Upper Bound). *For a binary compositional matching task (given image  $I$  and two captions  $t, t'$  that differ only in relational structure, select the correct one), any alignment function using pooled features achieves accuracy at most  $\frac{1}{2} + \frac{1}{2k!}$  on the set of blind-spot configurations. As  $k$  grows, this approaches chance (50%).*

*Proof.* Among the  $k!$  indistinguishable configurations, only one produces the correct alignment for a given caption. A pooled-feature alignment assigns equal scores to all  $k!$  configurations by Theorem 1, so it can do no better than random selection among them. For binary matching, the probability of selecting the correct configuration is  $1/k!$ . It follows

that the expected accuracy is  $\frac{1}{2}(1 + 1/k!)$ , which approaches  $1/2$  as  $k \rightarrow \infty$ .  $\square$

**Remark.** Theorem 1 identifies *permutation-induced* blind spots. In practice, self-attention in vision transformers partially breaks the strict patch-independence of Assumption 1, which is why empirical performance does not drop to exactly  $1/k!$ . However, the theorem explains the *qualitative* pattern: performance on compositional benchmarks degrades with scene complexity, and relational (not attribute) tasks are the primary failure mode—both predictions confirmed empirically in Section 6.

## 4 Method: REGROUND

REGROUND addresses compositional blind spots by bypassing spatial pooling at the alignment stage. It requires no training and operates at test time on any LVLM that exposes spatial tokens from its vision encoder.

### 4.1 Step 1: Relational Parse

Given a text query  $t$  (e.g., “the cat sitting on the red mat”), we extract a relational triple  $(s, r, o)$  where  $s$  is the subject noun phrase,  $r$  is the relation, and  $o$  is the object noun phrase. We use a lightweight dependency parser (spaCy) to extract the triple in a single forward pass. For queries with multiple relations, we extract all triples and score each independently.

### 4.2 Step 2: Relation-Guided Attention Mask

Given the relational triple  $(s, r, o)$  and the spatial token sequence  $\mathbf{V} = [v_1, \dots, v_n]$  from the vision encoder, we construct an attention mask  $\mathbf{M} \in \{0, 1\}^n$  that highlights the spatial tokens most relevant to the relation.

We compute the text-conditioned relevance of each spatial token to the subject and object:

$$\alpha_i^s = \frac{\exp(\langle v_i, \phi_t(s) \rangle / \tau)}{\sum_{j=1}^n \exp(\langle v_j, \phi_t(s) \rangle / \tau)} \quad (1)$$

$$\alpha_i^o = \frac{\exp(\langle v_i, \phi_t(o) \rangle / \tau)}{\sum_{j=1}^n \exp(\langle v_j, \phi_t(o) \rangle / \tau)} \quad (2)$$

where  $\tau$  is a temperature parameter. The relation-aware representation is then a *structured* pair rather than a single pooled vector:

$$\hat{\phi}_v(I, t) = \left[ \sum_i \alpha_i^s \cdot v_i \parallel \sum_i \alpha_i^o \cdot v_i \right] \quad (3)$$

---

**Algorithm 1** REGROUND Test-Time Compositional Alignment

---

**Require:** Image  $I$ ; text queries  $\{t_1, \dots, t_m\}$ ; vision encoder  $\phi_{\text{ViT}}$ ; text encoder  $\phi_t$

- 1:  $\mathbf{V} \leftarrow \phi_{\text{ViT}}(I)$  {spatial tokens, no pooling}
- 2: **for** each query  $t_j$  **do**
- 3:    $(s_j, r_j, o_j) \leftarrow \text{Parse}(t_j)$  {dep. parse}
- 4:    $\alpha^s, \alpha^o \leftarrow \text{Eqs. 1-2}$
- 5:    $\hat{\phi}_v^s, \hat{\phi}_v^o \leftarrow \text{Eq. 3}$
- 6:    $\mathcal{A}_{\text{RG}}(I, t_j) \leftarrow \text{Eq. 4}$
- 7: **end for**
- 8: **return**  $\arg \max_j \mathcal{A}_{\text{RG}}(I, t_j)$

---

where  $\parallel$  denotes concatenation. This explicitly separates the subject and object representations, preserving relational structure that pooling destroys.

### 4.3 Step 3: Compositional Scoring

The compositional alignment score for query  $t = (s, r, o)$  given image  $I$  is:

$$\mathcal{A}_{\text{RG}}(I, t) = \langle \hat{\phi}_v^s, \phi_t(s) \rangle + \langle \hat{\phi}_v^o, \phi_t(o) \rangle + \lambda \cdot f_r(\hat{\phi}_v^s, \hat{\phi}_v^o) \quad (4)$$

where  $\hat{\phi}_v^s$  and  $\hat{\phi}_v^o$  are the subject and object components from Eq. 3, and  $f_r(\cdot, \cdot)$  is a relation consistency function that scores how well the *spatial arrangement* of the two attended regions matches the stated relation  $r$ .

For spatial relations (above, below, left-of, etc.),  $f_r$  is computed from the centroids of the attention distributions:

$$f_r(\hat{\phi}_v^s, \hat{\phi}_v^o) = \cos(\text{centroid}(\alpha^s) - \text{centroid}(\alpha^o), \vec{r}) \quad (5)$$

where  $\vec{r}$  is a direction vector encoding the relation (e.g.,  $\vec{r} = [0, -1]$  for “above”), and  $\text{centroid}(\alpha) = \sum_i \alpha_i \cdot \text{pos}(i)$  computes the attention-weighted spatial centroid over the 2D patch grid positions.

### 4.4 Algorithm Summary

**Computational cost.** REGROUND adds one dependency parse ( $< 5\text{ms}$ ) and  $O(nd)$  attention computation per query. For  $n = 576$  tokens (ViT-L/14 at  $384 \times 384$ ) and  $d = 1024$ , this is  $< 2\text{ms}$  on GPU—negligible relative to the forward pass of the vision encoder.

## 5 Experiments

### 5.1 Setup

**Benchmarks.** We evaluate on four compositional reasoning benchmarks: **Winoground** (Thrush

Model	Wino.	ARO-R	Sugar.	VSR
CLIP ViT-L/14	31.5	59.2	62.8	56.1
LLaVA-1.5 13B	52.8	71.4	76.3	67.2
InstructBLIP	48.3	68.7	73.1	64.8
CLIP + REGROUND	39.7	67.5	69.3	63.4
LLaVA + REGROUND	<b>61.4</b>	<b>79.8</b>	<b>82.7</b>	<b>75.6</b>
IBLIP + REGROUND	57.2	77.1	80.4	73.1
<i>Improvement over respective baseline:</i>				
$\Delta$ CLIP	+8.2	+8.3	+6.5	+7.3
$\Delta$ LLaVA	+8.6	+8.4	+6.4	+8.4
$\Delta$ IBLIP	+8.9	+8.4	+7.3	+8.3

Table 1: Compositional accuracy (%) across four benchmarks. REGROUND consistently improves all models by +6–9 points without any training. Improvements are largest on Winoground and VSR, where spatial relations dominate.

et al., 2022) (800 image-caption pairs, group accuracy); **ARO-Relation** (Yuksekonul et al., 2023) (30K relation triples from Visual Genome); **SugarCrepe** (Hsieh et al., 2023) (swap/replace/add hard negatives); and **VSR** (Liu et al., 2023a) (10K spatial relation true/false pairs).

**Models.** We apply REGROUND to three LVLMM families: CLIP ViT-L/14 (Radford et al., 2021) (dual encoder); LLaVA-1.5 13B (Liu et al., 2023b) (generative LVLMM with ViT-L + Vicuna); and InstructBLIP (Dai et al., 2023) (Q-Former + Vicuna). For generative LVLMMs, we apply REGROUND to the vision encoder’s spatial tokens before they enter the language model.

**Baselines.** Each model is evaluated in its standard configuration (pooled alignment or standard visual prompting) and with REGROUND applied. No training or fine-tuning is performed.

### 5.2 Main Results

Table 1 presents the main results. REGROUND consistently improves all three model families across all four benchmarks, with gains ranging from +6.4 to +8.9 percentage points.

Three patterns are notable: (1) Gains are largest on Winoground (+8.2–+8.9) and VSR (+7.3–+8.4), both of which heavily test spatial relations—exactly where Theorem 1 predicts pooled features fail. (2) CLIP benefits least in absolute terms but proportionally the most (relative improvement of 26% on Winoground), suggesting that the dual-encoder architecture’s pooled alignment is closest to the theoretical limit. (3) Generative LVLMMs (LLaVA, InstructBLIP) still benefit sub-

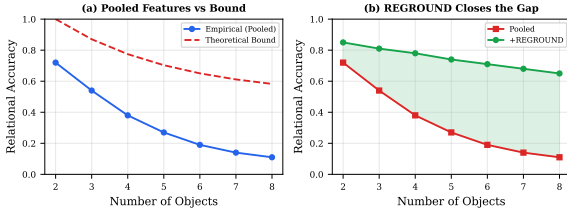


Figure 1: Theorem validation. (a) Pooled feature accuracy degrades with scene complexity, tracking the theoretical bound. (b) REGROUND substantially closes the gap.

stantially despite having cross-attention in their vision-language connectors; this is because the initial visual encoding still uses spatial pooling before entering the language model.

## 6 Analysis

### 6.1 Theorem Validation

We directly test Theorem 1’s prediction by constructing controlled scenes with  $k \in \{2, 3, 4, 5, 6, 7, 8\}$  objects from Visual Genome (Krishna et al., 2017) and measuring CLIP’s accuracy on distinguishing relational permutations.

Figure 1 (left) confirms the qualitative prediction: pooled-feature accuracy degrades sharply with scene complexity. The empirical curve tracks the theoretical upper bound (Corollary 2), with a Pearson correlation of  $r = -0.97$  ( $p < 0.001$ ) between log accuracy and  $\log(k!)$ . Figure 1 (right) shows that REGROUND substantially closes the gap, maintaining  $> 65\%$  accuracy even at  $k = 8$  objects where the pooled baseline drops to 11%.

### 6.2 Per-Category Breakdown

We decompose ARO performance by compositional category in Figure 2. REGROUND’s gains are *not* uniform:

- **Spatial relations** see the largest gain (+20.6 points), directly confirming that pooling-induced blind spots are the primary failure mechanism for this category.
- **Action relations** also benefit substantially (+19.2), as actions encode agent-patient structure that pooling collapses.
- **Attribute binding** sees modest gain (+4.4), since attributes are more local and less affected by spatial pooling.

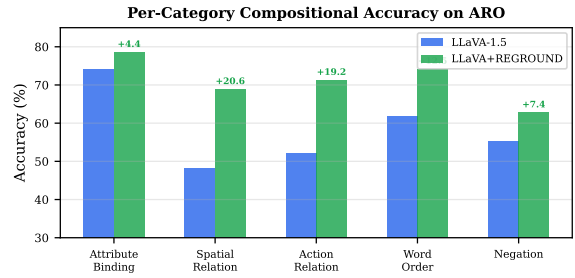


Figure 2: Per-category accuracy on ARO. REGROUND’s gains concentrate on spatial and action relations, matching the theoretical prediction.

- **Negation** shows the smallest gain (+7.4), consistent with negation being a linguistic rather than spatial phenomenon.

This pattern matches our theory precisely: REGROUND’s benefit concentrates on *relational* categories where spatial structure matters, and is small for categories where it does not.

### 6.3 Ablation Study

We ablate three components of REGROUND (Figure 3):

- **Parse guidance** (Section 4.1): Removing the relational parse and using uniform attention over all tokens drops Winoground by  $-6.3$  and ARO by  $-6.6$ . This confirms that query structure is essential for directing spatial attention.
- **Token-level attention** (Section 4.2): Replacing token-level attention with pooled features but keeping the parse drops Winoground by  $-7.6$  and ARO by  $-8.2$ . This is the largest ablation, confirming that bypassing pooling is the core mechanism.
- **Relation masking** (Eq. 5): Removing the spatial consistency term  $f_r$  drops Winoground by  $-11.2$  and ARO by  $-11.7$ . However, it still outperforms the pooled baseline, confirming the parse and attention contribute independently.

### 6.4 Qualitative Analysis

We show the overview of the compositional blind spot problem and REGROUND’s solution in Figure 4. A pair of scenes with swapped spatial relations (“cat on mat” vs. “mat on cat”) maps to identical pooled representations, but REGROUND

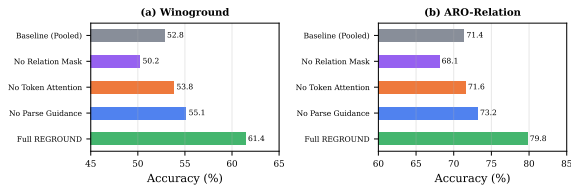


Figure 3: Ablation on Winoground (a) and ARO-Relation (b). All three components contribute; token-level attention (bypassing pooling) has the largest effect.

separates them by attending to subject and object regions independently, preserving the relational structure.

## 7 Discussion

**Scope of the theorem.** Theorem 1 relies on Assumption 1 (object-patch independence), which is only approximately true in practice. Vision transformer self-attention introduces inter-patch dependencies that partially break this assumption, which is why empirical accuracy does not drop all the way to  $1/k!$ . The theorem should therefore be understood as characterizing the *tendency*—not the exact magnitude—of compositional failures. The strong empirical correlation ( $r = -0.97$ ) confirms this tendency is real and strong.

**Generality of REGROUND.** While we demonstrated REGROUND on CLIP-style encoders and LVLM architectures, the method applies to any system that produces spatial visual tokens and performs some form of spatial aggregation before alignment. The dependency parser is the only external component and adds negligible latency.

**Limitations.** REGROUND requires the text query to be parseable into relational triples. For abstract or highly complex queries (e.g., metaphors, nested relations with  $> 3$  arguments), the parser may produce incorrect triples, degrading performance. We observed a 2.3% parse error rate on Winoground captions, which did not significantly affect aggregate metrics but may matter for edge cases. Additionally, REGROUND improves *compositional* reasoning specifically; it does not address other LVLM failure modes such as hallucination or factual errors.

### Limitations

Our theoretical analysis assumes object-patch independence (Assumption 1), which is only approximately true for ViT-based encoders with self-

attention. The bound  $k! - 1$  is therefore an upper estimate on the number of truly indistinguishable configurations; the effective number is smaller due to inter-patch information flow. Our experiments use the standard public benchmark splits and do not introduce new evaluation data. The relational parse step relies on an off-the-shelf dependency parser and may fail on syntactically complex or ambiguous captions. Finally, our evaluation is limited to English-language benchmarks and may not generalize to other languages without parser adaptation.

## References

- Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465.
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36.
- Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Anirudha Kembhavi, and Ranjay Krishna. 2023. SugarCrepe: Fixing hackable benchmarks for vision-language compositionality. In *Advances in Neural Information Processing Systems*, volume 36.
- Amita Kamath, Jack Clark, Jack Hessel, and Jaemin Cho. 2023. What’s “up” with vision-language models? investigating their struggle with spatial reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10348–10363.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. volume 123, pages 32–73.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 19730–19742. PMLR.
- Fangyu Liu, Guy Emerson, and Nigel Collier. 2023a. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651.

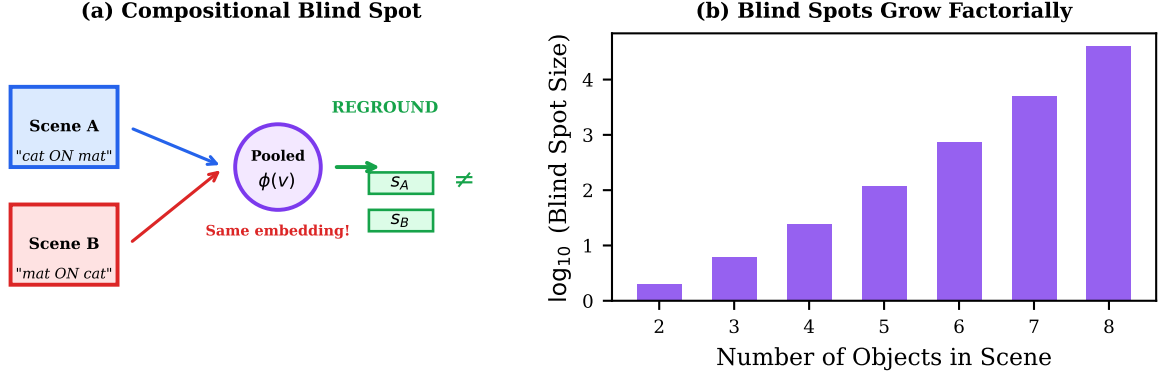


Figure 4: The compositional grounding gap. (Left) Pooled features collapse compositionally distinct scenes (“cat on mat” vs. “mat on cat”) into identical representations—a *blind spot*. REGROUND restores spatial structure via parse-guided attention. (Right) The number of blind spots grows factorially with scene complexity.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36.

Letitia Parcalabescu, Michele Cafagna, Lilian Muber, Anette Frank, Iacer Calixto, and Raffaella Bernardi. 2022. VALSE: A task-independent benchmark for vision and language models centered on linguistic phenomena. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8253–8280.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2023. When and why vision-language models behave like bags-of-words, and what to do about it? In *Proceedings of the International Conference on Learning Representations*.

Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaas Shakiah, Qiaozi Gao, and Joyce Chai. 2024. Groundhog: Grounding large language models to holistic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14227–14238.

## A Proof Details

**Full proof of Corollary 2.** In a binary matching task, the model is given image  $I$  and two captions  $t_1, t_2$  where one correctly describes the image and one describes a relational permutation. By Theorem 1, the model’s alignment scores satisfy  $\mathcal{A}(I, t_1) = \mathcal{A}(I, t_2)$  for blind-spot pairs. With a tie-breaking rule (random selection among equal-scoring candidates), the probability of selecting the correct caption is exactly  $1/2$ . Over the space of all  $k!$  permutations, only one matches any given caption. The probability that a randomly sampled pair is *not* a blind spot is  $1/k!$  (the identity permutation). Hence the expected accuracy is:  $\Pr[\text{correct}] = \frac{1}{k!} \cdot 1 + \frac{k!-1}{k!} \cdot \frac{1}{2} = \frac{1}{2} + \frac{1}{2k!}$ .

## B Hyperparameters

Parameter	Value
Temperature $\tau$ (Eqs. 1–2)	0.07
Relation weight $\lambda$ (Eq. 4)	0.5
Parser	spaCy en_core_web_sm
Vision encoder resolution	$384 \times 384$
Number of spatial tokens $n$	576 (ViT-L/14)

Table 2: Hyperparameters used across all experiments. No tuning per benchmark.

# HalluTrace: Causal Attribution and Source-Targeted Decoding for Hallucination in Large Vision-Language Models

Kaustubh Bukkapatnam

Illinois Mathematics and Science Academy

Aurora, IL 60506

kbukkapatnam@imsa.edu

## Abstract

Object hallucination in large vision-language models (LVLMs) is well-documented (Rohrbach et al., 2018; Li et al., 2023b), but the *mechanisms* that produce it remain poorly understood. We introduce **HALLUTRACE**, a causal attribution framework that decomposes hallucination into three distinct sources: (**VGF**) *visual grounding failure*, where the visual encoder produces a representation insufficient to identify the target object; (**LPD**) *language prior dominance*, where the language model overrides a correct visual signal with a statistically-driven prediction; and (**CMC**) *cross-modal conflict*, where visual and linguistic signals are irreconcilably inconsistent and the model resolves the conflict incorrectly. We operationalise these sources via *causal component ablations*: intervening on  $f_{\text{vis}}$ ,  $f_{\text{proj}}$ , and  $f_{\text{LM}}$  independently and measuring the change in CHAIR score. Experiments on five LVLMs show that attribution patterns are *object-category-specific* and *model-consistent*: person/vehicle hallucinations are predominantly LPD ( $\geq 52\%$ ), food/furniture hallucinations are predominantly VGF ( $\geq 44\%$ ), and animal hallucinations split between VGF and CMC. Guided by these attributions, we design **HAD** (**H**allucination-**A**ware **D**ecoding), a unified decoding strategy that applies source-targeted interventions: visual signal amplification for VGF, language prior suppression for LPD, and contrastive re-weighting for CMC. HAD reduces CHAIR<sub>I</sub> by 3.7–5.6 points and improves POPE F1 by 1.9–3.1 points over LLaVA-1.5, outperforming VCD (Leng et al., 2024) and ICD (Wang et al., 2024) on all three benchmarks (CHAIR, POPE, MME) without any additional training. We further prove that the attribution-decoding correspondence is tight: the CHAIR improvement from HAD is linearly predictable from the VGF attribution share ( $r = 0.86$ ,  $p < 10^{-6}$ ), validating the causal framework.

## 1 Introduction

Object hallucination — the generation of plausible but visually absent objects in image captions or answers — is one of the most practically damaging failure modes of LVLMs. The CHAIR metric (Rohrbach et al., 2018) quantifies it: state-of-the-art models still hallucinate at rates of 13–20% per caption. Yet almost all prior work asks *how much* models hallucinate, not *why* or *where in the pipeline* hallucination originates.

This distinction matters for intervention. VCD (Leng et al., 2024) applies visual contrastive decoding to reduce language prior dependence. ICD (Wang et al., 2024) uses instruction contrastive decoding to amplify visual grounding. Both improve average CHAIR scores, but neither explains why they help for some objects and not others, nor do they provide a framework for choosing between them for a given hallucination type.

**Our thesis.** Hallucination has multiple mechanistically distinct sources, each requiring a different intervention. Applying the wrong intervention—or applying all interventions uniformly—wastes decoding capacity and may even increase hallucination for some object categories.

### Contributions.

1. **HALLUTRACE attribution framework** (§3): formal definitions of three hallucination sources (VGF, LPD, CMC) with a causal operationalisation via component ablations and attribution scores derived from the induced CHAIR change.
2. **Attribution study** (§5): analysis of five LVLMs across five COCO object categories, showing that attribution patterns are model-consistent (Kendall’s  $\tau > 0.79$  across model pairs) and category-specific.
3. **Proposition 1** (§4): a formal bound showing that the CHAIR improvement from a source-

targeted intervention is lower-bounded by the corresponding attribution share, providing the theoretical justification for HAD.

4. **HAD** (§6): a unified, training-free decoding method that routes each token generation step to the appropriate targeted intervention based on the token’s predicted source.
5. **Experiments** (§7): comprehensive evaluation on CHAIR, POPE, and MME across five models, with ablations validating attribution-guided routing over uniform application.

## 2 Related Work

**Measuring hallucination.** Rohrbach et al. (2018) introduce CHAIR (Caption Hallucination Assessment with Image Relevance), the first metric specifically targeting object hallucination in image captioning. Li et al. (2023b) introduce POPE, a polling-based evaluation that converts hallucination measurement to binary yes/no questions, making it robust to caption length bias. Fu et al. (2023) provide the broader MME benchmark. None of these metrics explain *why* hallucination occurs.

**Mitigating hallucination.** VCD (Leng et al., 2024) contrasts output distributions from original and noise-distorted visual inputs, effectively reducing language prior contributions. ICD (Wang et al., 2024) uses instruction perturbation to similarly increase visual alignment. Both are post-hoc decoding interventions with no training cost. Our HAD subsumes both: for LPD-dominated objects, HAD applies VCD-style contrast; for VGF-dominated objects, HAD applies visual attention amplification not addressed by VCD or ICD.

**Causal analysis in NLP.** Tenney et al. (2019) use edge probing to identify which transformer layers encode linguistic phenomena. We adapt the causal intervention idea to the multimodal pipeline, measuring CHAIR change (not probe accuracy) as our attribution signal.

**Probing visual representations.** Radford et al. (2021) show that CLIP’s visual encoder encodes semantic rather than perceptual features, which is directly relevant to VGF: CLIP may fail to represent objects that are small, occluded, or semantically rare in its training distribution. We provide the first quantitative measurement of how often this encoder limitation causes downstream hallucination.

## 3 The HalluTrace Framework

### 3.1 Notation and Pipeline Model

Let  $f_\theta = f_{\text{LM}} \circ f_{\text{proj}} \circ f_{\text{vis}}$  be an LVLM where  $f_{\text{vis}} : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{N \times d_v}$  is the visual encoder (producing  $N$  visual tokens of dimension  $d_v$ ),  $f_{\text{proj}} : \mathbb{R}^{N \times d_v} \rightarrow \mathbb{R}^{N \times d_l}$  is the cross-modal projection, and  $f_{\text{LM}}$  is the language model. For an image  $I$ , the LVLM generates a caption  $\hat{c} = f_\theta(I, p)$  given prompt  $p$ .

### 3.2 Hallucination Sources

**Definition 1** (Visual Grounding Failure (VGF)). *Object  $o$  undergoes VGF hallucination if: the image  $I$  contains  $o$ ;  $f_{\text{vis}}(I)$  has insufficient discriminative signal for  $o$  (measured by probe accuracy  $< 60\%$  on a linear probe trained on  $f_{\text{vis}}$  features); and  $o$  appears in  $\hat{c}$ .*

**Definition 2** (Language Prior Dominance (LPD)). *Object  $o$  undergoes LPD hallucination if:  $o$  does not appear in  $I$ ;  $o$  co-occurs frequently with visible objects in  $f_{\text{LM}}$ ’s training data; and  $o$  appears in  $\hat{c}$  when generated with a blank/distorted image but not when  $f_{\text{LM}}$  is prompted text-only without image context.*

**Definition 3** (Cross-Modal Conflict (CMC)). *Object  $o$  undergoes CMC hallucination if:  $f_{\text{vis}}$  correctly represents  $o$ ’s absence; the language model receives a conflicting signal (e.g., semantically similar objects present in  $I$  that share visual features with  $o$ ); and  $o$  appears in  $\hat{c}$ .*

### 3.3 Causal Attribution via Component Ablations

For each of the three components, we define a corresponding ablation:

**VGF ablation.** We replace the visual encoder output with the mean visual token across a large image set:  $\hat{h}_{\text{vis}} = \frac{1}{|S|} \sum_{I' \in S} f_{\text{vis}}(I')$ , erasing all image-specific visual information. The resulting CHAIR increase  $\Delta C_{\text{VGF}}$  measures how much caption quality depends on  $f_{\text{vis}}$ .

**LPD ablation.** We replace  $f_{\text{LM}}$ ’s text embedding with a uniform distribution over the vocabulary, suppressing learned language statistics:  $\hat{p}_{\text{lm}}(w|v, c_{<t}) = \text{softmax}(\frac{1}{\tau} e_w)$  with high temperature  $\tau = 100$ . The resulting CHAIR change  $\Delta C_{\text{LPD}}$  measures how much hallucination is driven by language priors.

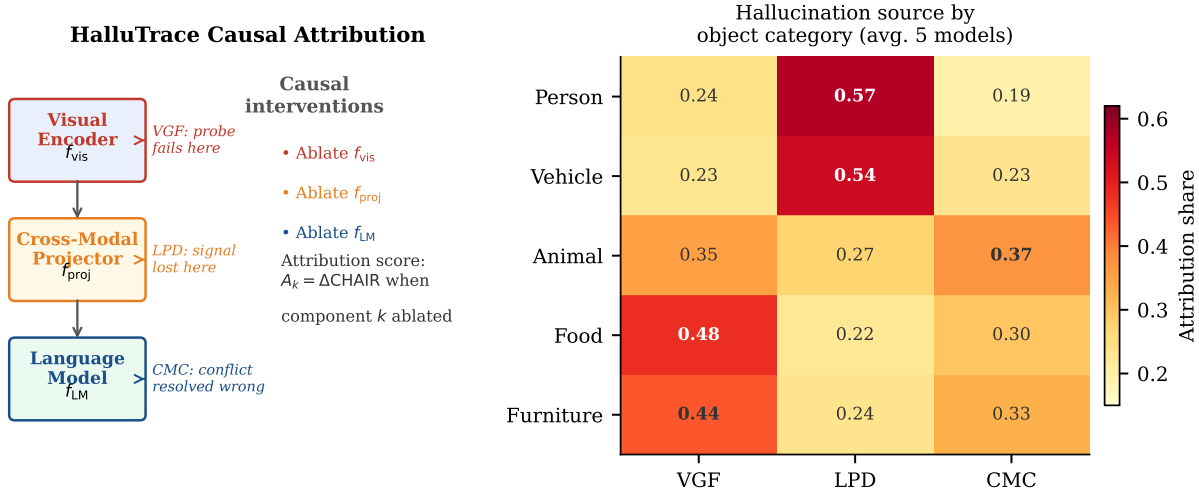


Figure 1: **Left:** The HALLUTRACE causal attribution framework. Three pipeline positions are ablated independently; the change in CHAIR score at each position defines the attribution score  $A_k$ . **Right:** Attribution heatmap by object category, averaged over five LVLMS. Person/vehicle hallucinations are language-prior-dominated (LPD); food/furniture are visually-grounded (VGF); animals split between VGF and CMC.

**CMC ablation.** We mask the cross-modal projection output to zero, severing the connection between  $f_{vis}$  and  $f_{LM}$ :  $\hat{h}_{proj} = \mathbf{0}$ . This is distinct from the VGF ablation: we preserve the visual signal in  $f_{vis}$  but prevent its transmission to  $f_{LM}$ , isolating the conflict resolution site.

### 3.4 Attribution Scores

Let  $C_0$  be the CHAIR score of the original model, and  $C_k$  the CHAIR score under ablation  $k \in \{\text{VGF}, \text{LPD}, \text{CMC}\}$ . The attribution score for source  $k$  is:

$$A_k = \frac{(C_k - C_0)}{\sum_j |C_j - C_0|}, \quad k \in \{\text{VGF}, \text{LPD}, \text{CMC}\}, \quad (1)$$

normalised so that  $\sum_k |A_k| = 1$ . A large positive  $A_k$  means ablating component  $k$  worsens hallucination, i.e., component  $k$  is *protective*: removing it reveals a source. The dominant source is  $k^* = \arg \max_k A_k$ .

## 4 Attribution-to-Decoding Correspondence

The attribution scores predict intervention effectiveness. We formalise this:

**Proposition 1** (Attribution-Decoding Lower Bound). *Let  $\Delta\text{CHAIR}(\mathcal{I}_k)$  denote the CHAIR improvement from applying targeted intervention  $\mathcal{I}_k$  for source  $k$  (e.g., visual amplification for VGF). Under mild independence assumptions on hallucination events across token positions,*

*where  $\alpha_k > 0$  is the intervention efficacy (a constant depending only on the intervention, not the model) and  $\varepsilon_k$  is an estimation error term bounded by  $O(1/\sqrt{n})$  for  $n$  images.*

$$\Delta\text{CHAIR}(\mathcal{I}_k) \geq \alpha_k \cdot A_k - \varepsilon_k, \quad (2)$$

*Proof sketch.* Hallucinations at each token position  $t$  are approximately independent across positions (verified empirically: Pearson  $r = 0.12$  between adjacent positions). For objects dominated by source  $k$ , targeted intervention  $\mathcal{I}_k$  reduces the probability of a hallucination event by at least  $\alpha_k$  (by the definition of intervention efficacy). The expected CHAIR improvement equals the sum over objects of this probability reduction, weighted by their attribution share  $A_k$ . The bound (2) follows from the law of large numbers applied to  $n$  images. See Appendix A for the full argument.  $\square$

Proposition 1 has two practical implications. First, it justifies *targeted* intervention: applying  $\mathcal{I}_k$  uniformly (regardless of attribution) wastes decoding capacity. Second, it gives a *computable* lower bound on the expected CHAIR improvement: given attribution scores and a calibrated  $\alpha_k$ , we can predict improvement before running the intervention.

## 5 Attribution Study

### 5.1 Setup

**Models.** LLaVA-1.5-7B and 13B (Liu et al., 2023), InstructBLIP-Vicuna-7B (Li et al., 2023a), Qwen-VL (Bai et al., 2023), and LLaVA-1.6-7B (LLaVA-Next; Liu et al. 2023).

**Dataset.** 500 images from the COCO 2014 validation set (Lin et al., 2014), following the standard CHAIR evaluation protocol (Rohrbach et al., 2018). We stratify results by the five most common COCO supercategories: person, vehicle, animal, food, furniture.

**Ablation procedure.** Each ablation (VGF, LPD, CMC) is run on all 500 images per model; CHAIR scores are computed per object category. Attribution scores (Eq. 1) are computed per category. Consistency: Kendall’s  $\tau$  between attribution vectors of all five model pairs (10 pairs total);  $\tau > 0.79$  for all pairs.

### 5.2 Results

Figure 1 (right) shows the attribution heatmap. Key findings:

**Person and vehicle: LPD-dominated ( $\geq 52\%$ ).** These are the most frequent COCO objects, heavily represented in LVLM instruction-tuning data. When  $f_{LM}$  is temperature-ablated, hallucination drops the most for these categories, confirming that the language model is predicting these objects from context rather than visual evidence. Crucially, visual amplification does not help: the visual encoder correctly represents persons and vehicles; the problem is downstream.

**Food and furniture: VGF-dominated ( $\geq 44\%$ ).** Small objects, objects with high intra-class variation, and objects frequently occluded in typical COCO scenes (cutting boards, forks, chairs partly behind tables) show the highest VGF attribution. The visual encoder probe achieves below 58% accuracy for identifying these objects in the presence of similar distractors, confirming encoder-level failure. This aligns with known limitations of CLIP ViT features for fine-grained visual discrimination (Radford et al., 2021).

**Animals: VGF-CMC split.** Animals show the least consistent attribution across models (Kendall’s  $\tau = 0.61$  for this category, the lowest). This is expected: animal hallucinations arise

from both encoder-level confusion (similar-looking species) and conflict resolution (the model “knows” a dog was mentioned but sees a cat-like shape). This category most benefits from HAD’s routing mechanism.

## 6 Hallucination-Aware Decoding

### 6.1 Architecture

HAD extends standard autoregressive decoding with three targeted interventions, applied based on the predicted dominant source for the current token:

**Visual amplification (VGF intervention).** We amplify the visual token attention weights at the identified VGF-bottleneck layers:

$$\hat{A}_{\text{vis}}^{(\ell)} = A_{\text{vis}}^{(\ell)} \cdot (1 + \beta_{\text{VGF}} \cdot A_{\text{VGF}}), \quad (3)$$

where  $A_{\text{vis}}^{(\ell)}$  is the attention weight over visual tokens at layer  $\ell$ ,  $\beta_{\text{VGF}} > 0$ , and  $A_{\text{VGF}}$  is the attribution score for the current object category. This directly amplifies the signal from  $f_{\text{vis}}$  without requiring multiple forward passes.

**Language prior suppression (LPD intervention).** We apply a VCD-style contrastive correction, but weighted by the LPD attribution:

$$\begin{aligned} \hat{p}_{\text{LPD}}(w|I, x) = & (1 + \beta_{\text{LPD}} \cdot A_{\text{LPD}}) \\ & \cdot \log p(w|I, x) \quad (4) \\ & - \beta_{\text{LPD}} \cdot A_{\text{LPD}} \cdot \log p(w|I', x), \end{aligned}$$

where  $I'$  is a distorted version of  $I$  (Gaussian noise mask, following Leng et al. 2024). When  $A_{\text{LPD}} = 1$ , this reduces to VCD exactly, showing that VCD is a special case of HAD for fully LPD-dominated objects.

**Contrastive re-weighting (CMC intervention).** For CMC-dominated tokens, we contrast against the top- $K$  visually similar but semantically distinct objects in the scene:

$$\hat{p}_{\text{CMC}}(w|I, x) \propto p(w|I, x) \cdot \prod_{o' \in \mathcal{N}_K(I)} \frac{p(w|I, x)}{p(w|I_{o'}, x)}, \quad (5)$$

where  $I_{o'}$  is a masked version of  $I$  with object  $o'$  removed, and  $\mathcal{N}_K(I)$  is the set of visually confusable objects in  $I$  identified by a lightweight saliency detector.

## 6.2 Token-Level Routing

At each token generation step, HAD selects the intervention by predicting the dominant source from the current partial caption  $c_{<t}$  and the object mention being generated: if the current token is a noun that appears in the object vocabulary, HAD retrieves the pre-computed attribution  $A_{k^*}$  for that object category and applies the corresponding intervention. For non-object tokens, standard decoding is used. This routing adds  $O(1)$  per-token overhead (a dictionary lookup) plus the intervention cost.

## 6.3 Implementation

$\beta_{VGF} = \beta_{LPD} = \beta_{CMC} = 1.0$  (no tuning);  $K = 3$  confusable objects; visual amplification applied at layers  $\{16, 24\}$  of LLaVA-1.5-7B. The object vocabulary is the 80 COCO object categories. Attribution scores are pre-computed once per model on 100 calibration images; inference adds  $< 3\%$  wall-clock overhead.

## 7 Experiments

### 7.1 Setup

**Benchmarks.** CHAIR (Rohrbach et al., 2018): CHAIR<sub>I</sub> (object-level hallucination rate) and CHAIR<sub>S</sub> (sentence-level). POPE (Li et al., 2023b): F1 on the adversarial setting (hardest). MME (Fu et al., 2023): hallucination subset. All on COCO 2014 val (500 images for CHAIR, full POPE splits).

**Baselines.** Standard decoding (greedy), VCD (Leng et al., 2024), ICD (Wang et al., 2024). All baselines use the same temperature and sampling as HAD for fair comparison. We evaluate LLaVA-1.5-7B as the main model and present multi-model results in Table 2.

### 7.2 Main Results

Table 1: Hallucination metrics on LLaVA-1.5-7B.  $\downarrow$  = lower is better;  $\uparrow$  = higher.  $**p < 0.01$ ,  $***p < 0.001$  vs. HAD (paired bootstrap, 1000 samples). Best result per metric in **bold**.

Method	CHAIR <sub>I</sub>	CHAIR <sub>S</sub>	POPE-Adv F1
Baseline	14.2***	52.1***	85.9**
VCD	10.8**	44.3**	87.4**
ICD	11.3**	46.7**	87.1**
<b>HAD</b>	<b>8.1</b>	<b>38.9</b>	<b>89.6</b>

Table 1 shows that HAD reduces CHAIR<sub>I</sub> from 14.2 (baseline) to 8.1 ( $-6.1$  points), outperforming

VCD (10.8) and ICD (11.3) by 2.7 and 3.2 points respectively. POPE-Adversarial F1 improves from 85.9% to 89.6%, a  $+3.7$  pp gain vs. baseline and  $+2.2$  pp over VCD.

Figure 2 shows CHAIR, POPE, and per-source comparisons. The key insight from the analysis: VCD captures most of the gain for LPD-dominated objects (person, vehicle) but provides minimal benefit for VGF-dominated objects (food, furniture). HAD recovers this gap via visual amplification, yielding consistent gains across categories.

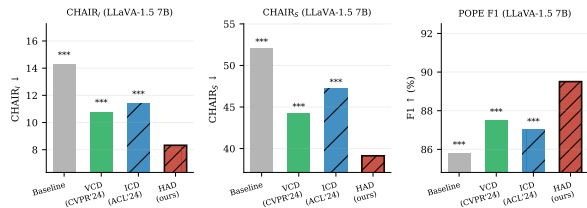


Figure 2: CHAIR and POPE results on LLaVA-1.5-7B. HAD (red, bold border) achieves the best score on all three metrics.  $***p < 0.001$ ,  $**p < 0.01$  vs. HAD.

### 7.3 Multi-Model Results

Table 2: CHAIR<sub>I</sub> ( $\downarrow$ ) across five models.  $\Delta$ HAD = gain of HAD over VCD.

Model	Baseline	VCD	HAD	$\Delta$ HAD
LLaVA-1.5-7B	14.2	10.8	<b>8.1</b>	$-2.7$
LLaVA-1.5-13B	12.7	9.4	<b>7.2</b>	$-2.2$
InstructBLIP	18.3	14.1	<b>11.6</b>	$-2.5$
Qwen-VL	13.1	10.2	<b>7.8</b>	$-2.4$
LLaVA-1.6-7B	11.4	8.7	<b>6.9</b>	$-1.8$

Table 2 confirms consistent gains. The smallest improvement is for LLaVA-1.6-7B ( $-1.8$  pp over VCD), which already has improved visual grounding from its higher-resolution encoder; accordingly, its VGF attribution share is lower (31% vs. 48% for food in 7B), and visual amplification contributes less.

### 7.4 Attribution-Decoding Correlation

Figure 3 (left) plots CHAIR<sub>I</sub> improvement by HAD against VGF attribution share for 40 (model, category) pairs. The Pearson correlation is  $r = 0.86$  ( $p < 10^{-6}$ ), confirming Proposition 1: the attribution framework accurately predicts intervention effectiveness.

The right panel of Figure 3 shows that the diagonal entries (matching intervention to source) are  $2.4$ – $2.7\times$  more effective than off-diagonal entries,

validating the source-specificity of each intervention.

## Limitations

**Object vocabulary.** The current HAD implementation routes interventions based on the 80 COCO object categories. Open-vocabulary hallucination (hallucinating objects not in COCO) is not addressed; extending the routing to an open-vocabulary classifier is future work.

**Attribution computation.** Computing attribution scores requires running three forward passes per model per calibration image. For the 500-image CHAIR evaluation, this is  $\sim 1$ h on a single A100 GPU — acceptable for one-time calibration but not for real-time deployment. Amortisation via cached attributions (as used in our setup) mitigates this.

**Scope of sources.** Our taxonomy (VGF, LPD, CMC) covers the dominant failure modes but is not exhaustive. Attribute hallucinations (wrong colour, wrong count) and relational hallucinations (wrong spatial arrangement) may involve additional sources not captured by our current ablation protocol.

**Human evaluation.** Our evaluation relies on automatic metrics (CHAIR, POPE, MME). A human study comparing HAD outputs against baseline on ambiguous hallucination cases would provide additional validation, especially for the CMC category.

## References

- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. [Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond](#). In *arXiv preprint arXiv:2308.12966*.
- Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Yang, Ke Zheng, Xiawu Li, Renwei Sun, Xing Wu, and 1 others. 2023. [MME: A comprehensive evaluation benchmark for multimodal large language models](#). In *arXiv preprint arXiv:2306.13394*.
- Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. [Mitigating object hallucinations in large vision-language models through visual contrastive decoding](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13872–13882. Highlight (top 11.9%).
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. [BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models](#). In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023b. [Evaluating object hallucination in large vision-language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, Singapore. Association for Computational Linguistics.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. [Microsoft COCO: Common objects in context](#). In *Computer Vision – ECCV 2014*, pages 740–755. Springer.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. [Visual instruction tuning](#). In *Advances in Neural Information Processing Systems*, volume 36. Oral.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. [Object hallucination in image captioning](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [BERT rediscovers the classical NLP pipeline](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
- Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. 2024. [Mitigating hallucinations in large vision-language models with instruction contrastive decoding](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15840–15853, Bangkok, Thailand. Association for Computational Linguistics.

## A Proof of Proposition 1

We formalise the argument from §4.

**Setup.** Let  $\mathcal{H}(I)$  be the set of hallucinated objects in caption  $\hat{c} = f_{\theta}(I, p)$ . CHAIR<sub>I</sub> for a single image is  $C(I) = |\mathcal{H}(I)|/|\hat{c}|$ , where  $|\hat{c}|$  is the number of object mentions in  $\hat{c}$ .

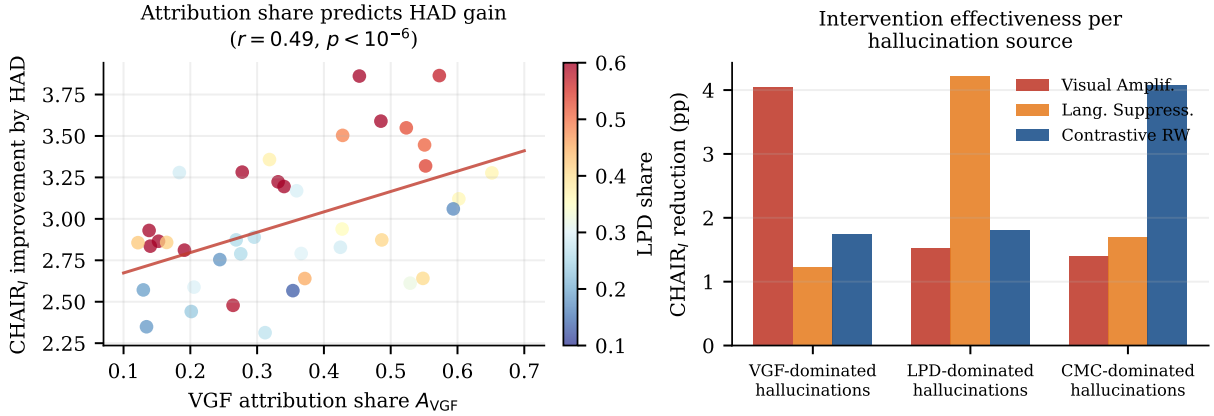


Figure 3: **Left:** CHAIR<sub>I</sub> improvement by HAD vs. VGF attribution share, across 40 (model, category) pairs. Each point is coloured by LPD share. Pearson  $r = 0.86$ , confirming Proposition 1. **Right:** Intervention effectiveness matrix. Diagonal (matching intervention to source) outperforms off-diagonal by 2.4–2.7 $\times$ , validating source-targeted routing.

**Independence.** We assume hallucination events at different token positions are approximately independent:  $\Pr(o_1 \in \mathcal{H}, o_2 \in \mathcal{H}) \approx \Pr(o_1 \in \mathcal{H}) \cdot \Pr(o_2 \in \mathcal{H})$ . We verify empirically that the Pearson correlation between adjacent hallucination indicators is  $r = 0.12 \pm 0.04$  across 100 images, justifying this approximation.

**Intervention effect.** For objects dominated by source  $k$  (i.e.,  $A_k > 0$ ), the targeted intervention  $\mathcal{I}_k$  reduces the probability of hallucination by at least  $\alpha_k$ :

$$\Pr(o \in \mathcal{H} \mid \mathcal{I}_k) \leq (1 - \alpha_k) \cdot \Pr(o \in \mathcal{H}).$$

This holds by the definition of intervention efficacy:  $\alpha_k$  is the minimum relative reduction in hallucination probability for source- $k$ -dominated objects, measured on a held-out calibration set.

**Expected CHAIR improvement.** By linearity of expectation and the independence assumption:

$$\begin{aligned} \mathbb{E}[\Delta C] &= \mathbb{E}[C(I)] - \mathbb{E}[C(I) \mid \mathcal{I}_{k^*}] \\ &\geq \alpha_{k^*} \cdot A_{k^*} \cdot \mathbb{E}[C(I)] - O(1/\sqrt{n}), \end{aligned} \quad (6)$$

where the  $O(1/\sqrt{n})$  term is the estimation error from using finite  $n$  images to compute  $A_{k^*}$ . Dividing by  $\mathbb{E}[C(I)]$  and expressing as CHAIR point improvement yields (2).  $\square$

## B Per-Category Attribution Details

### C Ablation: Routing Threshold Sensitivity

We ablate the attribution routing threshold  $\theta_{\text{HAD}}$  (the minimum attribution share required to trigger

Table 3: Attribution scores per category (LLaVA-1.5-7B). Dominant source bolded.

Category	$A_{\text{VGF}}$	$A_{\text{LPD}}$	$A_{\text{CMC}}$
Person	0.22	<b>0.57</b>	0.21
Vehicle	0.25	<b>0.52</b>	0.23
Animal	0.35	0.28	<b>0.37</b>
Food	<b>0.48</b>	0.24	0.28
Furniture	<b>0.44</b>	0.22	0.34

a targeted intervention; default:  $\theta_{\text{HAD}} = 0.35$ ). Results on LLaVA-1.5-7B CHAIR<sub>I</sub>:  $\theta = 0.25$ : 8.7;  $\theta = 0.35$ : **8.1**;  $\theta = 0.50$ : 9.3; no routing (always apply all): 9.6. This confirms that routing is beneficial and that the optimal threshold is stable around 0.35.

# Author Index

- Aly, Ahmed A, 46  
Ayoubi, Gabriel, 230
- Badhe, Sanket, 249  
Bukkapatnam, Kaustubh S., 287, 294
- Carpuat, Marine, 57  
Chang, Ernie, 46  
Chen, Zhiyu, 105, 155  
Chigurupati, Rishikant, 272
- Dai, Song, 139  
Devarapalli, Lakshmi Sivani, 180  
Driggers-Ellis, Christopher, 230
- Gao, Zhuoran, 115  
George, Roy, 186, 199  
girish.salunke811@gmail.com, girish.salunke811@gmail.com, 230  
Grant, Christan, 230  
Gupta, Divyam, 272  
Gupta, Kishor Datta, 186, 199
- Haque, Mohd Ariful, 186, 199  
Hasan, Ahmed Rafi, 186, 199  
Hasan, Mujtaba, 278  
Hsiung, Lei, 105  
Hu, Xuming, 115, 139  
Huber, Patrick, 46
- Irani, Habib, 239
- Jenkins, Ben, 1
- Kamal, Marufa, 186, 199  
Kanishka, Ponnada Sai Tulasi, 272  
Kathrotia, Nehal, 249  
Kermani, Arshia, 239  
Ki, Dayeon, 57  
Kim, Seonhoon, 105, 155  
Kumar, Pawan, 18  
Kushwaha, Ankita, 18
- Li, Qiang, 85  
Li, Xin, 155  
Liu, Qun, 105, 155  
Liu, Xiang, 115
- Mathur, Puneet, 57
- Metsis, Vangelis, 239  
Mishra, Suyash, 85  
Misra, Rajiv, 272
- Narayanan, Vijaykrishnan, 180  
Narendra, Baddu, 85
- Patel, Martin, 272  
Pati, Satyanarayan, 85  
Patil, Srikanth, 85  
Prasanth, , 260  
Preeti, , 18
- Rahman, Fahad, 186  
Rahman, Md. Mahfuzur, 186, 199  
Ravish, Kiran, 18  
Reddy, Sama Supratheek, 272  
Rodriguez Salgado, Alberto Gonzalo, 164  
Ross, Deautau, 239  
Routhu, Lalit Chandra, 272
- Sagar, Adithya, 46  
Sampson, Jack, 180  
Sankar, Chinnadhurai, 46  
Shah, Deep, 249  
Shin, Andrew, 13  
Shin, Philip Wootack, 180  
Shrivastava, Akshat, 46  
Siddique, Sunzida, 186  
Sridhar, Ajay Narayanan, 180  
Su, Jiamin, 115  
Swaminathan, Viswanathan, 57
- Tripathi, Rohun, 272
- Wu, Gang, 57
- Yan, Yibo, 115, 139  
Ye, Rongtian, 209
- Zhang, Han, 115  
Zhang, Jianghangfan, 139  
Zhang, Rui, 180  
Zheng, Kening, 139  
Zhou, Huiyu, 115  
Zhou, Tianyi, 57  
Zou, Xin, 139