On Efficient Language and Vision Assistants for Visually-Situated Natural Language Understanding: What Matters in Reading and Reasoning

Geewook Kim
NAVER Cloud AI
KAIST AI
gwkim.rsrch@gmail.com

Minjoon Seo
KAIST AI
minjoon@kaist.ac.kr

Abstract

Recent advancements in language and vision assistants have showcased impressive capabilities but suffer from a lack of transparency, limiting broader research and reproducibility. While open-source models handle general image tasks effectively, they face challenges with the high computational demands of complex visually-situated text understanding. Such tasks often require increased token inputs and large vision modules to harness high-resolution information. Striking a balance between model size and data importance remains an open question. This study aims to redefine the design of vision-language models by identifying key components and creating efficient models with constrained inference costs. By strategically formulating datasets, optimizing vision modules, and enhancing supervision techniques, we achieve significant improvements in inference throughput while maintaining high performance. Extensive experiments across models ranging from 160M to 13B parameters offer insights into model optimization. We will fully open-source our codebase, models, and datasets at https://github.com/naver-ai/elva.

1 Introduction

Recent advancements in integrating Large Language Models (LLMs) with computer vision have led to the creation of sophisticated Language and Vision Assistants (Liu et al., 2023b, 2024c; Kim et al., 2023a; Laurençon et al., 2024b,a). These systems are capable of interpreting text within images, enabling them to excel in complex tasks requiring both visual and textual understanding. Notably, models like GPT-4(V) (OpenAI, 2023) demonstrate a range of sophisticated capabilities, including visually-situated Natural Language Understanding (NLU) tasks, positioning them as powerful assistants. However, these models also face significant challenges related to transparency and accessibility, limiting broader utilization.

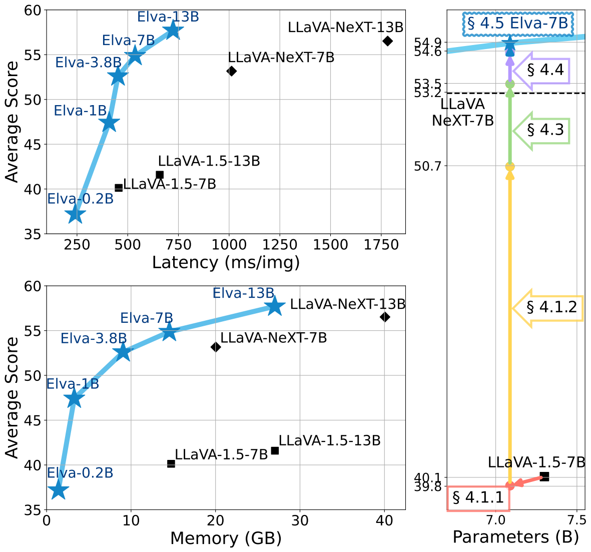


Figure 1: **Graphical comparison illustrating average score against latency and memory consumption for various models.** Scores are derived from eight benchmarks: DocVQA (Mathew et al., 2021), ChartQA (Masry et al., 2022), InfographicVQA (Mathew et al., 2022), SEED-IMG (Li et al., 2024b), SEED-2-Plus (Li et al., 2024a), MMStar (Chen et al., 2024), ScienceQA (Lu et al., 2022), and HallusionBench (Guan et al., 2024). See Section 4.2 for benchmark details. **Elva** excels with high performance, reduced latency, and lower memory usage. Right: Performance improvements from LLaVA to Elva at the 7B scale, achieved through strategies in Section 4.

Open-source alternatives such as LLaVA (Liu et al., 2023b, 2024c) have emerged to address these issues. However, as these models grow in complexity, concerns about their reproducibility and resource efficiency persist. Some open-source models provide only the model weights without comprehensive specifications, making replication and use more challenging.

In the fast-evolving realm of Vision-Language Models (VLMs), simply expanding model size and consuming more resources does not necessarily enhance practical utility. It is crucial to strike a

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 16978–17000
November 12-16, 2024 ©2024 Association for Computational Linguistics

balance between high performance and resource efficiency to democratize access to advanced VLMs. Particularly, inference costs are a significant concern for practitioners developing real-world applications. Despite the importance of this balance, key elements contributing to VLM success are still not fully explored.

Traditionally, to enhance the performance, many VLMs have increased their model resolution, often leading to larger and more resource-intensive models. In this work, we challenge this approach by introducing ELVA (**E**fficient **L**anguage and **V**ision **A**ssistant), a suite of VLMs designed to maintain high performance while reducing inference costs. While we do increase training costs to a manageable extent, the primary research target of ELVA is to create models capable of handling high-resolution tasks with low inference costs.

Our key contributions are as follows:

1. **Efficiency and Reproducibility:** We present ELVA, an efficient and scalable model architecture trained on open-source data, demonstrating superior reproducibility and cost-effectiveness as shown in Figure 1.

2. **Empirical Validation:** We conduct thorough experiments to validate the effectiveness of ELVA's primary components.

3. **Model Scalability:** We develop ELVA versions ranging from 160M to 13B parameters, showcasing its scalability and adaptability.

4. **Dataset Contributions:** To test ELVA as a document assistant, we introduce two new datasets, CORD-Instruct and Parsing-Bench.

5. **Open-Source Initiative:** To foster further community research and ensure model reproducibility, we will fully open-source the trained models and datasets from this study.

Our ultimate goal is to shed light on the complexities of VLMs, helping readers identify the critical factors driving model success while presenting a practical, cost-effective solution for diverse real-world applications. Following this introduction, §2 provides an overview of the foundational LLaVA framework; §3 discusses computational challenges; §4 outlines our proposals; §5 presents our empirical results and analysis; §6 offers further analysis and ablations; and §7, along with §8, surveys related work and concludes the study, respectively.

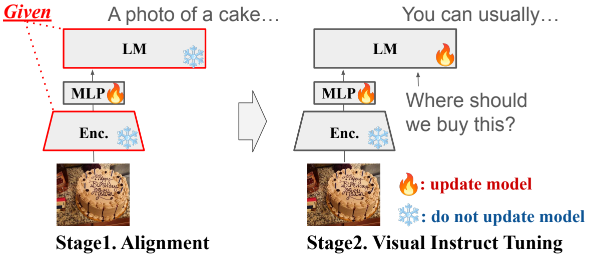


Figure 2: **Training pipeline consists of two stages.** Alignment of visual and textual features through the MLP, followed by joint training of the LM and the MLP.

2 Large Language and Vision Assistants

Architecture. The LLaVA framework (Figure 2) employs a pre-trained Vision Transformer (ViT) (Dosovitskiy et al., 2021) as its vision encoder. Input images are resized and divided into patches of size $n \times (p_h \times p_w \times c)$, where $n = (h/p_h) \times (w/p_w)$, with h and w representing the resized image size, and p_h and p_w denoting the patch size, which are hyperparameters. Here, $c = 3$ denotes the number of channels, typically for RGB images. These patches are processed by the encoder to generate embeddings $\{\mathbf{z}_i \in \mathbb{R}^d\}$, where d is the embedding dimension of the encoder and also a hyperparameter. These embeddings are then mapped to the input space of the language model via a Multi-Layer Perceptron (MLP) before being fed into the model. Optionally, the AnyRes mechanism (Liu et al., 2024c) can be applied, allowing for processing of larger images by first segmenting images into m parts to better capture local features. Simultaneously, the entire image is processed to extract global features. Both global and local features are sequentially fed into the language model, allowing it to utilize comprehensive image information. Therefore, the total token count becomes $(1+m) \times n$. See Liu et al. (2024c) for more details. However, this approach poses a computational challenge due to the increased token count.

Training Objectives and Datasets. The model is trained to minimize Cross-Entropy (CE) loss. During pre-training (alignment phase), it generates captions for images, with CE loss computed on the text. In the instruct tuning stage, given an image, question, and answer, the loss is computed on the answer text. The LLaVA-1.5 dataset (Liu et al., 2024b) is widely used and this study aims to further enhance the dataset. More details are in Section 4.

Model	Token Usage (#tok)	s/img	Memory (GB)
LLaVA-1.5-7B	576	0.46	15
LLaVA-1.5-13B	576	0.66	27
LLaVA-NeXT-7B	approx. 1.7–2.9K	1.01	20
LLaVA-NeXT-13B	approx. 1.7–2.9K	1.78	40
LLaVA-NeXT-34B	approx. 1.7–2.9K	4.00	88

Table 1: **Inference latency and memory costs for LLaVA models.** Tested with NVIDIA V100 GPUs.

3 Efficiency Challenges in LLaVA Models

This section addresses common overhead issues in LLaVA models, identifying critical limitations and defining the problem space for future work (Liu et al., 2024c; Dong et al., 2024).

3.1 Inference Overhead Sources

Inference overhead in LLaVA models stems from several factors:

- **Model Scale:** Larger models (e.g., 34B parameters) offer enhanced capabilities but incur significant computational costs.
- **Vision Encoder Complexity:** Advanced image encoders like SO400M and ViT-G (1.8B) improve performance but increase overhead (Sun et al., 2024; Zhai et al., 2023).
- **Image Resolution:** High resolutions (e.g., 4K) for detailed visual tasks like *DocVQA* (Mathew et al., 2021) increase computational demands on the vision encoder.
- **Vision Token Quantity:** Higher resolutions lead to more vision tokens, increasing the computational load on the LLM (e.g., LLaVA-NeXT uses up to 2880 tokens).

Higher image resolutions and complex tasks further increase the computational demands on both vision encoders and language models.

3.2 Benchmarking Baseline Models

Table 1 shows resource usage during inference for LLaVA and LLaVA-NeXT models, evaluated on the DocVQA and ChartQA test sets (Mathew et al., 2021; Masry et al., 2022). The LLaVA-1.5 models demonstrate manageable computational costs, operable on a single V100 GPU. However, LLaVA-NeXT models, with up to 2.9K tokens, present significant challenges. Testing on an NVIDIA V100 32GB reveals that LLaVA-NeXT-13/34B cannot be accommodated on a single GPU. These findings emphasize the challenges of larger models, especially in resource-constrained environments.

3.3 Existing Approaches to Efficiency

Existing methods offer trade-offs. Sampler modules like the Perceiver resampler (Alayrac et al., 2022) reduce token counts but add architectural complexity and require extra training (Li et al., 2023; Bai et al., 2024). Some studies (Liu et al., 2024b; Dai et al., 2023) have also noted that resamplers may introduce difficulties in generating both lengthy and concise responses effectively, leading to the development of supplementary models to ensure fluent responses (Bai et al., 2024; Laurençon et al., 2024b). Ongoing initiatives, such as the use of convolutional or pooling layers, are also being explored (Cha et al., 2024; Abdin et al., 2024). We believe these concurrent developments could complement our work, demonstrating the potential for integrated use. For a more detailed discussion, please visit Section 7 and Appendix A.2.

Notably, improving models based on the simple architecture for enhanced speed and performance remains a high-impact research area. Due to its inherent simplicity, the LLaVA architecture is already seamlessly integrated with many popular libraries, such as *SGLang* (Zheng et al., 2023b), thereby facilitating broader use and easier deployment in a variety of ongoing real-world applications. With these considerations in mind, in this work, we enhance the LLaVA architecture to address existing limitations, focusing on improving performance and usability without sacrificing simplicity.

4 Efficient Language and Vision Assistant

4.1 Preliminary: Base Architecture Modification and Initial Data Curation

To identify the most effective model architecture, we test various LLMs ranging from 160M to 13B as follows: Llama-160M, Tiny-Vicuna-1B, Phi3-3.8B, Vicuna-7B, and Vicuna-13B[1].

For the vision encoder, we replace the OpenAI CLIP-Large-336-14 module (used in LLaVA-1.5) with OpenAI CLIP-Base-224-32 and utilize the AnyRes technique (Liu et al., 2024c) to optimize the balance between resolution and token count. OpenAI CLIP-Large-336-14 processes a 336x336 area into 576 tokens, while OpenAI CLIP-Base-224-32 processes a 224x224 area into 49 tokens.

---

[1] The links are `https://huggingface.co/Felladrin/Llama-160M-Chat-v1`, `https://huggingface.co/Jiayi-Pan/Tiny-Vicuna-1B`, `https://huggingface.co/microsoft/Phi-3-mini-4k-instruct`, `https://huggingface.co/lmsys/vicuna-7b-v1.5`, and `https://huggingface.co/lmsys/vicuna-13b-v1.5`.

By applying AnyRes, we increase the resolution to 896x676px, with the maximum token count capped at 637, in contrast to LLaVA-NeXT's 2880 tokens for 672x672px. It is important to note that patch size alone does not determine performance; rather, a balanced consideration with resolution ensures optimal results. Through this modification, we achieve a higher input resolution with a slight increase in maximum token count and a marginal decrease in performance (from LLaVA-1.5-7B's 40.1 to 39.8; see §4.1.1 in Figure 1).

Next, we expand the dataset using Idefics2 (Laurençon et al., 2024b), LLaVAR (Zhang et al., 2023), and several open-source datasets to enhance performance. This includes 1.1M images for alignment tasks and 1M for instruction tuning. Further dataset details are in Appendix B.2. As shown in Figure 1, this approach improves performance (39.8 to 50.7) but does not reach LLaVA-NeXT-7B's levels at 53.2 (see §4.1.2 in Figure 1).

4.2 Problem Definition and Strategies

Despite multiple optimizations, the model exhibits performance issues, particularly in generating hallucinations—incorrect responses resulting from inherent bias rather than accurate visual interpretation. See Appendix A.5 for our preliminary analysis on this issue. These problems are especially critical in tasks that require strong integration of visual and textual information.

Hypothesized Challenges. We hypothesize two main challenges: (1) inadequate embeddings from the vision encoder, and (2) a deficiency in the basic comprehension of text within images, which is essential for performing more complex tasks.

Improvement Strategies. To address these challenges, we implement: (1) a more efficient vision encoder to enhance the quality of embeddings, and (2) a training regimen that prioritizes text comprehension before proceeding to more complex tasks.

To test our hypotheses, we conduct a series of comprehensive ablation experiments. Figure 1 illustrates our development stages. We track the effectiveness of our model modifications using various text-centric evaluation benchmarks, including DocVQA (**Doc**) (Mathew et al., 2021), ChartQA (**Chart**) (Masry et al., 2022), InfographicVQA (**Info**) (Mathew et al., 2022), and SEED-2-Plus (**SD2P**) (Li et al., 2024a). Additionally, we employ widely-used general multimodal benchmarks such as SEED-IMG (**SD-I**) (Li et al., 2024b), MMStar (**MMS**) (Chen et al., 2024), ScienceQA-IMG (**SciQA**) (Lu et al., 2022), and HallusionBench (**Hall**) (Guan et al., 2024). Our primary objective is to enhance performance on text-centric tasks while maintaining competitive performance on general tasks and ensuring low inference costs.

Vision Encoder Configuration	**Text-Centric**	**General**	**Overall**
C1. CLIP-B-224-AnyRes (CLIP)	40.3	54.5	47.4
C2. Unfreeze CLIP	34.1	47.6	40.9
C3. *REncoder* (RE)	45.2	52.2	48.7
C4. Avg (CLIP& RE)	45.6	54.4	50.0
C5. ELVA-encoder (Avg (CLIP& 12 REs))	45.7	54.7	**50.2**
Supplementary ablations			
C6. CLIP-L-336 (*LLaVA-1.5 on our data*)	37.5	**58.6**	48.1
C7. CLIP (7%) + *REncoder* (93%)	**45.9**	53.5	49.7

Table 2: **Ablation study results for different vision encoder configurations.** Average scores for text-centric tasks (DocVQA, ChartQA, InfoVQA, and SEED-2-Plus), and general image tasks (SEED, MMStar, ScienceQA, and HallusionBench) are reported. These results are obtained with Phi-3 (3.8B). The overall scores for other scales (from 1B to 13B) are shown in Figure 3.

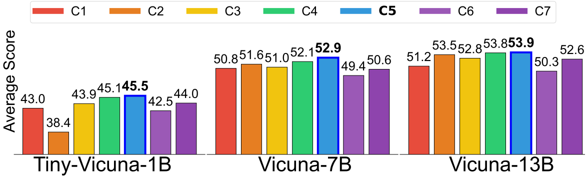


Figure 3: **Performance of various vision encoder configurations at 1B, 7B, and 13B.** Average scores for each configuration (C1 to C7) across 8 benchmarks.

In the followings, we introduce each proposed module in detail and conduct extensive ablation studies, analyzing the impact of removing each component from the final model configuration.

4.3 Developing an Enhanced Vision Encoder with Weight Averaging

To improve visually-situated text comprehension, we develop a new vision encoder optimized specifically for reading text within images. Table 2 presents ablation studies on various vision encoder configurations. Initially, we find that merely unfreezing the vision encoder during VLM training does not lead to notable performance improvements (**C2**). Next, we implement a two-step approach: (1) We first unfreeze the vision encoder and train it on a small scale VLM (1B is used) using text-centric datasets such as OCR-IDL (Biten et al., 2023). This training emphasizes the *Text Reading* task (Kim et al., 2022), where the model reads text embedded within images, allowing the vision encoder to adjust and enhance its text recognition capabilities. (2) Subsequently, we extract the enhanced vision encoder, denoted as *REncoder*, from this text-centric VLM. Note that, the text-centric VLM used to derive the *REncoder* is not utilized in later stages. When training a VLM with the newly obtained *REncoder*, we observe significant improvements in text-centric tasks (**C3**), although its performance on general image tasks diminishes.

Now we have two specialized encoders: the original CLIP for general tasks and the *REncoder*. Drawing inspiration from previous work on *Weight Averaging* (Wortsman et al., 2022), we experiment with averaging the weights of the original encoder and the *REncoder*. Interestingly, this approach yields promising results (**C4**). Furthermore, by slightly adjusting the weight averaging ratios to favor the *REncoder*, we achieve marginally better performance on text-centric tasks (**C7**).

To further enhance robustness, we train 12 *REncoder*s with different random seeds and then average their weights, a practice inspired by Wortsman et al. (2022). This averaging process, taking approximately 1.7 days on 8 V100 GPUs per phase, yields an encoder that substantially improves text comprehension while maintaining general capabilities (**C5**). More training details are in Appendix C.

In summary, the core idea is simple. We **(1) unfreeze the encoder and train a small VLM for text reading tasks, and retrieve the specialized encoder**, and **(2) make it robust to various tasks by applying weight averaging**. Finally, the produced vision encoder is used to build an efficient language and vision assistant, ELVA. Our new **ELVA-Encoder** (**C5**) brings substantial enhancements in text-centric tasks compared to the original base (**C1**). While there is still a reduction in general image performance compared to merely training LLaVA-1.5 on our data (**C6**), understanding the trade-offs in **C6** is key to fully appreciating the balance we achieve. We effectively balance overall performance and computational cost within the scope of CLIP-Base parameters (88M). The ELVA-Encoder configurations demonstrate notable success overall, as shown in Table 2 and Figure 3.

4.4 Augmenting Text Understanding in Images with Read-and-Reason Prompt

Models like LLaVA (Liu et al., 2023b, 2024b) utilize OCR, augmenting user queries with OCR results. However, a comprehensive investigation of methods for supervising textual information during visual instruction tuning remains underexplored.

	1B Text/Gen/All	3.8B Text/Gen/All	7B Text/Gen/All	13B Text/Gen/All
R1.	41.6/48.7/45.2	43.7/53.9/48.8	47.5/56.0/51.8	48.5/56.5/52.5
R2.	**42.4**/48.6/**45.5**	**45.7/54.7/50.2**	**49.2/56.6/52.9**	**50.4**/57.4/**53.9**
Supplementary ablations				
R3.	41.2/46.5/43.9	45.4/53.7/49.6	47.0/55.3/51.2	50.8/56.1/53.5
R4.	42.0/**48.8**/45.4	44.1/54.2/49.1	48.2/56.0/52.1	49.6/**57.6**/53.6

Table 3: **Comparison of different model sizes and RR-Prompt variants. R1** represents standard models trained without additional text reading prompt. **R2** employs explicit initial text reading steps for text-rich taks. **R3** carries out text reading at the end, while **R4** provides OCR results just as context without explicit supervision.

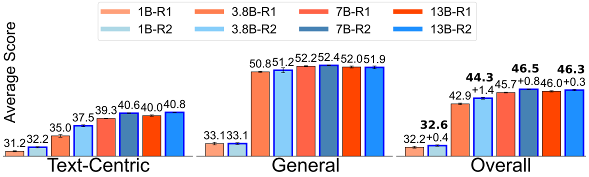


Figure 4: **Impact of RR-Prompt with a 10% dataset subset.** Results demonstrate effects during training.

Table 3 shows ablation studies investigating text reading tasks during visual instruction tuning. Set **R1** follows standard practices using datasets without additional text reading components. In **R2**, inspired by *Prompting* (Brown et al., 2020), we incorporate an initial QA task, *"What is written in this image?"* before QA on text-rich images. For example, with an image of a restaurant menu, the model first **reads all text before querying** about menu items or prices. This incremental addition improves performance significantly from **R1 to R2**, especially in text-rich tasks. We annotate datasets using OCR engines for this purpose.

Our further explorations assess this approach in resource-scarce environments, using 10% of the original instruction tuning dataset size. Figure 4 shows results across 1B to 13B parameter models. We also explore the supervision structure's impact by comparing "Read-and-Reason" versus "Reason-and-Read" approaches. **R3** models perform text reading last to evaluate this. Results confirm that "Read and Reason" is more effective, emphasizing structured prompting's importance in model learning. Lastly, we evaluate the effect of providing read text as context without explicit supervision (**R4**). Explicit supervision with text information yields marginal improvements in text-centric tasks.

In summary, the proposed core idea is to utilize Read-and-Reason Prompt (RR-Prompt) during model training to enhance text understanding in images. This approach, validated through ablation studies, shows significant performance improvements, especially in text-rich tasks. Note that the **RR-Prompt is used during training**; during inference, the model directly engages in reasoning, leveraging the enhanced capabilities acquired through the RR-Prompt, ensuring efficiency without needing an explicit text reading stage.

4.5 Bringing It All Together

To develop a more robust model capable of handling a wider range of tasks, we scale up the model development by incorporating diverse datasets beyond merely text-centric tasks. Our final model integrates four additional datasets: Vision-Flan (Xu et al., 2024), RefCOCO (Kazemzadeh et al., 2014), VG (Krishna et al., 2017), and CORD (Park et al., 2019). By incorporating these additional datasets, we aim to enhance both the performance and generalizability of our model. The final training involved 11K steps with a batch size of 128. The specific dataset details and schedules are in Appendix B. As demonstrated in Figure 1, our final configuration shows solid performance.

5 Experimental Assessment

In this section, we rigorously test and evaluate our ELVA models under varying conditions. We aim to understand their capabilities and limitations by benchmarking them against baseline models across both text-based and image-based tasks.

5.1 Framework

Our evaluation process extends beyond our initial eight datasets, utilized in our ablation studies (See §4.2). To further enrich our examination, we have included additional diverse datasets such as **AI2D** (Kembhavi et al., 2016), MathVista-TestMini (Lu et al., 2024) (**Math**), LLaVA-Bench (Liu et al., 2023b) (**LBen**), along with Parsing-Bench (**PBen**) proposed in this work.

5.2 Generated Scenario-Based Benchmarks

In our research, we identify a significant gap in datasets representing real-world user scenarios for document assistants. To address this, we create the following datasets. These datasets will be open-source, and more details are in Appendix E.

CORD-Instruct. Building on the CORD dataset, which consists of Indonesian receipts and their JSON annotations, CORD-Instruct provides instructional sets for models to generate outputs in JSON, XML, or Markdown formats. We utilize the OpenAI GPT-3.5 API to create these instructional sets, ensuring the exclusion of any erratic samples.

Parsing-Bench. Inspired by the LLaVA-Bench and *LLM-as-a-Judge* (Zheng et al., 2023a), we develop Parsing-Bench to address the limitations of existing benchmarks like LLaVA-Bench, which include limited document-related samples and do not sufficiently reflect real user needs. Figure 6 presents example cases and model predictions. To test the model's ability to extract information from new documents, we create this dataset using 30 images from Brazilian Identity Documents (Álysson Soares et al., 2020) and SROIE (Huang et al., 2019), which are not used during training.

5.3 Results

Table 4 provides a comprehensive comparison of our ELVA models against baselines such as PaliGemma (Beyer et al., 2024), Qwen-VL (Bai et al., 2024), and LLaVA models (Liu et al., 2023b, 2024b,c) across multiple tasks. The results, either reproduced or sourced from original papers, are validated using *VLMEvalKit* (Contributors, 2023) and the official code by Liu et al. (2023b).

The ELVA models consistently exhibit strong performance on both text-centric and general multimodal benchmarks. The ELVA-0.2B model, despite its smaller parameter count, performs commendably across various tasks. Larger models ranging from 1B to 13B demonstrate superior performance, highlighting the advantages of increased model capacity. Notably, ELVA achieves excellent latency and memory efficiency, reinforcing its practicality for diverse real-world applications.

Although the text-centric benchmark performance is strong, there is a slight difference compared to concurrent leading specialized models (Hu et al., 2024), as shown in Table 5. However, our focus is on developing generalist models rather than specialized models. A detailed analysis on this is provided in Section 6.1. Additionally, a notable limitation is observed in the LLaVA-Bench, where ELVA models underperform compared to LLaVA models. As this dataset comprises only 24 images, interpretation requires caution. Further analysis is discussed in Section 6.4.

In summary, ELVA models demonstrate robust performance across a wide range of tasks and

Model	# Param		#tok	s/img	vram	*Text-Centric Benchmarks*					*General Multimodal Benchmarks*						
	Vision	LM				**Doc**	**Chart**	**Info**	**SD2P**	**PBen**	**SD-I**	**MMS**	**SciQA**	**Hall**	**AI2D**	**Math**	**LBen**
LLaVA-v1-13B	300M	13B	576	1.43	26.9	9.8	7.0	19.9	39.5	14.0	51.2	32.9	62.4	43.0	43.9	25.9	69.9
LLaVA-1.5-7B	300M	7B	576	0.46	14.7	22.8	17.8	22.4	41.2	17.9	65.9	33.1	69.2	48.5	55.6	25.6	59.6
LLaVA-1.5-13B	300M	13B	576	0.66	27.0	24.5	18.5	24.9	44.4	19.6	68.2	34.1	72.3	45.7	60.7	27.7	66.1
LLaVA-NeXT-7B	300M	7B	1728-2880	1.01	20.0	68.3	51.9	31.6	51.7	49.6	69.8	38.2	69.0	44.8	66.8	31.8	**72.3**
LLaVA-NeXT-13B	300M	13B	1728-2880	1.78	40.1	69.8	59.0	**34.9**	**55.6**	57.3	**71.5**	41.2	73.4	46.7	**71.7**	34.1	**72.3**
ELVA-0.2B (ours)	88M	0.2B	98-637	0.24	1.4	44.7	50.3	14.8	31.4	12.3	37.8	31.5	39.0	48.1	31.0	27.0	28.4
ELVA-1B (ours)	88M	1B	98-637	0.41	3.3	62.6	57.7	23.7	36.8	27.3	52.3	32.6	63.3	50.4	46.9	31.7	36.0
ELVA-3.8B (ours)	88M	3.8B	98-637	0.45	9.1	66.1	61.9	24.2	44.7	31.0	61.3	36.9	74.2	52.7	63.0	35.6	45.3
ELVA-7B (ours)	88M	7B	98-637	0.54	14.5	69.1	61.8	30.7	47.7	45.0	62.6	35.4	74.7	**56.8**	66.2	36.6	50.7
ELVA-13B (ours)	88M	13B	98-637	0.72	27.0	**71.7**	**65.2**	34.6	52.6	**59.2**	65.3	37.9	77.7	**56.8**	69.3	**38.1**	51.0
Supplementary baselines																	
Qwen-VL-7B	1882M	7B	224	0.50	19.2	65.1	60.2	–	41.0	–	56.5	33.9	60.6	37.4	57.2	15.5	12.9
Qwen-VL-7B-Chat	1882M	7B	224	0.56	19.2	62.6	49.3	–	46.9	–	62.9	34.0	64.0	40.8	59.7	34.9	67.7
PaliGemma-3B	428M	3B	1024	0.98	10.3	–	33.8	–	49.8	–	70.0	**48.6**	**94.3**	53.0	69.3	28.7	36.9

Table 4: **Performance comparison across different models and benchmarks.** This table summarizes model sizes (Vision and LM), token counts (#tok), latency (s/img), and memory cost (vram). The performance metrics across various benchmarks are presented, showcasing each model's strengths and weaknesses in different challenges.

Model	**Doc**	**Chart**	**SD2P**	**SD-I**	**MMS**	**SciQA**	**Hall**	**LBen**
DocOwl1.5-8B	81.6	**70.7**	50.2	50.2	34.7	65.7	28.9	35.3
DocOwl1.5-8B-Chat	**82.2**	69.6	52.4	50.9	34.4	65.0	30.4	39.5
Elva-7B (ours)	69.1	61.8	47.7	62.6	35.4	74.7	**56.8**	50.7
Elva-13B (ours)	71.7	65.2	**52.6**	**65.3**	**37.9**	**77.7**	**56.8**	**51.0**

Table 5: **Comparison with Specialized VLMs.** ELVA shows the balanced scores on diverse benchmarks.

benchmarks. While increased model capacity generally enhances performance, efficiency and latency considerations are essential for practical deployment. Our main results highlight ELVA's efficiency and balanced performance, underscoring the contributions and objectives of our study.

6 Further Analyses and Discussions

6.1 Comparisons with Specialized Models

Given the rapid evolution in this field, evaluating our model against recent advancements is vital to highlight our contributions. We compare our work with mPLUG-DocOwl1.5 (Hu et al., 2024), one of the concurrent state-of-the-art models. Since their results on general multimodal benchmarks are not available, we measure the performances, ensuring accuracy with a sanity check on ChartQA. The results in Table 5 suggest that while mPLUG-DocOwl1.5 excels in text-rich document VQA, it faces challenges in general multimodal tasks. This underscores our focus on developing a generalist model that balances task proficiency and broader efficiency.

Method	**s/img**	**vram**	**Doc**	**Chart**	**Info**	**SD2P**
LLaVA-NeXT-7B	1.01	20.0	68.3	51.9	**31.6**	**51.7**
– w/ max. 1728 tokens	0.70	17.1	51.7	48.0	27.9	44.9
ELVA-7B (ours)	**0.54**	**14.5**	**69.1**	**61.8**	30.7	47.7
LLaVA-NeXT-13B	1.78	40.1	69.8	59.0	**34.9**	**55.6**
– w/ max. 1728 tokens	1.11	30.4	53.9	52.3	30.9	49.2
ELVA-13B (ours)	**0.72**	**27.0**	**71.7**	**65.2**	34.6	52.6

Table 6: **Ablations on reduced vision token counts.** Given time and memory costs, ELVA shows benefits.

6.2 Ablations with LLaVA-NeXT Variants

To test the impact of reducing the number of tokens in LLaVA-NeXT models, we constraine the grid size, resulting in a maximum token count of 1728 (either 336x672 or 672x336 pixels). As shown in Table 6, reducing the vision token count leads to significant performance drops across all evaluated tasks. For example, the performance of the 13B model on DocVQA decreases from 69.8 to 53.9 when the token count is restricted, with similar trends observed in other variants. This analysis highlights the trade-off between token count and model performance: while reducing tokens can enhance computational efficiency, it may lead to a compromise in accuracy.

In contrast, ELVA models demonstrate strong performance along with improved efficiency in both speed and memory usage, underlining their robustness in handling high-resolution text-centric tasks efficiently. The ELVA models effectively balance performance and efficiency, outperforming the LLaVA-NeXT variants with reduced token counts.

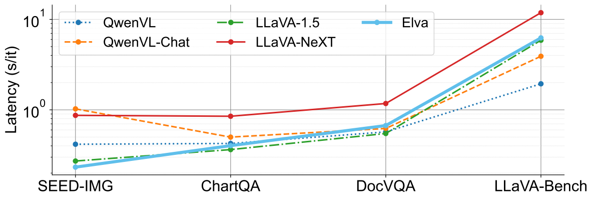


Figure 5: **Latency comparison across multiple benchmarks.** ELVA delivers promising results.

6.3 Discussion on Memory and Time Costs

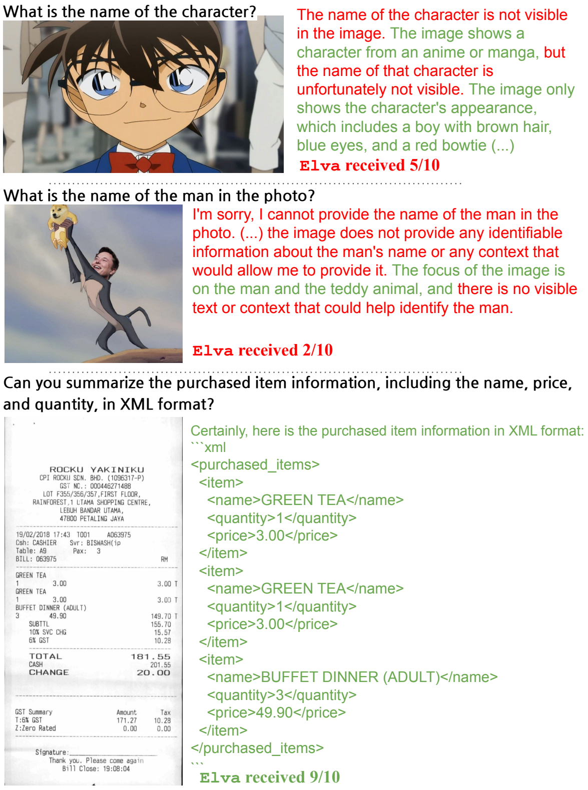
We evaluate latency for ChartQA and DocVQA, as these tasks relate closely to real-world document information extraction scenarios and offer user-centric metrics. Multiple-choice evaluations like SD-I are less indicative of actual user scenarios. Benchmarks requiring longer answers, such as LLaVA-Bench, show inconsistent results due to varied answer lengths across models. Consequently, we focus on ChartQA and DocVQA for latency assessments but also include SD-I and LLaVA-Bench results in Figure 5. These findings indicate ELVA maintains promising latency across varied contexts.

While this study primarily focuses on inference time costs, training costs are equally important for practitioners. Despite handling large datasets, our lightweight vision encoding ensures high training throughput. With training time costs 1.32 to 1.78 times that of LLaVA-NeXT, we find this a fair trade-off for the gains in efficiency and performance. Our approach remains competitive and more resource-efficient compared to several contemporary models. For detailed analysis, please refer to Appendix A.

Regarding memory usage, practical deployment often uses quantization (Dettmers et al., 2023), significantly reducing memory costs. For instance, the LLaVA-NeXT-13B model originally requires two V100 GPUs but can run on a single V100 with quantization, albeit with more latency and some performance loss. Despite these drawbacks, quantization shows promise and will likely improve. Our ELVA models, built for efficiency, complement these advancements, promising even greater value when combined with quantization techniques.

6.4 Case Study on LLM-as-a-Judge Sets

ELVA models perform robustly across benchmarks, excelling in text-centric tasks, but show performance dips on LLaVA-Bench. These issues often arise from entity recognition failures, leading to penalized scores despite the response being logically sound. For example, when asked to name an anime character, ELVA scores 0.5 for failing to provide the correct name despite a detailed description. Similarly, it scores 0.2 in identifying a person in a photo without textual clues. These dips highlight limitations in the ELVA-Encoder's (88M) entity memorization capacity and its reliance on sufficient information present in images. Adding targeted data to memorize entities during instruct tuning might address these issues, though feasibility is debatable. On the other hand, ELVA excels in tasks like summarizing receipt data into XML format, scoring 0.9, demonstrating its ability to accurately read, comprehend, and organize text information. For more discussion and analysis on this, refer to Section 9 and Appendix A.4.



What is the name of the character?

The name of the character is not visible in the image. The image shows a character from an anime or manga, but the name of that character is unfortunately not visible. The image only shows the character's appearance, which includes a boy with brown hair, blue eyes, and a red bowtie (...)

Elva received 5/10

What is the name of the man in the photo?

I'm sorry, I cannot provide the name of the man in the photo. (...) the image does not provide any identifiable information about the man's name or any context that would allow me to provide it. The focus of the image is on the man and the teddy animal, and there is no visible text or context that could help identify the man.

Elva received 2/10

Can you summarize the purchased item information, including the name, price, and quantity, in XML format?

```
ROCKU YAKINIKU
CPI ROCKU SDN. BHD. (1096317-P)
GST NO.: 000446271488
LOT F355/356/357,FIRST FLOOR,
RAINFOREST,1 UTAMA SHOPPING CENTRE,
LEBUH BANDAR UTAMA,
47800 PETALING JAYA
---------------------------------------------
19/02/2018 17:43  T001     A063975
Csh: CASHIER    Svr: BISWASH(ip
Table: A9       Pax:   3
BILL: 063975                              RM
---------------------------------------------
GREEN TEA
1         3.00                          3.00 T
GREEN TEA
1         3.00                          3.00 T
BUFFET DINNER (ADULT)
3        49.90                        149.70 T
    SUBTTL                            155.70
    10% SVC CHG                        15.57
    6% GST                             10.28
    -----------------------------------------
    TOTAL                             181.55
    CASH                              201.55
    CHANGE                             20.00
---------------------------------------------
GST Summary                 Amount       Tax
T:6% GST                    171.27     10.28
Z:Zero Rated                  0.00      0.00
---------------------------------------------
     Signature:_______________________
        Thank you. Please come again
           Bill Close: 19:08:04
```

~~~
Certainly, here is the purchased item information in XML format:
```xml
<purchased_items>
  <item>
    <name>GREEN TEA</name>
    <quantity>1</quantity>
    <price>3.00</price>
  </item>
  <item>
    <name>GREEN TEA</name>
    <quantity>1</quantity>
    <price>3.00</price>
  </item>
  <item>
    <name>BUFFET DINNER (ADULT)</name>
    <quantity>3</quantity>
    <price>49.90</price>
  </item>
</purchased_items>
```
~~~

Elva received 9/10

Figure 6: **Example Results of ELVA on LLaVA-Bench and Parsing-Bench.** The strengths and weaknesses of ELVA are illustrated.

6.5 Discussion on Leveraging OCR

Incorporating OCR can be an effective option for handling text-rich high-resolution images (Kim et al., 2023a). When OCR outputs are incorporated as contextual information during inference, as demonstrated in Table 7, notable enhancements are observed, particularly benefiting ELVA. However, OCR processing has costs. Using the CLOVA

Method	**Doc**	**Chart**	**Info**	**SD2P**
LLaVA-NeXT-7B	74.5 (↑6.2)	53.7 (↑1.8)	35.5 (↑3.9)	55.3 (↑3.6)
ELVA-7B (ours)	**77.8** (↑8.7)	**64.0** (↑2.2)	**39.5** (↑8.8)	**55.7** (↑8.0)
LLaVA-NeXT-13B	76.5 (↑6.7)	62.5 (↑3.5)	40.4 (↑5.5)	58.9 (↑3.3)
ELVA-13B (ours)	**81.1** (↑9.4)	**67.5** (↑2.3)	**44.8** (↑10.2)	**60.6** (↑8.0)

Table 7: **Performance gains with OCR integration.** ELVA excels in both OCR-free and OCR-based modes.

OCR API[2], our tests on the DocVQA dataset average about 4 seconds per sample. Faster OCR engines exist but often at the expense of accuracy. Additionally, upscaling VLMs to handle very high resolutions (e.g., 4K, 8K) may not be practical. Thus, leveraging OCR and similar tools remains a valuable area of exploration, aiming to balance specialized tools and VLMs for optimal performance.

7 Related Work

Visually-Situated Natural Language Understanding (NLU). Visually-situated NLU requires detailed image comprehension and high-resolution processing. Initial VLMs relied on OCR for text extraction. For instance, Xu et al. (2020) utilizes OCR and integrates textual and layout information for document understanding. The field then moved to OCR-free methods (Kim et al., 2022, 2023b; Lee et al., 2023), with models like Donut (Kim et al., 2022) enabling efficient visually-situated NLU.

Multimodal LLMs (MLLMs). MLLMs enhance multimodal comprehension by utilizing LLMs' language understanding. Early models, such as LLaVA (Liu et al., 2023b) and BLIP-2 (Li et al., 2022), align visual representations with frozen LLMs. More recent models, such as LLaVA-1.5 (Liu et al., 2024b), LLaVA-NeXT (Liu et al., 2024c), and Qwen-VL (Bai et al., 2024), unfreeze LLM parameters and use extensive resources (Laurençon et al., 2024b,a; Dong et al., 2024; Hu et al., 2024). Studies like MM1 (McKinzie et al., 2024) provide thorough architecture ablations but focus less on visually-situated NLU, and without offering code or model weights. Laurençon et al. (2024b) conduct multiple ablations and make their methods more accessible to the public, but focus less on reducing resource costs. Small VLMs with under 3B scales (Chu et al., 2024) are emerging. However, there is still a need for compact VLMs for tasks like high-resolution document image processing, emphasizing our work's focus and contribution.

MLLMs for Visually-Situated NLU. Early efforts integrated OCR for text-heavy inputs (Liu et al., 2023b; Li et al., 2022; Liu et al., 2024b), but there is a shift to OCR-free designs (Kim et al., 2023a; Liu et al., 2024c; Laurençon et al., 2024b,a; Dong et al., 2024; Hu et al., 2024). High-performance models use increased input resolutions, raising costs. For instance, LLaVA-NeXT (Liu et al., 2024c) uses 2880 tokens for 672x672 pixels, leading to high costs, furthered by later models (Dong et al., 2024; Wang et al., 2024). Dong et al. (2024) scaled input resolution to 3840×1600 pixels, requiring over 8K tokens.

MLLMs with Vision Token Sampling. To reduce token usage, studies have explored vision token sampling techniques. For example, Qwen-VL (Bai et al., 2024) and Idefics2 (Laurençon et al., 2024b) utilize the Perceiver resampler (Alayrac et al., 2022). These models, however, entail high training costs: Qwen-VL uses over 1.4B data points and Idefics2 over 1B, contrasting with smaller data usage in the LLaVA series (Liu et al., 2023b, 2024b,c). Some studies (Liu et al., 2024b; Dai et al., 2023) have found that models using resamplers can face challenges in generating both lengthy and brief responses, leading to the development of additional models like Qwen-VL-Chat (Bai et al., 2024) and Idefics2-Chatty (Laurençon et al., 2024b). Simpler approaches, such as using convolutional or pooling layers, have also been explored (Cha et al., 2024; Abdin et al., 2024). These methods align orthogonally with the proposed methods in this paper, underscoring the potential for combined use.

Overall, further research is needed to identify key factors for effective VLM design. For additional comparisons, please refer to Appendix A.

8 Conclusion

This study introduces ELVA, a robust and efficient model framework for diverse multimodal tasks, including visually-situated NLU. Empirical results demonstrate that ELVA surpasses existing baselines, with notable memory and latency efficiency. Comprehensive experiments and analyses identify key components driving ELVA's enhanced performance. Additionally, our analysis highlights both strengths and limitations, offering insights for further development. We envision our approach extending to other domains and tasks, particularly those requiring high-resolution and visually-situated NLU, even in resource-constrained environments.

---

[2] `https://clova.ai/ocr/en`

9 Limitations

Despite the significant advancements demonstrated by ELVA, several limitations remain. Firstly, ELVA occasionally struggles to recognize specific entities within images, leading to reduced accuracy in responses, even when they are logically sound. This suggests that the vision encoder may have limitations in recognizing long-tail entities, highlighting the need for further analyses and future research.

Managing very high-resolution images (4K or 8K) is still challenging. While the proposed methods advance the handling of such images, they are not sufficient for easy processing beyond this resolution. We should continue to balance performance improvements with computational resource requirements. For high-resolution document images, incorporating OCR could be a viable option, but it introduces latency and potential accuracy trade-offs, necessitating additional research.

Although ELVA achieves lower inference costs and maintains reasonable training times, processing large data volumes can lead to moderate time differences. As discussed in Appendix A, we have made significant improvements with acceptable increases in training costs, but ongoing optimization in both training efficiency and performance remains necessary.

Future research should focus on enhancing entity recognition, improving training efficiency, and refining OCR integration. Exploring the balance between specialized tools like OCR and an end-to-end VLM is crucial for optimizing performance. Additionally, expanding ELVA's capabilities to handle multilingual or video tasks would further increase its applicability and utility.

10 Ethical Considerations

Developing ELVA involves important ethical responsibilities such as reducing data biases and ensuring transparency. To manage these, we use only controlled and verified open-source datasets for model training. Currently, we rely on the autoregressive models' direct output, but we could also use post-processing techniques or additional training methods to address biases and privacy issues better. By open-sourcing our models and datasets, we encourage peer reviews and collaboration to solve ethical challenges, promoting accountability. These steps help ensure that ELVA upholds high ethical standards and is used for beneficial purposes while minimizing risks.

Acknowledgements

We extend our sincere gratitude to Bado Lee, Daehee Kim, Taeho Kil, and Hodong Lee for their meticulous proofreading of this manuscript. Their input greatly increased its clarity and coherence. We are also immensely grateful to our colleagues in the NAVER Cloud Hyperscale AI Vision Understanding Team and KAIST AI LKLab. Their constant support and encouragement have been a great source of motivation throughout this work. This work was partly supported by KAIST-NAVER Hypercreative AI Center.

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 Santacroce, 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. *Preprint*, arXiv:2404.14219.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2024. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. 2024. PaliGemma: A versatile 3B VLM for transfer. *Preprint*, arXiv:2407.07726.

Ali Furkan Biten, Rubèn Tito, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. 2023. OCR-IDL: OCR Annotations for Industry Document Library Dataset. In *Computer Vision – ECCV 2022 Workshops*, pages 241–252, Cham. Springer Nature Switzerland.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. 2024. Honeybee: Locality-enhanced Projector for Multimodal LLM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. 2023a. Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic. *arXiv preprint arXiv:2306.15195*.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. 2024. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2023b. ShareGPT4V: Improving Large Multi-Modal Models with Better Captions. *arXiv preprint arXiv:2311.12793*.

Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, and Chunhua Shen. 2024. MobileVLM V2: Faster and Stronger Baseline for Vision Language Model. *Preprint*, arXiv:2402.03766.

OpenCompass Contributors. 2023. OpenCompass: A Universal Evaluation Platform for Foundation Models. `https://github.com/open-compass/opencompass`.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Zhe Chen, Xinyue Zhang, Wei Li, Jingwen Li, Wenhai Wang, Kai Chen, Conghui He, Xingcheng Zhang, Jifeng Dai, Yu Qiao, Dahua Lin, and Jiaqi Wang. 2024. InternLM-XComposer2-4KHD: A Pioneering Large Vision-Language Model Handling Resolutions from 336 Pixels to 4K HD. *arXiv preprint arXiv:2404.06512*.

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 *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2024. HallusionBench: An Advanced Diagnostic Suite for Entangled Language Hallucination and Visual Illusion in Large Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14375–14385.

Yu-Chung Hsiao, Fedir Zubach, Maria Wang, and Jindong Chen. 2024. Screenqa: Large-scale question-answer pairs over mobile app screenshots. *Preprint*, arXiv:2209.08199.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. 2024. mPLUG-DocOwl 1.5: Unified Structure Learning for OCR-free Document Understanding. *Preprint*, arXiv:2403.12895.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. 2019. ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520.

Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Han. 2024. Model Stock: All we need is just a few fine-tuned models. In *Computer Vision – ECCV 2024*.

Kushal Kafle, Scott Cohen, Brian Price, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *CVPR*.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar. Association for Computational Linguistics.

Aniruddha Kembhavi, Michael Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. *ArXiv*, abs/1603.07396.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. OCR-Free Document Understanding Transformer. In *Computer Vision – ECCV 2022*, pages 498–517, Cham. Springer Nature Switzerland.

Geewook Kim, Hodong Lee, Daehee Kim, Haeji Jung, Sanghee Park, Yoonsik Kim, Sangdoo Yun, Taeho Kil, Bado Lee, and Seunghyun Park. 2023a. Visually-situated natural language understanding with contrastive reading model and frozen large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11989–12010, Singapore. Association for Computational Linguistics.

Geewook Kim, Shuhei Yokoo, Sukmin Seo, Atsuki Osanai, Yamato Okamoto, and Youngmin Baek. 2023b. On Text Localization in End-to-End OCR-Free Document Understanding Transformer Without Text Localization Supervision. In *Document Analysis and Recognition – ICDAR 2023 Workshops*, pages 215–232, Cham. Springer Nature Switzerland.

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. *Int. J. Comput. Vis.*, 123(1):32–73.

Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. 2024a. Building and better understanding vision-language models: insights and future directions. *Preprint*, arXiv:2408.12637.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. 2024b. What matters when building vision-language models? *Preprint*, arXiv:2405.02246.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin 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*, volume 202 of *Proceedings of Machine Learning Research*, pages 18893–18912. PMLR.

Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. 2024a. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024b. SEED-Bench: Benchmarking Multimodal Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13299–13308.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pretraining with Frozen Image Encoders and Large Language Models. *Preprint*, arXiv:2301.12597.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pretraining for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. 2023a. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*.

Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. 2024a. MMC: Advancing multimodal chart understanding with large-scale instruction tuning. 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 1287–1310, Mexico City, Mexico. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024b. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024c. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual Instruction Tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc.

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)*.

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 *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

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, Dublin, Ireland. Association for Computational Linguistics.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C.V. Jawahar. 2022. InfographicVQA. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1697–1706.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. DocVQA: A Dataset for VQA on Document Images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier Biard, Sam Dodge, Philipp Dufter, Bowen Zhang, Dhruti Shah, Xianzhi Du, Futang Peng, Haotian Zhang, Floris Weers, Anton Belyi, Karanjeet Singh, Doug Kang, Ankur Jain, Hongyu He, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. 2024. MM1: Methods, Analysis & Insights from Multimodal LLM Pre-training.

Sachin Mehta, Mohammad Hossein Sekhavat, Qingqing Cao, Maxwell Horton, Yanzi Jin, Chenfan Sun, Iman Mirzadeh, Mahyar Najibi, Dmitry Belenko, Peter Zatloukal, and Mohammad Rastegari. 2024. OpenELM: An Efficient Language Model Family with Open Training and Inference Framework. *Preprint*, arXiv:2404.14619.

Nostalgebraist. 2020. Interpreting GPT: The logit lens. `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`.

OpenAI. 2023. GPT-4 Technical Report. *Preprint*, arXiv:2303.08774.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: A consolidated receipt dataset for post-ocr parsing. In *Document Intelligence Workshop at Neural Information Processing Systems*.

Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. 2024. EVA-CLIP-18B: Scaling CLIP to 18 Billion Parameters. *Preprint*, arXiv:2402.04252.

Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. VisualMRC: Machine Reading Comprehension on Document Images. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13878–13888. AAAI Press.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *Preprint*, arXiv:2409.12191.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1192–1200. ACM.

Zhiyang Xu, Chao Feng, Rulin Shao, Trevor Ashby, Ying Shen, Di Jin, Yu Cheng, Qifan Wang, and Lifu Huang. 2024. Vision-flan: Scaling human-labeled tasks in visual instruction tuning. In *Findings of the*

Association for Computational Linguistics ACL 2024, pages 15271–15342, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid Loss for Language Image Pre-Training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986.

Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. 2023. LLaVAR: Enhanced Visual Instruction Tuning for Text-Rich Image Understanding. *Preprint*, arXiv:2306.17107.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023a. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Jeff Huang, Chuyue Sun, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2023b. Efficiently programming large language models using sglang. *Preprint*, arXiv:2312.07104.

Álysson Soares, Ricardo das Neves Junior, and Byron Bezerra. 2020. BID Dataset: a challenge dataset for document processing tasks. In *Anais Estendidos do XXXIII Conference on Graphics, Patterns and Images*, pages 143–146, Porto Alegre, RS, Brasil. SBC.

A Additional Analysis and Comparison

A.1 Model Training Cost and Data Efficiency

Table 8 provides a detailed analysis of the training costs associated with different model sizes of ELVA, using 8 NVIDIA A100 80GB GPUs. According to estimates from the official LLaVA-NeXT blog[3], our models take approximately 1.32 to 1.78 times longer to train. The blog reports that it takes 20 hours to train a 7B model with 8 A100 GPUs and 24 hours for a 13B model using 16 A100 GPUs. While training times can vary depending on the testing environment, our data shows that ELVA's training duration results in a moderate increase that remains within reasonable expectations.

Base Model	Alignment Time	Instruct Tuning Time	Total Time
Llama-160M	0.5 hours	3.5 hours	4 hours
Tiny-Vicuna-1B	1.5 hours	6 hours	7.5 hours
Phi3-3.8B	4.5 hours	16.5 hours	21 hours
Vicuna-7B	6.5 hours	29 hours	35.5 hours
Vicuna-13B	11 hours	52.5 hours	63.5 hours

Table 8: **Training times for various model sizes on 8 A100 GPUs.**

This observation becomes clearer when ELVA is compared to other contemporary, data-intensive models. For instance, Qwen-VL (Bai et al., 2024) requires 1.4B data points for pretraining and 50M data points for instruction tuning, whereas ELVA demonstrates a more moderate yet effective use of resources. Similarly, models such as Shikra (Chen et al., 2023a), Idefics2 (Laurençon et al., 2024b), and InternLM-XComposer2-4KHD (Dong et al., 2024) illustrate varying scales of resource utilization, with Shikra using 600K data points for alignment and 5.5M for instruction tuning, Idefics2 achieving results with over 1B data points, and InternLM-XComposer2-4KHD demonstrating scalability with a massive dataset and more than 8K input tokens.

Our observations in Section 4.3 further reinforce the argument for ELVA's efficiency. Despite the extended data, LLaVA-1.5 failed to surpass the overall score we achieved with ELVA (**C5** vs. **C6**), corroborating the efficiency and effectiveness of ELVA. Furthermore, it is crucial to highlight the importance of inference cost. Models aiming for reduced inference costs often face expensive training costs and challenges in maintaining instruction-following capabilities across varied response lengths (Dai et al., 2023; Liu et al., 2024b; Laurençon et al., 2024b). Thus, ELVA emerges as a quick, lightweight, and cost-effective alternative within LLaVA-like simple architectures.

A.2 Ablations with AnyRes and Resampler

As discussed in Section 7, previous research (Liu et al., 2024b; Dai et al., 2023) highlights several limitations associated with resampler-based techniques. To fully understand the limitations, it is essential to empirically investigate them. This section presents our additional experiments on the effectiveness of the Perceiver Resampler (Alayrac et al., 2022), a tool commonly used in many MLLMs to reduce vision token counts (Cha et al., 2024). We conduct these experiments using the Vicuna-7B model.

For this experiment, we begin by training models using CLIP-Large-336-14, as employed in LLaVA-1.5 (Liu et al., 2024b). We then introduce AnyRes (Liu et al., 2024c), which can be interpreted as training the LLaVA-NeXT architec-

---

[3] `https://llava-vl.github.io/blog/2024-01-30-llava-next`

Configuration	Chart	SD2P	SD-I	MMS	SciQA	Hall	AI2D	Math
ELVA-88M	61.8	47.7	62.6	35.4	74.7	56.8	66.2	36.6
CLIP-L-300M	43.9	47.3	66.8	38.2	80.3	55.6	68.9	35.6
+ AnyRes	61.6	53.8	67.9	39.8	77.3	55.8	66.8	37.8
+ Resampler	18.7	38.2	43.1	30.2	72.4	50.9	62.8	28.5

Table 9: **Performance evaluation with AnyRes and Perceiver resampler.** This table illustrates the performance of our proposed ELVA-Encoder (88M) and OpenAI CLIP-Large-336-14 (300M) configurations with AnyRes and Resampler optimizations across diverse tasks. All configurations are trained using the same alignment and visual instruction tuning schedule to guarantee consistent evaluation conditions. For the ELVA setting, the vision token ranges are 98-637, 576 for CLIP-Large; for AnyRes, it is expanded to 1728-2880, and with Resampler, it ranges between 432-720 tokens.

ture on our same dataset and training parameters. Finally, we apply the Perceiver Resampler in an attempt to reduce the token count.

Our findings are summarized in Table 9. The results suggest that performance is notably constrained. This limitation likely arises from the disparity in resources and data used during our compact ELVA training, which may not be sufficient for the resampler to fully realize its potential. Increasing the dataset size and training steps might enhance the effectiveness of the resampler. Additionally, as discussed in Section 7, several improved resampling methods are emerging, and combining them with our approach would likely yield better results.

A.3 Additional Model Variants

In this paper, we employ different LLM families across each scale. We adopt Vicuna as our base model to enable fair comparisons with the LLaVA family (Liu et al., 2023b, 2024b,c). While more advanced LLMs like LLaMA-3[4] could potentially achieve better scores, reaching state-of-the-art benchmarks is not our primary goal. Therefore, we stick with the Vicuna family for consistency. For other scales, we select LLMs that have garnered attention in recent open-source VLM projects. Tiny-Vicuna and LLaMA-Chat are chosen because they are fully open-source models, developed transparently by academic practitioners with limited resources.

We acknowledge that using LLMs trained on the same corpus could provide additional insights into scaling effects. If all these models are trained using different datasets or regimes, it may complicate the evaluation setup, making it difficult to precisely identify performance-related issues. To address this, we also train more variants using the OpenELM family (Mehta et al., 2024), which has been recently released as open-source models. OpenELM offers transparent training details, which help elucidate scaling effects more clearly. Table 10 presents the results, demonstrating that our proposed ELVA model scales effectively and consistently across various LLM architectures, thereby validating the robustness of our approach.

Base LLM	Chart	SD2P	SD-I	MMS	SciQA	Hall	AI2D	Math
LLaMA-160M	50.3	31.4	37.8	31.5	39.0	48.1	31.0	27.0
OpenELM-270M	54.4	32.4	45.0	30.9	46.2	46.9	34.8	29.5
OpenELM-450M	56.8	35.4	50.4	31.8	62.3	51.5	44.0	29.1
Tiny-Vicuna-1.1B	57.7	36.9	52.3	32.6	63.3	50.4	46.9	31.7
OpenELM-1.1B	59.3	40.1	57.1	31.7	67.8	50.3	54.4	33.7

Table 10: **Performance results across different LLM variants.** The results demonstrate the scalability and consistency of our proposed ELVA model across different architectures.

Configuration	Size	Text-Centric	General	Overall	LLaVA-Bench
C1. CLIP-B-Anyres	88M	44.0	58.5	51.2	51.1
C5. Elva-Encoder	88M	50.4	57.4	**53.9**	47.3
C6. CLIP-Large	300M	39.6	60.9	50.3	69.1

Table 11: **Performance comparison across different vision encoder configurations.** The table shows the text-centric, general, and overall benchmark scores, as well as the LLaVA-Bench scores.

A.4 Further Analysis on LLaVA-Bench

In Section 6.4, we examine the lower performance of the ELVA model on the LLaVA-Bench. Given the small number of model parameters in the ELVA-Encoder, we hypothesize that its ability to memorize entities might be limited, potentially contributing to its lower performance. While ELVA often provides logically sound responses to user queries, it sometimes fails to recall specific entity names—a situation comparable to humans struggling to remember the name of an unfamiliar animated character without any contextual clues.

To investigate this further, we consider whether increasing model scale can enhance memorization capacity. We revisit our ablation models from Section 4.3, focusing on the 13B models shown in Figure 3. Table 11 presents additional evaluations on the LLaVA-Bench using these models.

The results indicate that CLIP-Large performs exceptionally well on the LLaVA-Bench. However, both smaller encoder settings, **C1** and **C5**, face

---

[4] `https://ai.meta.com/blog/meta-llama-3`

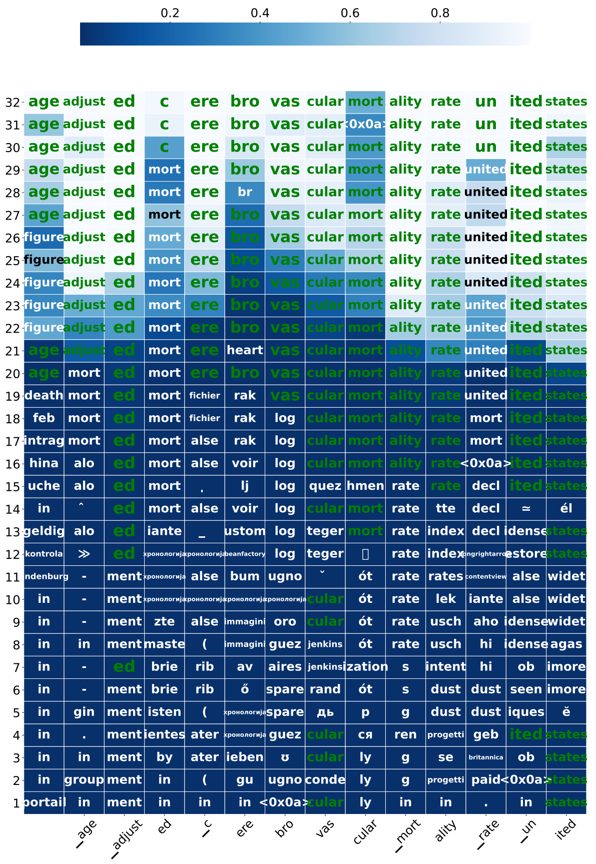


Figure 7: **Results from the ELVA model.** The model accurately predicts the correct answer, with the correct token emerging early in the processing layers, highlighting effective vision and text integration.

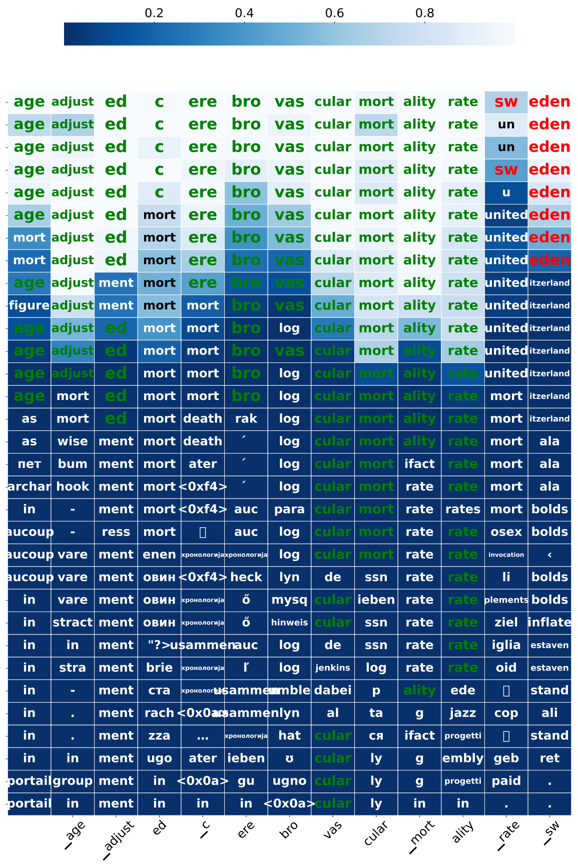


Figure 8: **Results from the ablated model.** This model incorrectly predicts *"sweden"* as the answer, demonstrating the challenges faced without the ELVA-Encoder and RR-Prompt enhancements.

challenges on this benchmark. Despite these challenges, we find the Elva-Encoder to be effective in many other scenarios.

Through this analysis, we recognize the benefits of larger encoders. However, whether increasing parameters significantly enhances memorization is still an open question for future research. We also question whether VLMs should even prioritize memorizing all entities. Our work lays a solid foundation for exploring the trade-offs between model scalability and performance.

A.5 Preliminary Analysis on Hallucinations

As demonstrated in Section 4.1, we explore increasing input resolution without significantly raising inference costs by integrating AnyRes into the LLaVA-1.5 model, thus avoiding excessive growth in vision token counts. Additionally, we enhance performance by expanding the training dataset. Despite these improvements, as discussed in Section 4.2, the model still faces performance issues, particularly in generating hallucinations—incorrect responses due to inherent bias rather than accurate visual interpretation. This section delves into

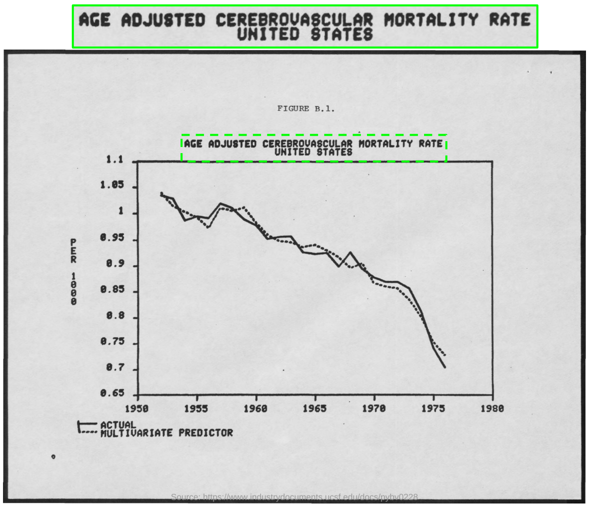


Figure 9: **DocVQA sample.** The question posed is, *"What is the title of the plot?"* The model received instructions to respond concisely in lowercase. For the query, *"age adjusted cerebrovascular mortality rate united states"* is the expected answer.

our preliminary analysis, presenting and explaining samples that illustrate these issues.

Without applying the proposed modules, the ablation model sometimes produces unexpectedly incorrect responses in tasks that require interpreting text within images. However, it's important to note that this model, which excludes the Elva-Encoder and RR-Prompt, is not inherently inadequate. In fact, as shown in Figure 1, its overall performance significantly surpasses that of LLaVA-1.5 (improving from 40.1 to 50.7). This underscores the significant impact of our methods. Figures 7, 8, and 9 illustrate results from our Elva model at the 7B scale, an ablation model without the Elva-Encoder and RR-Prompt, and a DocVQA sample used for analysis. To better understand the model's internal processes, we employ the *Logit Lens* technique from Nostalgebraist (2020) to visualize the behavior across all model layers in this study.

As seen in Figure 7, the Elva model accurately identifies the correct answer, with the correct token emerging as the top candidate relatively early in the processing layers. For simplicity, this analysis did not differentiate between uppercase and lowercase letters. On the other hand, Figure 8 presents an intriguing result where the model becomes confused among various country names and incorrectly outputs *"sweden"* as the answer. Notably, Figure 9 shows that there is no indication in the image that resembles "sweden." This suggests that Vicuna-7B's inherent language modeling capabilities possibly override image reference interpretation.

Through this analysis, as explained in Section 4.2, we hypothesize two main challenges: (1) inadequate embeddings from the vision encoder and (2) a poor grasp of basic text comprehension tasks, crucial for complex document interpretation. These insights guide our strategic approach in developing Elva, addressing these challenges step by step.

B Experimental Details

B.1 Software and Hardware Setup

Our experiments are based on the official codebase[5] of LLaVA (Liu et al., 2023b). We utilize NVIDIA V100 32GB and A100 80GB GPUs for the computations. Ablation studies are conducted on V100 GPUs, whereas the final configuration models run on A100 GPUs. We do not observe any significant performance difference based on the type of GPU used. However, training on V100 GPUs is approximately 2 to 3 times slower per step compared to A100 GPUs. Although our codebase is based on LLaVA, to ensure better reproducibility, we will release the scripts used for training our models and any necessary code modifications as open-source.

Dataset	# Samples
LLaVA	157,712
SG40k	40,688
VQA-v2	82,783
GQA	72,140
OKVQA	8,998
OCRVQA	80,000
A-OKVQA	66,160
TextCaps	21,953
RefCOCO	48,447
VG	86,417

Table 12: **Curated dataset from LLaVA-1.5.** Dataset proportions are shown. RefCOCO and VG are not used in the ablation studies.

B.2 Datasets and Hyperparameters

Curated Dataset from LLaVA-1.5. Table 12 provides detailed quantities of the subsets within the dataset[6].

Curated Dataset from Elva. Table 13 lists the datasets in Elva's final configuration. Meanwhile, for the alignment phase, we use the alignment datasets from LLaVA (Liu et al., 2023b) and LLaVAR (Zhang et al., 2023), which consist of 558K and 422K samples respectively.

Hyperparameters. Table 14 and Table 15 provide the hyperparameters used during the alignment and instruction tuning stages, noting that smaller models benefit from larger learning rates. In the final model training configuration, we employ the data and sampling ratios outlined in Table 13 and train the model for 11K steps. Calculating the exact number of unique images is complex due to overlap across datasets; however, we estimate using approximately 1M unique images. LLaVA-NeXT reported using 760K samples[7], and our use represents a modest increase. Furthermore, as we leverage multiple curated datasets with slightly different questions on the same images, we consider a synthetic epoch to consist of 1.4M examples. Thus, with a batch size of 128, we complete 11K steps (1.4M / 128).

---

[5] `https://github.com/haotian-liu/LLaVA`

[6] `https://huggingface.co/datasets/liuhaotian/LLaVA-Instruct-150K/blob/main/llava_v1_5_mix665k.json`

[7] `https://llava-vl.github.io/blog/2024-01-30-llava-next`

Dataset	# Samples	Sampling Ratio %
LLaVA-1.5-Set (See Table 12)	665,298	46.49
Vision-Flan-Set (Xu et al., 2024)	186,103	12.99
WikiArt (Chen et al., 2023b)	500	0.03
Celebrity (Chen et al., 2023b)	498	0.03
Landmark (Chen et al., 2023b)	500	0.03
Share-TextVQA (Chen et al., 2023b)	500	0.03
DocVQA (Mathew et al., 2021)	11,480	1.60
ChartQA (Masry et al., 2022)	18,317	2.56
*Cauldron-Set-AI2D	2,434	0.17
*Cauldron-Set-Chart2Text	26,961	1.88
*Cauldron-Set-Diagram-Image-to-Text	300	0.02
*Cauldron-Set-HITAB	2,500	0.17
*Cauldron-Set-IAM	5,663	0.40
*Cauldron-Set-RenderedText	10,000	0.70
*Cauldron-Set-Robut-SQA	8,514	0.59
*Cauldron-Set-Robut-WTQ	38,246	2.67
*Cauldron-Set-ScienceQA	4,976	0.35
*Cauldron-Set-Screen2words	15,730	1.10
*Cauldron-Set-STVQA	17,247	1.20
*Cauldron-Set-TabMWP	22,722	1.59
*Cauldron-Set-InfoVQA	2,118	0.15
*Cauldron-Set-TQA	1,493	0.10
CORD-Instruct (Proposed in this work, §5.2)	680	0.05
VisualMRC (Tanaka et al., 2021)	7,959	0.56
LLaVAR-Inst (Zhang et al., 2023)	19,732	1.38
DocReason (Hu et al., 2024)	25,877	1.81
DocVQA-single†	44,815	9.38
ChartQA-single†	28,068	5.88
Layout-en-sampled‡ (Kim et al., 2023a)	50,000	3.49
DVQA-sampled‡ (Kafle et al., 2018)	10,000	0.70
MMC-Chart-sampled‡ (Liu et al., 2024a)	10,000	0.70
ScreenQA-sampled‡ (Hsiao et al., 2024)	10,000	0.70
LRV-Chart-sampled‡ (Liu et al., 2023a)	6,746	0.47

Table 13: **Overview of datasets used in the final data configuration.** All datasets are open-source and freely accessible. Datasets marked with * are subsets curated by Laurençon et al. (2024b), with only selected portions adopted in this work. † indicates that each question-answer pair is considered as a single sample. Datasets marked with ‡ had a large volume of data, hence, only partial images were randomly sampled.

Model Size	LR	Epsilon	Grad Clip Norm	Weight Decay	Warmup Ratio
0.2B	1e-3	1e-6	0.5	0.0	0.03
1B	1e-3	1e-6	0.5	0.0	0.03
3.8B	1e-3	1e-6	0.5	0.0	0.03
7B	1e-4	1e-6	0.5	0.0	0.03
13B	1e-4	1e-6	0.5	0.0	0.03

Table 14: **Hyperparameters used during the alignment stage.**

Model Size	LR	Epsilon	Grad Clip Norm	Weight Decay	Warmup Ratio
0.2B	3e-4	1e-6	0.5	1e-3	0.03
1B	2e-4	1e-6	0.5	1e-3	0.03
3.8B	2e-4	1e-6	1.0	0.0	0.03
7B	2e-5	1e-6	1.0	0.0	0.03
13B	2e-5	1e-6	1.0	0.0	0.03

Table 15: **Hyperparameters used during the instruct tuning stage.** Larger learning rates were noted to be more effective for smaller models.

For ablation studies, we exclude datasets like VG, RefCOCO, and Vision-Flan to minimize costs, resulting in 9K training steps. Additionally, we find that the 0.2B model converges more slowly, so we extend its instruct tuning to twice the number of steps compared to the other models (1B to 13B). This increased number of steps is applied solely to the 0.2B scale.

B.3 Evaluation Details

As described in Section 4.2, which outlines our strategic model development approach, we use eight benchmarks: DocVQA (**Doc**) (Mathew et al., 2021), ChartQA (**Chart**) (Masry et al., 2022), InfographicVQA (**Info**) (Mathew et al., 2022), SEED-2-Plus (**SD2P**) (Li et al., 2024a), SEED-IMG (**SD-I**) (Li et al., 2024b), MMStar (**MMS**) (Chen et al., 2024), ScienceQA-IMG (**SciQA**) (Lu et al., 2022), and HallusionBench (**Hall**) (Guan et al., 2024). Additionally, for the main experiments and analyses, we include several more benchmarks: **AI2D** (Kembhavi et al., 2016), MathVista-TestMini (**Math**) (Lu et al., 2024), LLaVA-Bench (**LBen**) (Liu et al., 2023b), and the Parsing-Bench (**PBen**) proposed in this work. We conduct evaluations using *VLMEvalKit* (Contributors, 2023) and the official code by Liu et al. (2023b). When evaluating LLaVA-Bench, we transition to using `gpt-4-0613` for judging, as the previously widely-used `gpt-4-0314` is deprecated.

DocVQA and InfographicVQA employ active leaderboards[8], which require JSON-formatted submissions for performance verification. This procedure, while thorough, can hinder rapid evaluations needed for iterative experimentation. To mitigate this, our ablation studies detailed in Section 4 utilize custom evaluation scripts for **Doc** and **Info**. This involves parsing ground truth data: for DocVQA, we extract information from CSV files provided by the leaderboard, while we use the validation set for InfographicVQA. Our custom evaluation scores show high correlation with the official leaderboard results, confirming their credibility. That is, for ablation studies, we adopt a simplified evaluation to effectively compare different architectures within the ELVA framework.

However, for the main results and analyses (e.g., Table 4, 5, 6, and 7), which require comparisons with other models, we use test set performance metrics from the official leaderboard to ensure accurate, apple-to-apple comparisons.

C Details on ELVA-Encoder Training

C.1 Dataset and Hyperparameters

Our primary training focus is on text reading tasks, aimed at enhancing the vision encoder's text recog-

---

[8] `https://rrc.cvc.uab.es`

Batch Size	Learning Rate (LR)	Weight Decay
128	5e-5	1e-3
128	6e-5	1e-3
128	7e-5	1e-3
128	8e-5	1e-3
256	5e-5	1e-3
256	6e-5	1e-3
256	7e-5	1e-3
256	8e-5	1e-3
512	5e-5	0.0
512	6e-5	0.0
512	7e-5	0.0
512	8e-5	0.0

Table 16: **Configurations for 12 *REncoder* trainings.** The table shows batch size, learning rate, and weight decay. All other hyperparameters are essentially identical to those in §B.2.

nition capabilities. The training datasets include OCR-IDL (Biten et al., 2023) (837,922 samples), PDFA[9] (1,048,569 samples), as well as the alignment sets from LLaVA (Liu et al., 2023b) and LLaVAR (Zhang et al., 2023). We employ duplicate sampling, treating 3.5M samples as a synthetic epoch, completing one epoch as detailed in Table 16. This process is repeated to produce 12 distinct *REncoder* variants. With the selected 1B model scale, each training session requires approximately 1.7 days using 8 V100 GPUs.

C.2 Details on Small VLM Usage

As outlined in Section 4.3, each *REncoder* variant undergoes training by unfreezing the vision encoder and fine-tuning it within a 1-billion-parameter model setting focused on text-centric datasets. We selected the 1B scale to balance computational demands and model performance. Preliminary experiments, detailed in Table 17, suggest that while a 0.2B ELVA-Encoder configuration offers some benefits, the 1B variant yields noticeably better results. We attribute this improvement to the 1B model's superior text reading capabilities, enhancing learning outcomes during the ELVA-Encoder process. Although coupling each REncoder with a larger model could further improve text recognition, it comes at the cost of increased computational resources and training time. For instance, a 3.8B model requires about 3.5 times longer to train than a 1B model, making it less feasible for many practitioners. Thus, the 1B parameter model was chosen to achieve significant enhancements while maintaining computational efficiency.

Target Size	No ELVA-Encoder*	With 0.2B Trained ELVA-Encoder	With 1B Trained† ELVA-Encoder
0.2B	32.1	**35.0**	34.9
1B	43.0	44.4	**45.5**
3.8B	47.4	49.4	**50.2**

Table 17: **Effect of ELVA-Encoder integration.** This table illustrates the performance impact across various model sizes with and without ELVA-Encoder integration, as well as the influence of VLM size used in ELVA-Encoder training. Columns marked with $*$ correspond to results for C1 in Section 4.3, while those marked with † correspond to C5 in the same section.

Ratio	Text-Centric	General	Overall	Desc.
0%	41.4	46.5	43.9	*REncoder* (Denoted as **C3** in Sec 4.3)
5%	41.8	45.7	43.7	
7%	41.5	46.6	44.0	**C7** in Sec 4.3
10%	41.3	46.5	43.9	
25%	41.2	46.5	43.9	
50%	41.2	49.1	45.1	Avg (CLIP&RE) (**C4** in Sec 4.3)
100%	37.4	48.6	43.0	CLIP-B-224-AnyRes (**C1** in Sec 4.3)

Table 18: **Performance outcomes with varying weight averaging ratios.** This table illustrates the text-centric, general, and overall scores derived from different integration ratios of OpenAI CLIP weights and *REncoder* contributions. Notable configurations such as **C1**, **C3**, **C7**, and **C4** are reported as detailed in Section 4.3.

C.3 Details on Model Weight Averaging

In this study, we propose an approach to develop an efficient and effective vision encoder by keeping the encoder's parameters learnable within a small VLM training framework. The resulting specialized vision encoder weights are merged with the original CLIP weights, known for general image understanding prowess. The idea of simply averaging multiple model weights has been shown to be effective in various studies (Wortsman et al., 2022; Jang et al., 2024). Our approach closely relates to the *Uniform Soup* method used as a baseline by Wortsman et al. (2022), where uniform merging is applied. Recently, more advanced weight merging techniques have been explored (Wortsman et al., 2022; Jang et al., 2024). Given our framework's orthogonal nature to these methods, we expect our proposed practice to be complementary and potentially used in conjunction with these novel techniques.

In addition, we present supplementary experimental results conducted to design the experiment in Section 4.3. These experiments are carried out using multiple 1B scale VLMs. First, we examine how different mixing ratios of OpenAI CLIP weights influence performance. It is observed that a balanced improvement in performance occurs

---

[9] `https://huggingface.co/datasets/pixparse/pdfa-eng-wds`

	No merge	Avg (CLIP&RE)	w/ 2REs	w/ 4REs	w/ 8REs	w/ 12REs
Text	**37.4**	**41.2**	**41.7**	**41.9**	**42.0**	**42.4**
General	48.6	49.1	47.6	45.1	46.3	48.6
Overall	43.0	45.1	44.7	43.5	44.1	45.5

Table 19: **Effect of increasing the number of *REncoders* on performance.** This table presents the text-centric, general, and overall scores resulting from utilizing different numbers of *REncoder* integrations, detailing the observed trends in performance improvement across configurations.

across various ratios (see Table 18). Subsequently, we increase the number of *REncoder* instances used in merging, with results presented in Table 19. As shown, performance generally improves as more *REncoder* instances are included, with our decision ultimately favoring the use of 12 encoders as a promising configuration. Importantly, this does not imply that 12 is an optimal or necessary quantity, as the use of even a single trained encoder followed by weight averaging results in appealing performance. With weight merging algorithms continuing to advance (Wortsman et al., 2022), we believe achieving higher performance with lower training costs is feasible and a potential direction for future work. Given our framework's orthogonal nature to these methods, we expect our proposed practice to be complementary and potentially used in conjunction with these novel techniques (Wortsman et al., 2022; Jang et al., 2024).

D Details on RR-Prompt

In our study, we employ the RR-Prompt strategy to enhance the text understanding capability of the ELVA model. This strategy involves inserting an initial QA turn for text-rich images requiring reasoning, prompting the model to first identify the text within the image. This approach ensures that the model reads the text before engaging in complex reasoning within a dialogue scenario. We apply the RR-Prompt selectively to specific text-rich datasets from our curated dataset, as shown in Table 13, ensuring that not all text-containing datasets are affected, to avoid potential mismatches.

Even within applicable subsets, RR-Prompt is not uniformly applied to all samples; samples with too little or too much text are excluded, and 20% of samples are randomly skipped to help the model balance between reading all text when prompted and performing direct reasoning during inference. This selective application enables robust operation out-of-the-box without notable mismatches. To prevent over-specialization to a single OCR engine, we generate annotations using a combination of MS OCR[10] and CLOVA OCR[11]. Processing text-heavy samples, such as those in DocVQA, takes approximately 4 seconds per call using the CLOVA OCR API. However, we anticipate optimizing this cost in future iterations.

As shown in Table 20, the RR-Prompt incorporates an initial QA turn that instructs the model to read the text using straightforward commands. Furthermore, Table 21 presents a sample where the RR-Prompt is applied. Despite its simplicity, this approach significantly enhances the training outcomes.

E Details on the Construction of CORD-Instruct and Parsing-Bench

E.1 CORD-Instruct

The Consolidated Receipt Dataset (CORD), introduced by Park et al. (2019), is crafted for post-OCR parsing tasks, featuring Indonesian receipt images with structured data in JSON format. Our goal is to adapt this dataset to scenarios where models must generate user-requested structured information in formats like JSON, XML, or Markdown. To achieve this, we construct CORD-Instruct using the OpenAI GPT-3.5 API[12]. A detailed prompt, presented in Table 22, guides the API in generating synthetic user queries and corresponding responses based on the provided structured information.

After generating initial instructions and API responses, we manually filter out any unsuccessful samples. This process involves verifying that samples reflect real-world scenarios, removing ambiguous instructions, and correcting errors in JSON, XML, or Markdown transcriptions. Examples of the generated CORD-Instruct data are shown in Figure 10.

E.2 Parsing-Bench

Parsing-Bench is a dataset designed to fulfill the practical needs of visual document assistants. Inspired by LLaVA-Bench (Liu et al., 2023b), this task requires the model to accurately interpret and analyze input document images to generate the desired structured output. Many industries have expressed a need to extract specific information from

---

[10] `https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/overview-ocr`

[11] `https://clova.ai/ocr/en`

[12] Specifically, `gpt-3.5-turbo-0125`.

Prompt
Carefully decipher the text in this image. Provide the text in the image only.
Investigate the image for any text. Provide the text in the image only.
Examine the image for any letters or words. Provide the text in the image only.
Identify all written characters present in the image. Provide the text in the image only.
Do a careful reading of the image and transcribe all text. Provide the text in the image only.
Inspect the image and write down all readable characters. Provide the text in the image only.
Translate the image content into written text. Provide the text in the image only.
Review the image and offer a transcription of the text. Provide the text in the image only.
Look over the image and jot down all visible text. Provide the text in the image only.
Scrutinize the image for any discernible words or letters. Provide the text in the image only.
Study the image and document all characters found within. Provide the text in the image only.
Assess the image and record any visible words or letters. Provide the text in the image only.
Kindly extract any text or characters from this image. Provide the text in the image only.
Evaluate the image and isolate all written content. Provide the text in the image only.
Assimilate all readable characters within the image. Provide the text in the image only.
Decode any legible text visible in the image. Provide the text in the image only.
Grasp all written elements within the image. Provide the text in the image only.
Conduct a thorough examination of the image and capture all text. Provide the text in the image only.
Peel all readable characters out from the image. Provide the text in the image only.
Interpret all characters from the image. Provide the text in the image only.
Understand and transcribe any textual content from the image. Provide the text in the image only.
Dissect the image and recover any written characters. Provide the text in the image only.
Deduce and present all visible text from the image. Provide the text in the image only.
Inspect the image and translate any visible characters into text. Provide the text in the image only.

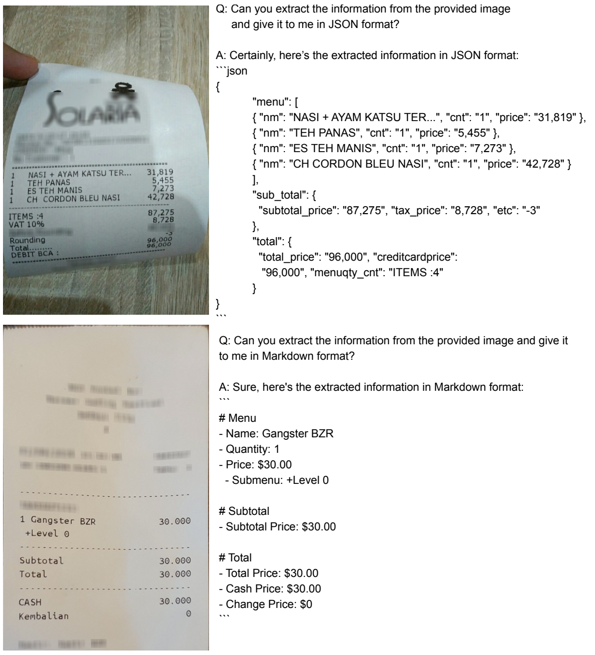
Table 20: **Prompt examples for text reading tasks used in RR-Prompt.**

A data sample from our train set
User: Deduce and present all visible text from the image.
Assistant: 13 PAKET WEEKEND 1 55.000 715.000 6 PAKET WEEKEND 2 (...) 6 Qty: 23 Tunai 1.250.000 Kembali 29.500
User: Can you provide the menu items extracted from the document image in Markdown format?
Assistant: Sure, here are the menu items: - PAKET WEEKEND 1: $55.000 x 13 = $715.000 - PAKET WEEKEND 2: $80.000 x 6 = $480.000 - ES JERUK: $13.000 - ES TEH TAWAR: $3.000 - KOPI HITAM: $8.500 - ES BATU: $1.000

Table 21: **An example data sample from our train set with RR-Prompt applied.**

document images and convert it into formats like JSON or XML (Kim et al., 2022). Parsing-Bench uses Brazilian Identity Documents (BID) (Álysson Soares et al., 2020) and SROIE (Huang et al., 2019) as image sources. Figure 11 visually depicts the overall evaluation process for better comprehension of our method.

To create the benchmark, we first extract context information from images, with prompts detailed in Table 23. An example is shown in Figure 11. This extracted context is utilized during the LLM-as-a-Judge (Zheng et al., 2023a) evaluation process.



```
[redacted]
1   NASI + AYAM KATSU TER...   31,819
1   TEH PANAS                   5,455
1   ES TEH MANIS                7,273
1   CH  CORDON BLEU NASI       42,728
ITEMS :4                       87,275
VAT 10%                         8,728
[redacted]
                                   -3
Rounding                       96,000
Total.........                 96,000
DEBIT BCA :
```

~~~
Q: Can you extract the information from the provided image
   and give it to me in JSON format?

A: Certainly, here’s the extracted information in JSON format:
```json
{
        "menu": [
        { "nm": "NASI + AYAM KATSU TER...", "cnt": "1", "price": "31,819" },
        { "nm": "TEH PANAS", "cnt": "1", "price": "5,455" },
        { "nm": "ES TEH MANIS", "cnt": "1", "price": "7,273" },
        { "nm": "CH CORDON BLEU NASI", "cnt": "1", "price": "42,728" }
        ],
        "sub_total": {
          "subtotal_price": "87,275", "tax_price": "8,728", "etc": "-3"
        },
        "total": {
          "total_price": "96,000", "creditcardprice":
           "96,000", "menuqty_cnt": "ITEMS :4"
        }
}
```
~~~

```
[redacted]
-----------------------------------
1 Gangster BZR              30.000
 +Level 0
-----------------------------------
Subtotal                    30.000
Total                       30.000
-----------------------------------
CASH                        30.000
Kembalian                        0
```

~~~
Q: Can you extract the information from the provided image and give it
to me in Markdown format?

A: Sure, here's the extracted information in Markdown format:
```
# Menu
- Name: Gangster BZR
- Quantity: 1
- Price: $30.00
  - Submenu: +Level 0

# Subtotal
- Subtotal Price: $30.00

# Total
- Total Price: $30.00
- Cash Price: $30.00
- Change Price: $0
```
~~~

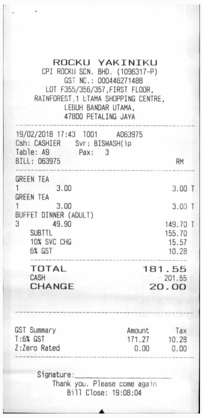
Figure 10: **The generated CORD-Instruct examples.**

We prepare reference answers using the OpenAI GPT-4o API model[13] to serve as benchmarks for evaluation. Evaluation rules, or the rubric, are detailed in Table 23. Finally, the judge model, utilizing the OpenAI GPT-4o API, evaluates the target

---

[13] Specifically, `gpt-4o-2024-05-13`.

Can you summarize the purchased item information, including the name, price, and quantity, in XML format?



```
          ROCKU YAKINIKU
    CPI ROCKU SDN. BHD. (1096317-P)
        GST NO.: 000446271488
     LOT F355/356/357,FIRST FLOOR,
  RAINFOREST,1 UTAMA SHOPPING CENTRE,
         LEBUH BANDAR UTAMA,
         47800 PETALING JAYA
-----------------------------------------
19/02/2018 17:43  T001     A063975
Csh: CASHIER     Svr: BISWASH(ip
Table: A9        Pax:   3
BILL: 063975                          RM
-----------------------------------------
GREEN TEA
1          3.00                    3.00 T
GREEN TEA
1          3.00                    3.00 T
BUFFET DINNER (ADULT)
3          49.90                 149.70 T
   SUBTTL                        155.70
   10% SVC CHG                    15.57
   6% GST                         10.28
-----------------------------------------
   TOTAL                         181.55
   CASH                          201.55
   CHANGE                         20.00
-----------------------------------------
GST Summary              Amount      Tax
T:6% GST                 171.27    10.28
Z:Zero Rated               0.00     0.00
-----------------------------------------
      Signature:____________________
       Thank you. Please come again
          Bill Close: 19:08:04
```

Certainly, here is the purchased item information in XML format:

```xml
<purchased_items>
  <item>
    <name>GREEN TEA</name>
    <quantity>1</quantity>
    <price>3.00</price>
  </item>
  <item>
    <name>GREEN TEA</name>
    <quantity>1</quantity>
    <price>3.00</price>
  </item>
  <item>
    <name>BUFFET DINNER (ADULT)</name>
    <quantity>3</quantity>
    <price>49.90</price>
  </item>
</purchased_items>
```

Elva received 9/10

REVIEW:
8 9

Assistant 1 provided a clear and accurate summary of the purchased items in XML format. However, the XML tags used ("Receipt", "Item", "Name", "Price", "Quantity") are not consistent with the tags used in the original XML provided by the user. This inconsistency might cause confusion or require additional work to integrate with the user's existing data structure.

Assistant 2 also provided a clear and accurate summary of the purchased items in XML format. The tags used ("purchased_items", "item", "name", "quantity", "price") are more consistent with the original XML structure provided by the user, making it easier to integrate and more relevant. Additionally, Assistant 2's response includes a root tag ("purchased_items") that encapsulates all items, which is a good practice for XML structure.

Overall, both assistants performed well, but Assistant 2's response is slightly more aligned with the user's existing data structure and includes better XML formatting practices.

The image is a scanned receipt from ROCKU YAKINIKU, a restaurant located at CPI ROCKU SDN. BHD. (1096317-P), LOT F355/356/357, FIRST FLOOR, RAINFOREST, 1 UTAMA SHOPPING CENTRE, LEBUH BANDAR UTAMA, 47800 PETALING JAYA. The receipt includes the GST number 000446271488. The transaction took place on 19/02/2018 at 17:43, with the cashier identified as CASHIER and the server as BISWASH(ip). The table number is A9, and there were 3 people (Pax: 3). The bill number is 063975.

The items ordered include:
- 1 Green Tea at RM 3.00
- 1 Green Tea at RM 3.00
- 3 Buffet Dinner (Adult) at RM 49.90 each, totaling RM 149.70

The subtotal is RM 155.70. Additional charges include a 10% service charge of RM 15.57 and a 6% GST of RM 10.28, bringing the total to RM 181.55. The payment was made in cash, with RM 201.55 given, resulting in a change of RM 20.00.

The GST Summary shows:
- T: 6% GST on an amount of RM 171.27, with a tax of RM 10.28
- Z: Zero Rated with an amount of RM 0.00 and a tax of RM 0.00

The receipt concludes with a space for a signature and a thank you note, "Thank you. Please come again." The bill was closed at 19:08:04.

```xml
<receipt>
  <restaurant>
    <name>ROCKU YAKINIKU</name><company>CPI ROCKU SDN. BHD. (1096317-P)</company><gst_no>000446271488</gst_no>
    <address><lot>LOT F355/356/357, FIRST FLOOR</lot><location>RAINFOREST, 1 UTAMA SHOPPING CENTRE</location><street>LEBUH BANDAR UTAMA</street><city>47800 PETALING JAYA</city></address>
  </restaurant>
  <transaction>
    <date>19/02/2018</date><time>17:43</time><terminal>T001</terminal><bill_no>A063975</bill_no>
    <cashier>CASHIER</cashier><server>BISWASH(ip)</server><table>A9</table><pax>3</pax><bill>063975</bill>
  </transaction>
  <items>
    <item><name>GREEN TEA</name><quantity>1</quantity><price>3.00</price><total>3.00</total><tax>T</tax></item>
    <item><name>GREEN TEA</name><quantity>1</quantity><price>3.00</price><total>3.00</total><tax>T</tax></item>
    <item><name>BUFFET DINNER (ADULT)</name><quantity>3</quantity><price>49.90</price><total>149.70</total><tax>T</tax></item>
  </items>
  <summary><subtotal>155.70</subtotal><service_charge>15.57</service_charge><gst>10.28</gst><total>181.55</total>
    <payment><method>CASH</method><amount>201.55</amount><change>20.00</change></payment>
  </summary>
  <gst_summary>
    <gst><type>T</type><rate>6%</rate><amount>171.27</amount><tax>10.28</tax></gst>
    <gst><type>Z</type><rate>Zero Rated</rate><amount>0.00</amount><tax>0.00</tax></gst>
  </gst_summary>
  <footer>
    <signature>_________________________</signature><note>Thank you. Please come again</note><bill_close>19:08:04</bill_close>
  </footer>
</receipt>
```

Figure 11: **Overview of Parsing-Bench with an example.** The top left shows the question and model predictions. The top right contains the evaluation review. The bottom section, discussed in Section E.2, shows context extracted from the image using prompts. Evaluation compares two model predictions input into a high-performance LLM judge model. In this example, ELVA is Assistant 2, highlighting its comparative performance.

model outputs by comparing them against these reference answers, using the rubric and context to determine performance scores.

Unlike traditional benchmarks relying on rigid rule-based evaluations, Parsing-Bench offers more adaptability for assessing MLLMs. Parsing-Bench

CORD-Instruct Generation Prompt

Create a synthetic user query requesting information extraction from a given document image. The extracted information should be provided in various formats such as JSON, XML, or Markdown based on the user's request. Users may ask for specific parts of the information, like menu items or payment amounts. Your task is to generate both a user query and the corresponding response from the information extraction system. Ensure the queries and responses vary in detail and format. Sometimes include concise responses, particularly when indicated with the word "concisely."
The provided JSON is to guide your response creation - do not display or mention it in the user queries. Specifically, do not create queries that ask for information extraction from a provided JSON (e.g., "Can you extract the information from the provided JSON" is not allowed). Additionally, you do not need to strictly follow the tag names in the provided JSON while creating your responses (e.g., "nm" can be "name" or "cnt" can be "count"). Return results strictly in the format shown below:

```
Query: I need the payment amount
from the document in this image
in JSON format, answer concisely.
Answer: <<<json
{
  "payment_amount": "$123.45"
}
>>>

Query: Please parse the input document
and provide the menu details in XML.
Answer: Certainly, here is the menu
information in XML format:...

Query: Can you provide the extracted
customer information from the document
image in Markdown?
Answer: Sure, here it is:
<<<
# Customer Information
- Name: John Doe
- Email: john.doe@example.com
>>>
```

Remember, the goal is to include appropriate formatting such as JSON, XML, or Markdown in your responses to correspond with the user's query.

Table 22: **CORD-Instruct data generation prompt.**

includes 30 examples that test models' comprehension and reasoning from document images. We believe that future work can expand Parsing-Bench by increasing the number of examples and encompassing a wider variety of documents and scenarios, enhancing its robustness and applicability. In line with our commitment to open research, we will make these datasets publicly available at https://github.com/naver-ai/elva.

Parsing-Bench BID Context Generation Prompt

Using the provided Brazilian Identity Document image, please compose a comprehensive and detailed caption that encapsulates all the elements depicted in the image. Ensure precision in extracting any text present, maintaining case sensitivity and retaining the exact original form. Begin with a well-written caption in natural language, detailing the image's content, layout, and nuances. Conclude with a well-structured XML format that meticulously documents the extracted information, preserving the image's original layout and details.

Parsing-Bench SROIE Context Generation Prompt

Using the provided scanned receipt image, please compose a comprehensive and detailed caption that encapsulates all the elements depicted in the image. Ensure precision in extracting any text present, maintaining case sensitivity and retaining the exact original form. Begin with a well-written caption in natural language, detailing the image's content, layout, and nuances. Conclude with a well-structured XML format that meticulously documents the extracted information, preserving the image's original layout and details.

Parsing-Bench Evaluation Rubric

We would like to request your feedback on the performance of two AI assistants in response to the user's question displayed above. The user asks the question related to the document image. For your reference, the visual content in the document image is described with a caption, and a corresponding XML file summarizes the information. Please note that the information provided in the caption and XML file may not be completely perfect. Please rate the helpfulness, relevance, accuracy, and level of detail of their responses. Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance. Please first output a single line containing only two values indicating the scores for Assistant 1 and 2, respectively. The two scores should be separated by a space. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment. Keep in mind that the XML file is a summarized version of the full information and may not be directly related to the user's question. Providing unrelated information will not be particularly beneficial. Responses should directly address the user's query in a clear and useful manner to achieve a higher score. Focus especially on whether the responses would be convenient and useful for the user. Minor typographical errors should not heavily impact the scoring. When parsing data formats like XML or JSON, if the user hasn't specified a particular format, minor differences in tag or key names are acceptable as long as the overall meaning is preserved.

Table 23: **Parsing-Bench context generation prompts and LLM-as-a-Judge evaluation rules.**