Text or Pixels? Evaluating Efficiency and Understanding of LLMs with Visual Text Inputs

Yanhong Li[*]
Allen Institute for AI
yanhongl@allenai.org

Zixuan Lan[*]
University of Chicago
zixuanlan@uchicago.edu

Jiawei Zhou
Stony Brook University
jiawei.zhou.1@stonybrook.edu

Abstract

Large language models (LLMs) and their multimodal variants can now process visual inputs, including images of text. This raises an intriguing question: *can we compress textual inputs by feeding them as images to reduce token usage while preserving performance*? In this paper, we show that visual text representations are a practical and surprisingly effective form of input compression for decoder LLMs. We exploit the idea of rendering long text inputs as a single image and provide it directly to the model. This leads to dramatically reduced number of decoder tokens required, offering a new form of input compression. Through experiments on two distinct benchmarks—RULER (long-context retrieval) and CNN/DailyMail (document summarization)—we demonstrate that this *text-as-image* method yields substantial token savings (often nearly half) without degrading task performance.

1 Introduction

Running large language model (LLM) inference on long text inputs is computationally expensive due to the underlying architecture of the Transformer model, where the self-attention mechanism's complexity scales quadratically with the input length (Vaswani et al., 2017). This can be prohibitive when processing long documents (Liu et al., 2023b), interactive dialogues (Zhou et al., 2022), or complex multi-step reasoning (Feng et al., 2025). Even with recent increases in context length, deploying LLMs at scale (e.g., in chat assistants or document analysis) is constrained by throughput and cost per token (Chowdhery et al., 2022; Xiao et al., 2024; Xu et al., 2025; Li et al., 2025a). Reducing the token length of inputs without losing information is therefore highly desirable for improving LLM efficiency (Xu et al., 2024a; Tan et al., 2025; Xing et al., 2025; Li et al., 2025b).

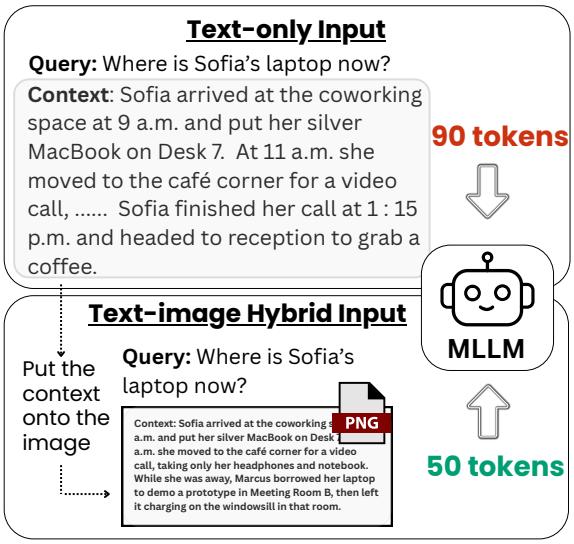


Figure 1: Illustration of our *text-as-image* compression pipeline. Instead of feeding the entire 90-token context to the model (top), we convert the context into a single image and supply only the textual query alongside the image (bottom). The multimodal LLM (MLLM) reads the context from the image, so it processes just 50 visual tokens as input to the LLM decoder—cutting token usage by nearly half while still providing all the information needed to answer the question.

One novel avenue for input compression is to leverage the ability of multimodal LLMs (MLLMs) (Liu et al., 2023a; Fang et al., 2024) to read text from images. The adage "a picture is worth a thousand words" hints that visual representations might convey the equivalent of many text tokens in a compact form. Modern multimodal models like GPT-4 Vision (OpenAI, 2023) and Google Gemini 2.5 (Gemini Team, 2025) can accept images as part of their input and reason over them. This raises the question: *Can we feed an LLM an image of text in lieu of the text itself, to save input tokens and still get the correct output?* Early explorations (discussed in Section 2) suggest that multimodal LLMs can interpret text-based images, but the impact on

[*]These authors contributed equally to this work.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 10564–10578
November 4-9, 2025

efficiency and downstream performance remains under-examined.

In this paper, we present an empirical study of multimodal LLMs that explores using visual text inputs as a form of input compression. By rendering long passages as a single image, the vision encoder produces a compact set of visual tokens for the decoder, directly reducing sequence length without fine-tuning or supervision. On the RULER needle-in-a-haystack task, GPT-4.1-mini and Qwen2.5-VL-72B sustain 97–99% accuracy with up to 58% fewer decoder tokens, and on CNN/DailyMail summarization, this approach *outperforms* two specialized pruning baselines at matched or higher compression rates. Although vision encoding adds some overhead on smaller models, the shorter decoder sequence yields up to 45% end-to-end speedup on larger ones, demonstrating that off-the-shelf multimodal LLMs can treat images as an implicit *compression layer*, preserving performance at nearly **half** the original text-token cost.

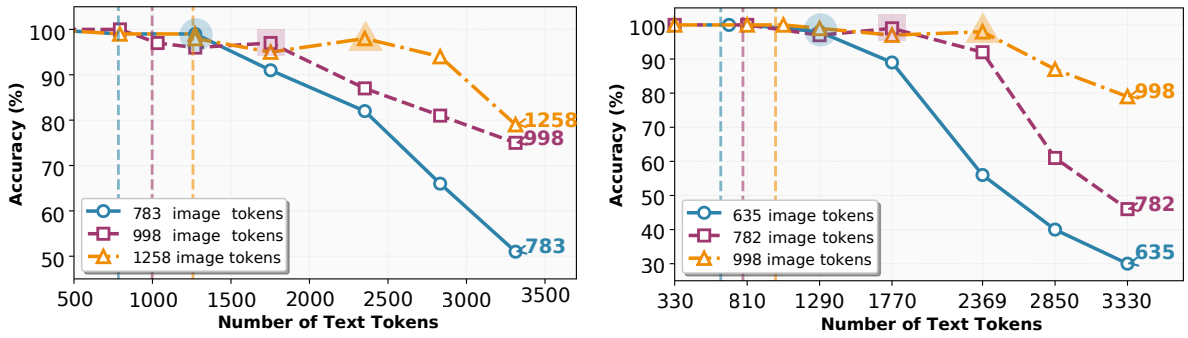
2 Related Work

Multimodal Variations of LLMs Recent advances in LLMs have extended their capabilities beyond text to handle images (Liu et al., 2023a), speech (Wang et al., 2023), and video (Tang et al., 2025). In the visual domain, numerous vision–language models (VLMs) have been developed, including BLIP(-2) (Li et al., 2022, 2023a), Flamingo (Alayrac et al., 2022), LLaVA (Liu et al., 2023a), InternVL (Chen et al., 2024), Qwen-VL (Wang et al., 2024a), PaLI-Gemma (Beyer et al., 2024), and proprietary systems such as GPT-4V (Achiam et al., 2023) and Gemini (Team et al., 2023). A widely adopted architecture comprises a *vision encoder* that extracts image features, a *projection layer* that maps these features into the LLM's token space, and a *text decoder* that jointly processes visual and textual tokens (Liu et al., 2023a; Beyer et al., 2024; Wang et al., 2024a). Importantly, the number of visual tokens produced by this pipeline is usually small—constrained by image size and model design—and can be far fewer than the text tokens needed to represent the same information. This property suggests that multimodal LLMs already embody a form of token compression, motivating our exploration of representing long passages of text directly as images to reduce decoder token usage.

Text as Image Several works have explored the idea of providing text to LLMs via images. Lyu et al. (2025) proposed a pixel-input benchmark, finding that some advanced VLMs can interpret and reason over text in images, though performance can drop without specialized training. Others, such as Lu et al. (2024), have investigated representing large documents as visual patches for handling extended context windows. While these efforts highlight the feasibility of *text-as-image* inputs, they have not systematically evaluated the trade-off between token usage and reasoning performance on multi-step tasks.

Our work also relates to the concept of *hybrid text-vision* prompting, where instructions are given partially in text and partially as an image (Aggarwal and Welleck, 2025). However, prior studies focus more on the novelty of multimodal usage or coding from screenshots, rather than on compression. We instead emphasize the efficiency gains and cost savings arising from replacing a sizeable chunk of text with an image for advanced reasoning tasks.

Information Compression There are many works focusing on context compression (Pradeep et al., 2023; Xu et al., 2024b; Jiang et al., 2024; Li et al., 2025b). Our approach is complementary to *soft prompt* line of work. Extreme Retrieval-Augmented Generation (xRAG) replaces full documents with one dense embedding token, achieving a $50\times$ compression ratio without fine-tuning the LM (Cheng et al., 2024). Instruction-Aware Contextual Compression (IACC) filters noisy RAG passages based on the user query, halving context length while retaining QA accuracy (Hou et al., 2024). Prompt-centric surveys catalogue a spectrum of hard pruning, abstractive summarization, and learned soft tokens that collectively reach 5–$10\times$ compression on reasoning benches (Li et al., 2024b; Jha et al., 2024). Broader reviews on extending LLM context windows emphasize that modality fusion and token dropping are orthogonal, and can be stacked for additive gains (Wang et al., 2024b; Li et al., 2024a). Distinct from these token-level approaches, we compress entire text spans by offloading them to the vision encoder of an off-the-shelf multimodal LLM—treating thousands of tokens as a single image, usually represented with fixed amounts of visual tokens or proportional to the image resolutions—and thus reduce context length to the the LLM decoder without any model fine-tuning.



(a) GPT-4.1-mini

(b) Qwen2.5-VL-72B-instruct

Figure 2: **Accuracy vs. text-token size (m) with fixed visual tokens (k)** for GPT-4.1-mini (left) and Qwen2.5-VL-72B-Instruct (right). Each curve varies the rendered text length at a fixed k; vertical dashed lines mark $m = k$, where text and visual tokens are equal. Both models degrade as text density increases, though Qwen's larger decoder sustains higher ratios before a sharper drop. The *text-token tolerance* (largest m within 3 points of the text-only baseline) is shaded in the plots and reported in Table 1a. Beyond these limits, accuracy falls rapidly, revealing the maximum achievable compression without loss.

3 Methodology

3.1 Problem Formulation

Let the *context* (e.g., a document or multi-turn dialogue) be a sequence of m text tokens, $\mathbf{c} = (t_1, t_2, \ldots, t_m)$, and let the accompanying question be a (usually short) sequence $\mathbf{q} = (q_1, q_2, \ldots, q_{|\mathbf{q}|})$.

Text-only baseline. In the standard setting we feed the concatenated token sequence $\mathbf{s}_{\text{text}} = (\mathbf{c}, \mathbf{q})$ to a *text-only* LLM. The input-length budget is therefore $T_{\text{text}} = m + |\mathbf{q}|$.

Text–image (hybrid) input. To compress long context, we first render it into an image via $I = \mathcal{R}(\mathbf{c})$, using a LaTeX-based typesetting pipeline that preserves the layout and line-breaks of the original text (see Figure 1). A frozen vision encoder[1]

$$\Phi : I \longmapsto (v_1, v_2, \ldots, v_k), \qquad v_j \in \mathbb{R}^d,$$

transforms the image into a *fixed-length* sequence of k *visual tokens*. These embeddings are passed through a projection layer ψ (e.g., a linear map) and become part of the language decoder's input:[2]

$$\mathbf{s}_{\text{img}} = \big(\psi(v_1), \ldots, \psi(v_k), \mathbf{q}\big).$$

The corresponding token budget is

$$T_{\text{img}} = k + |\mathbf{q}|, \quad \text{with } k \ll m \text{ possible in practice.}$$

Compression ratio. We define the *compression ratio*

$$\rho = \frac{T_{\text{text}}}{T_{\text{img}}} = \frac{m + |\mathbf{q}|}{k + |\mathbf{q}|} \approx \frac{m}{k} \quad \text{when } m \,\&\, k \gg |\mathbf{q}|.$$

A higher ρ indicates greater token savings.

3.2 Evaluation Protocol

For each example $(\mathbf{c}, \mathbf{q})$ we run both modes: (1) **Text-only:** Evaluate the LLM on $\mathbf{s}_{\text{text}}$ to obtain answer a_{text}; (2) **Hybrid:** Evaluate the multimodal LLM on $\mathbf{s}_{\text{img}}$ to obtain answer a_{img}. We measure **accuracy** on task-specific metrics; **token usage** $(T_{\text{text}}, T_{\text{img}})$ and thus ρ; and **throughput and latency** (wall-clock time per example). Unless stated otherwise, k is fixed by the vision encoder, so any reduction in m directly translates to lower LLM decoder token cost.

4 Experiments and Results

We test to what extent visual inputs of texts can reduce discrete token consumption in LLM decoders without performance degradation. Long context tasks are especially targeted, including information retrieval and summarization, with configurable context lengths. We test two prominent multimodal LLMs, the proprietary GPT-4.1-mini (OpenAI, 2025) and the open-weight Qwen2.5-VL-72B-Instruct (Bai et al., 2025).[3]

[1] For all experiments we use the native vision module shipped with each multimodal LLM. No fine-tuning or additional supervision is applied.

[2] The number of visual tokens k passed to the language decoder depends on the VLM architecture: some models (e.g., LLaVA (Liu et al., 2023a)) output a fixed number, while others (e.g., Qwen-VL (Wang et al., 2024a)) vary with image sizes. In our experiments, for Qwen models, we define k as the number of visual embeddings passed to the text decoder. For GPT-4.1-mini, we use the input token count returned by the API, which accounts for the visual input.

[3] Experiments on a smaller model Qwen2.5-VL-7B are also presented in Figure 3 and Appendix B.

All text-as-image rendering is performed with `pdflatex` followed by rasterization at 300 dpi. Algorithmic details can be found in Appendix D. During inference, we use the `temperature=0` setting for deterministic outputs and truncate responses to the first newline to obtain concise answers.

4.1 Long-Context Retrieval

Setup. We evaluate our text-as-image compression strategy on the RULER S-NIAH (single needle-in-a-haystack) task (Hsieh et al., 2024), where a single target number (*needle*) is hidden in a long distractor passage (*haystack*). The model must return the exact number, testing long-context retention. For each model, we generate 100 random passages and report *accuracy* (percentage of correct extractions). Since the query $\mathbf{q}$ is short, the effective token budgets simplify to $T_{\text{text}} = m$ and $T_{\text{img}} = k$, giving compression ratio $\rho = m/k$.

How much can we compress? Figures 2 sweep m while holding k fixed. Accuracy remains stable until a critical point $m^\star(k)$, after which it drops sharply. We define $m^\star$ as the largest m within three percentage points of the text-only baseline, referring to this threshold as the *text-token tolerance*. These values are highlighted in the plots (shaded) and reported in Table 1a. For example, at $k = 783$, GPT-4.1-mini tolerates $m^\star \approx 1{,}300$ tokens—equivalent to $\rho \approx 1.9$ compression—*without measurable degradation*. Larger visual budgets ($k = 998, 1{,}258$) increase tolerance to over 2,300 tokens while still saving 42–58% of the decoder context.[4] Qwen2.5-VL follows the same trend but exhibits a steeper accuracy drop once $m^\star$ is exceeded, underscoring that text-token tolerance is both model- and k-dependent.

Token savings and Latency analysis. Table 1a collates $(k, m^\star)$ and confirms that hybrid prompts cut the decoder context **nearly in half** while matching text-only accuracy across both models.[5] To better understand this relationship, Figure 3 plots the text-token tolerance $m^\star$ as a function of the visual token budget k. For all models tested, there is a strong positive correlation: $m^\star$ *increases with* k *at a steady compression ratio* ρ *around* 2. The larger Qwen2.5-VL-72B model consistently demonstrates a superior compression ratio compared to both the 7B model and GPT-4.1-mini. The

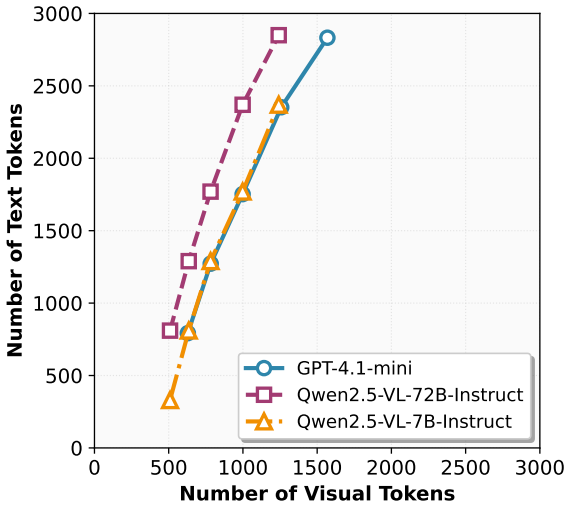


Figure 3: **Text-token tolerance vs. visual token count.** The maximum text tokens $m^\star$ that can be preserved without accuracy loss, plotted against the visual tokens k generated from the image. Results show a consistent reduction of roughly $1/2$ in decoder tokens.

approximately **linear relationship suggests a predictable trade-off** between visual budget and text compression capacity.

We next compare wall-clock generation time (Table 1b). For GPT-4.1-mini the vision processing adds a modest $<$ 1.5s overhead,[6] whereas for Qwen2.5-VL the shorter sequence length outweighs that cost, yielding 25–45% faster inference. Across accuracy and latency, converting long textual contexts into images yields substantial token savings without sacrificing RULER retrieval performance, demonstrating a simple yet effective way to reduce inference cost on large-context tasks.

4.2 Document–Level Summarization

While RULER stresses a model's ability to *retrieve* a single item from a long context, it is *not* expressly designed as a compression benchmark—every token in the haystack is, by construction, irrelevant to the answer. To gauge how well text–as-image compression fares on a genuine *compression* task, we turn to CNN/DailyMail long–document summarization, where *all* input sentences may contribute to the final summary.

Setup. We compare our approach against two widely used token–pruning techniques that operate purely in the text modality: (1) **Select–Context** (Li et al., 2023b): keeps tokens whose self-information (estimated by a small LLM) exceeds a learned

[4] Some visual examples and details in Appendix A.

[5] Additional results on a different long-context reasoning task with Gemini (Appendix C) also confirm this observation.

[6] **Not** exact on model due to API overhead. See Apendix A.

Model	Text–Image (hybrid)			Text-only		
	Image size	k	Acc. (%)	$m^\star(k)$	Acc. (%)	$k/m^\star(k)\downarrow$
GPT	600×800	783	99	1,272.4	100	0.61
	600×1000	998	97	1,752.5	100	0.57
	750×1000	1,258	98	2,352.2	100	0.53
QWEN	600×800	635	98	1,289.6	100	0.49
	600×1000	782	99	1,769.7	100	0.44
	750×1000	998	98	2,369.4	100	0.42

(a) **RULER accuracy and token statistics.** k is the visual token count, $m^\star(k)$ the maximum *text-token tolerance*, and $k/m^\star(k)$ the relative token footprint. Text-image input reduces the decoder tokens by 38–58%.

Model	k	Time_{img} (s)↓	$\text{Time}_{\text{text}}$ (s)↓
GPT*	783	1.29	0.60
	998	1.75	0.61
	1,258	1.95	0.58
QWEN	635	2.81	3.61
	782	3.04	4.16
	998	3.35	5.09

(b) **End-to-end latency.** Hybrid inputs add modest overhead for GPT but *reduce* total time for Qwen thanks to smaller decoder contexts. *measured from API responses; see Appendix A.

Table 1: RULER S-NIAH long-context retrieval accuracy, text-as-image compression statistics, and model latency. GPT refers to GPT-4.1-mini and QWEN denotes Qwen2.5-VL-72B-Instruct.

Model	Method	Remaining $k\downarrow$	BERTScore	ROUGE-L	ROUGE-1	ROUGE-2	$\text{ROUGE-L}_{\text{sum}}$
GPT	Baseline (text-only)	m=693	**86.25**	**16.26**	**23.78**	**8.60**	**18.91**
	Text-as-image (ours)	225 (−67%)	85.33	15.31	21.98	7.40	17.75
	Select-Context	295 (−57%)	85.01	12.79	18.82	5.30	15.71
	LLMLingua-2	265 (−62%)	85.25	13.75	20.42	7.00	16.60
QWEN	Baseline (text-only)	m=726	**86.37**	**17.70**	**25.18**	**9.47**	**20.77**
	Text-as-image (ours)	279 (−62%)	85.64	15.53	23.28	7.54	19.16
	Select-Context	181 (−75%)	84.60	13.40	20.36	5.13	17.63
	LLMLingua-2	258 (−64%)	85.46	15.32	22.95	7.09	18.94

Table 2: CNN/DailyMail document summarization with token compression baselines. GPT refers to GPT-4.1-mini and QWEN denotes Qwen2.5-VL-72B-Instruct. m is the uncompressed text length, k the remaining decoder tokens after compression. Percentage shows token reduction relative to m. At *similar* compression rates, rendering the document as an image beats both token-level baselines on every metric.

threshold. (2) **LLMLingua-2** (Pan et al., 2024): trains a Transformer encoder to predict, token by token, whether to retain or discard.

We compute the average input length in CNN/-DailyMail and use it to set the image resolution in our text-to-image pipeline, yielding visual token inputs at roughly half the original context length—the optimal ratio identified in previous section. This image size is applied consistently across the task. For fair comparison, baselines are configured with the same decoder token compression ratio. Summary quality is evaluated with ROUGE (Lin, 2004) and BERTSCORE (Zhang et al., 2020).

Results and Discussion. The evidence in Table 2 shows that, even though text–image compression was *not* tailored for summarization, it yields stronger summaries than two specialized pruning methods while retaining only $\sim$40% of the original tokens. We stress that the aim of this paper is to *characterize* the compression capacity of visual inputs, not to introduce yet another SOTA compression model. Nevertheless, the unexpectedly strong results suggest that our simple rendering trick is a competitive—and orthogonal—alternative to learned token-selection approaches. Future work could further combine text token pruning *before* rendering, stacking the benefits of both paradigms.

5 Conclusion

Our primary goal is to answer the question: *"How many tokens can be saved by replacing text with an image without harming downstream performance?"* By converting long contexts into visual form, we achieved nearly two-fold reductions in decoder token count while preserving task accuracy on both retrieval (RULER S-NIAH) and generation (CNN/DailyMail summarization) benchmarks. The approach is model- and task-agnostic, requires no parameter updates, and can even lower latency on large decoders. Our findings suggest several directions for future work: (i) applying token-level pruning before visual rendering to further compound compression gains; and (ii) expanding the approach to other domains (e.g. math) where most prompt tokens are critical and thus difficult to prune at the token level. We hope this study will spark broader exploration of modality shifting as a complementary axis for scaling the usability and efficiency of Large language models.

Limitations

Despite showing promising token savings on short to medium context scenarios, our work has not yet fully evaluated the impact of text-as-image prompting on *extremely large* contexts that span tens of thousands of tokens or more. Real-world applications such as document analysis or in-depth conversational histories may require specialized approaches (e.g., chunking, retrieval) to ensure reliable performance at these larger scales. Furthermore, our experiments focus on a limited number of benchmarks, leaving open questions about performance on other domains (e.g., medical, legal) and tasks (e.g., coding, translation).

Acknowledgment

We thank the Google Gemma Academic Program for their partial support of Jiawei Zhou and for providing computational resources.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pranjal Aggarwal and Sean Welleck. 2025. Programming with pixels: Computer-use meets software engineering. *Preprint*, arXiv:2502.18525.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, and 1 others. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Yixiong Fang, Ziran Yang, Zhaorun Chen, Zhuokai Zhao, and Jiawei Zhou. 2024. From uncertainty to trust: Enhancing reliability in vision-language models with uncertainty-guided dropout decoding. *arXiv preprint arXiv:2412.06474*.

Yiyang Feng, Yichen Wang, Shaobo Cui, Boi Faltings, Mina Lee, and Jiawei Zhou. 2025. Unraveling misinformation propagation in llm reasoning. *arXiv preprint arXiv:2505.18555*.

Google Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Haowen Hou, Fei Ma, Binwen Bai, Xinxin Zhu, and Fei Yu. 2024. Enhancing and accelerating large language models via instruction-aware contextual compression. *arXiv preprint arXiv:2408.15491*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *Preprint*, arXiv:2404.06654.

Siddharth Jha, Lutfi Eren Erdogan, Sehoon Kim, Kurt Keutzer, and Amir Gholami. 2024. Characterizing prompt compression methods for long context inference. *arXiv preprint arXiv:2407.08892*.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Preprint*, arXiv:2406.10149.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Yanhong Li, Karen Livescu, and Jiawei Zhou. 2025a. Chunk-distilled language modeling. In *The Thirteenth International Conference on Learning Representations*.

Yanhong Li, David Yunis, David McAllester, and Jiawei Zhou. 2025b. Context-efficient retrieval with factual decomposition. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 178–194, Albuquerque, New Mexico. Association for Computational Linguistics.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023b. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, Singapore. Association for Computational Linguistics.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024a. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. *Preprint*, arXiv:2407.16833.

Zongqian Li, Yixuan Su, and Nigel Collier. 2024b. 500xcompressor: Generalized prompt compression for large language models. *arXiv preprint arXiv:2408.03094*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023b. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Yujie Lu, Xiujun Li, Tsu-Jui Fu, Miguel Eckstein, and William Yang Wang. 2024. From text to pixel: Advancing long-context understanding in mllms. *Preprint*, arXiv:2405.14213.

Zhiheng Lyu, Xueguang Ma, and Wenhu Chen. 2025. Pixelworld: Towards perceiving everything as pixels. *Preprint*, arXiv:2501.19339.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2025. Gpt-4.1-mini. `https://platform.openai.com/docs/models/gpt-4.1-mini`. Accessed: 2025.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981, Bangkok, Thailand. Association for Computational Linguistics.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankVicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv:2309.15088*.

Xudong Tan, Peng Ye, Chongjun Tu, Jianjian Cao, Yaoxin Yang, Lin Zhang, Dongzhan Zhou, and Tao Chen. 2025. Tokencarve: Information-preserving visual token compression in multimodal large language models. *Preprint*, arXiv:2503.10501.

Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, and 1 others. 2025. Video understanding with large language models: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Mingqiu Wang, Wei Han, Izhak Shafran, Zelin Wu, Chung-Cheng Chiu, Yuan Cao, Nanxin Chen, Yu Zhang, Hagen Soltau, Paul K Rubenstein, and 1 others. 2023. Slm: Bridge the thin gap between speech and text foundation models. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. 2024b. Beyond the limits: A survey of techniques to extend the context length in large language models. *Preprint*, arXiv:2402.02244.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. 2024. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. *Preprint*, arXiv:2410.10819.

Ling Xing, Alex Jinpeng Wang, Rui Yan, Xiangbo Shu, and Jinhui Tang. 2025. Vision-centric token compression in large language model. *arXiv preprint arXiv:2502.00791*.

Fangyuan Xu, Tanya Goyal, and Eunsol Choi. 2024a. Recycled attention: Efficient inference for long-context language models. *OpenReview preprint*. ICLR 2025 submission.

Fangyuan Xu, Tanya Goyal, and Eunsol Choi. 2025. Refreshkv: Updating small kv cache during long-form generation. *Preprint*, arXiv:2411.05787.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024b. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Jiawei Zhou, Jason Eisner, Michael Newman, Emmanouil Antonios Platanios, and Sam Thomson. 2022. Online semantic parsing for latency reduction in task-oriented dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1576.

Appendix

A Results Across Context Lengths and Image Sizes on RULER

A central challenge is how to balance visual and textual tokens under a fixed sequence length. In many patch-based vision encoders used by VLMs such as Qwen, higher image resolution produces more patches, and thus more visual tokens, which offsets efficiency benefits of text-as-image processing for token reduction. Conversely, lowering resolution reduces token usage, but risks discarding semantically important visual details. This trade-off becomes especially critical in long-context settings such as retrieval-style reasoning or the RULER benchmark, where even small shifts in token allocation can lead to large changes in accuracy.

Our experiments demonstrate that model accuracy is highly sensitive to the effective resolution of image inputs, which directly controls the number of visual tokens generated. We observe a consistent trend across benchmarks: *when visual tokens account for roughly one half of the total context window, performance remains nearly indistinguishable from that achieved with the original, uncompressed textual context.* This $\frac{1}{2}$ allocation emerges as a near-optimal operating point, striking a balance between preserving visual fidelity and maintaining sufficient capacity for textual information. The finding aligns with broader evidence of redundancy in long-context models, where moderate compression often preserves accuracy despite substantial reductions in token count.

To illustrate, Figure 5–7 present representative cases from the RULER needle-in-a-haystack task: at a context length of 1500 with an image resolution of 600×800 pixels (Figure 5), at 2000 with 600×1000 pixels (Figure 6), and at 2500 with 750×1000 pixels (Figure 7). In all three cases, the model successfully recovers nearly *all* embedded image information, achieving accuracy close to 100%.

In contrast, when the context length is extended to 3000 tokens with a 600×1000 image (Figure 8), accuracy degrades substantially despite the increase in available tokens. This illustrates that recognition accuracy depends not only on the absolute context length but also on preserving a balanced allocation between textual and visual tokens. Simply enlarging the context window is therefore insufficient to ensure stable performance.

Collectively, these results indicate that tuning image resolution relative to the available context budget is critical for multimodal reasoning. The **observed $\frac{1}{2}$ allocation** (or compression ratio ρ of 2) emerges as a practical heuristic for balancing efficiency and fidelity, though further validation is required to assess its robustness across tasks, datasets, and model architectures.

Inference Latency Analysis To measure the inference latency of text-only input to the LLM decoder and text-image hybrid input to the full multimodal model (shown in Table 1b), we run the full RULER S-NIAH test set and report the average wall-clock time per sample. No batching is used. For the proprietary GPT-4.1-mini, latency is measured from API response times. For Qwen2.5-VL-72B-Instruct which is open sourced, inference is performed on 8 Nvidia RTX A6000 GPUs using the standard Hugging Face implementation without batching, vLLM, or other speedups.

Note that API response times for GPT-4.1-mini may *not reflect pure model latency*, as they include system overhead such as request queuing, routing, network latency, and data transmission. Since image payloads are larger than text, this overhead is likely greater for text-as-image inputs, partially explaining the higher latency reported in Table 1b.

B Results on Qwen2.5-VL-7B-Instruct

To understand the role of model scale in visual text compression, we replicate the long-context retrieval experiment from Section 4.1 on the much smaller **Qwen2.5-VL-7B-Instruct** model.[7] The results, plotted in Figure 4, show that while the general trend of performance degradation holds, the 7B model is significantly more sensitive to text density than its 72B counterpart.

The text-token tolerance of the 7B model is substantially lower across all corresponding visual token budgets (k). For instance, with $k = 998$ visual tokens, the 72B model maintains over 97% accuracy up to nearly 2,400 text tokens, whereas the 7B model's accuracy drops below 95% after only 2,000 text tokens. This highlights that larger model scale provides greater capacity to robustly process and reason over densely packed textual information presented visually, making it a critical factor for achieving high compression ratios without performance loss.

[7] https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct.

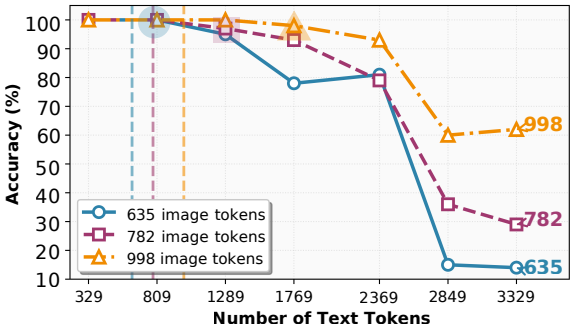


Figure 4: **Qwen2.5-VL-7B-Instruct: accuracy vs. text-token size (m) with fixed visual tokens (k).** Each curve varies the amount of rendered text for a fixed visual token size. Compared to the 72B version (Figure 2b), the smaller 7B model exhibits a much steeper performance degradation as text density increases, indicating that model scale is a critical factor for effective visual text processing.

C BABILong 1k Benchmark

To further validate our findings, we evaluate on the BABILONG benchmark (Kuratov et al., 2024), which is specifically designed to test the limits of long-context reasoning in LLMs. The benchmark extends the classical bAbI tasks to much longer contexts, where the relevant information must be retrieved from sequences containing up to 1k distractor tokens. This makes it well suited for assessing whether multimodal token allocation strategies remain effective in long-context scenarios.

We follow the standard setup of the BABILONG 1k variant, which consists of ten subsets (QA1–QA10) covering a range of reasoning tasks such as single- and multi-supporting fact retrieval, coreference, and induction. For this experiment, we use `gemini-2.5-flash-preview-04-17` (Gemini Team, 2025).

Since Gemini is closed-source, we estimate visual token usage from the input/output token statistics returned by the API. Specifically, we record both text and image token counts for each query, enabling us to analyze how token allocation is distributed between modalities.

Table 3 shows that across QA1–QA10, performance is stable when image tokens constitute **roughly half** the count of the original total context text tokens. In this configuration, the model achieves strong and well-aligned accuracies for both text (0.91) and image (0.83), confirming that the $\frac{1}{2}$ **allocation rule** observed in the RULER experiments generalizes to BABILong as well. This provides additional evidence that balancing image and text tokens at a near-equal ratio offers a practical operating point for multimodal long-context reasoning.

D ConTexImage: A Text-to-Image Conversion Pipeline

To standardize our experiments, we require a consistent way to convert textual sequences into image inputs of controlled resolution. We therefore design a lightweight text-to-image pipeline, which we term **ConTexImage** (Algorithm 1). ConTexImage renders arbitrary text into rasterized images while automatically adjusting font size to achieve a target content density. This ensures that the generated images preserve readability, maintain consistent tokenization patterns, and remain comparable across different resolutions.

LaTeX Rendering: The cleaned text is embedded into a minimal LaTeX document and compiled using `tectonic`[8].

The pipeline consists of three main stages:

1. **Preprocessing:** Input text is normalized by replacing typographic symbols (e.g., curly quotes, dashes, ellipses) and escaping LaTeX special characters. This step guarantees compatibility with the rendering backend.

2. **LaTeX Rendering:** The cleaned text is embedded into a minimal LaTeX document and compiled using `tectonic`. The output PDF page is rasterized into an image at a specified DPI and then resized to the target resolution (e.g., 600×800, 600×1000).

3. **Adaptive Font Optimization:** To maximize visual fidelity, the algorithm searches over candidate font sizes and evaluates the proportion of the image occupied by text (*fill ratio*). The largest font size that meets a pre-defined target fill ratio (default 0.8) is selected, ensuring both legibility and balanced whitespace.

The resulting images are consistent across different contexts and allow us to precisely control the number of image tokens relative to text tokens. More details are illustrated in Algorithm 1.

[8] https://github.com/tectonic-typesetting/tectonic.

Task	TxtAcc	ImgAcc	TxtIn	TxtOut	ImgIn	ImgOut
QA1	1.00	0.96	846.9	9.0	440.0	9.1
QA2	0.85	0.65	877.0	7.0	466.0	7.0
QA3	0.87	0.59	946.2	10.0	531.0	10.0
QA4	1.00	0.98	810.5	1.1	399.0	1.1
QA5	0.92	0.89	881.0	1.0	465.4	1.0
QA6	0.99	0.96	829.0	1.0	430.0	1.0
QA7	0.54	0.48	863.1	1.0	451.0	1.0
QA8	0.99	0.95	862.7	1.1	468.0	1.1
QA9	1.00	0.96	827.7	1.0	425.0	1.0
QA10	0.98	0.92	875.1	1.0	466.0	1.0
Avg	0.91	0.83	861.9	3.3	454.1	3.3

Table 3: **Gemini-2.5-flash-preview-04-17 on BABILONG 1k.** "TxtAcc" is accuracy for text input. "ImgAcc" is accuracy for image input. "TxtIn" and "TxtOut" are the average text input/output tokens. "ImgIn" and "ImgOut" are the average image input/output tokens.

Algorithm 1: **ConTexImage**: A Context-Aware Text-to-Image Pipeline

```
def escape_latex_special_chars(text):
    """Replace LaTeX special characters with safe tokens."""
    escape_map = {"\\": r"\textbackslash{}", "&": r"\&", "%": r"\%",
                  "$": r"\$", "#": r"\#", "_": r"\_", "{": r"\{", "}": r"\}",
                  "~": r"\textasciitilde{}", "^": r"\textasciicircum{}"}
    for k, v in escape_map.items():
        text = text.replace(k, v)
    return text

def text_to_image(text, width, height, dpi=300, target_fill_ratio=0.7):
    # Step 1: Preprocessing
    text = normalize_typography(text)        # replace curly quotes, dashes, ellipses
    text = escape_latex_special_chars(text)  # escape special symbols for LaTeX

    # Step 2: Font size search
    best_image, best_ratio = None, 0
    for font_size in candidate_font_sizes(descending=True):
        tex_doc = build_latex_template(text, font_size, width, height, margin=10)
        pdf = compile_with_tectonic(tex_doc)                  # LaTeX -> PDF
        image = convert_pdf_to_image(pdf, dpi=dpi,
                                     resize=(width, height))  # PDF -> raster
        ratio = calculate_fill_ratio(image)                   # bounding box occupancy

        if ratio > best_ratio:     # update best so far
            best_ratio, best_image = ratio, image
        if ratio >= target_fill_ratio:
            break

    return best_image

def generate_images(dataset, output_dir, width, height, dpi=300):
    """Batch conversion for all documents in dataset."""
    for doc in dataset:
        text = doc["input"]
        img = text_to_image(text, width, height, dpi=dpi)
        save_image(img, path=output_dir + f"/{doc['doc_id']}.png")
```


A special magic number is hidden within the following text. Make sure to memorize it. I will quiz you about the number afterwards. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is
yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. One of the special magic numbers for watchful-smuggling is: 2056700. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is
yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is
yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is
yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is
yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. What is the special magic number for
watchful-smuggling mentioned in the provided text?

Figure 5: Rendered image input for the RULER task at context length 1500 (600 × 800 image resolution). Here there is almost **no** accuracy degradation. This example illustrates how textual sequences are converted into rasterized images for multimodal processing.

Figure 6: Rendered image input at context length 2000 (600 $\times$ 1000 image resolution). Here there is almost **no** accuracy degradation. The figure demonstrates scaling of resolution while preserving readability and model performance.

A special magic number is hidden within the following text. Make sure to memorize it. I will quiz you about the number afterwards. The grass is green. The sky is blue. The sun is yellow. Here
we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. One of the special magic numbers for elite-butterfly is: 4132801. The grass
is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green.
The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky
is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue.
The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun
is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow.
Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we
go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There
and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again.
The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The
grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green.
The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky
is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue.
The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun
is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow.
Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we
go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There
and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again.
The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The
grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green.
The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky
is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue.
The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun
is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow.
Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we
go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There
and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again.
The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The
grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green.
The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky
is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue.
The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun
is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow.
Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we
go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There
and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. What is the special magic number for elite-butterfly mentioned in the provided text?

Figure 7: Rendered image input at context length 2500 (750×1000 image resolution). Here there is almost **no** accuracy degradation. Increased resolution produces more visual tokens while maintaining comparable visual fidelity and model performance.

A special magic number is hidden within the following text. Make sure to memorize it. I will quiz you about the number afterwards. The grass
is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here
we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green.
The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The
sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There
and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and
back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue.
The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The
sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again.
The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is
yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The
grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow.
Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here
we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green.
The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The
sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There
and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and
back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue.
The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The
sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again.
The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is
yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The
grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow.
Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here
we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green.
The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The
sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There
and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and
back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue.
The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The
sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again.
The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is
yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The
grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow.
Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here
we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green.
The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The
sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There
and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and
back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue.
The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The
sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again.
The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is
yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The
grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow.
Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is
green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here
we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green.
The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The
sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There
and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and
back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue.
The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back
again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The
sun is yellow. Here we go. There and back again. One of the special magic numbers for recondite-airplane is: 4951063. The grass is green.
The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go.
There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The
sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There
and back again. The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again. The grass is green. The sky is
blue. The sun is yellow. Here we go. There and back again. What is the special magic number for recondite-airplane mentioned in the
provided text?

Figure 8: Rendered image input at context length 3000 (600 $\times$ 1000 image resolution). Here accuracy **degrades** substantially. This setting illustrates the tradeoff between tolerable text token budget and image resolution to maintain model performance.