Glyph: Scaling Context Windows via Visual-Text Compression

Jiale Cheng[1,2*] , **Yusen Liu**[2*] , **Xinyu Zhang**[2*] , **Yulin Fei**[2*] , **Wenyi Hong**[2,3]
Ruiliang Lyu[2] , **Weihan Wang**[2] , **Zhe Su**[2] , **Xiaotao Gu**[2] , **Xiao Liu**[2,3] , **Yushi Bai**[2,3]
Jie Tang[3], **Hongning Wang**[1] , **Minlie Huang**[1†]
[1]The Conversational Artificial Intelligence (CoAI) Group, Tsinghua University
[2]Zhipu AI
[3]The Knowledge Engineering Group (KEG), Tsinghua University
chengjl23@mails.tsinghua.edu.cn, aihuang@tsinghua.edu.cn

Abstract

Large language models (LLMs) conventionally represent text as sequences of discrete tokens, making long-context scaling largely a matter of processing more tokens more efficiently. We instead explore a complementary direction: increasing how much original context each token represents. To this end, we introduce Glyph, a framework that renders long texts into compact visual pages and processes them with a vision-language model (VLM), allowing a fixed context window to cover substantially more text. To make visual compression practical, Glyph combines continual pre-training on rendered long-text data, an LLM-driven genetic search to identify rendering configurations that balance compression and task performance, and post-training with supervised fine-tuning and reinforcement learning. Across multiple long-context benchmarks, Glyph achieves 3–4× token compression while maintaining performance comparable to strong text-only LLMs such as Qwen3-8B, with over 4× faster prefilling and decoding and 2× faster supervised fine-tuning. Under more aggressive compression, a VLM with a 128K context window can handle tasks that would otherwise require up to 1M input tokens. Our code and model are released at https://github.com/thu-coai/Glyph.

1 Introduction

Large language models (LLMs) are increasingly required to reason over books, repositories, reports, and long interaction histories, making long-context modeling a central capability (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023; GLM et al., 2024; Yang et al., 2025; Bai et al., 2024; Comanici et al., 2025). In current LLMs, however, longer inputs almost always mean more tokens: text is represented as sequences of discrete tokens, and scaling to longer contexts therefore becomes largely a problem of processing more tokens more efficiently. This formulation makes long-context scaling inherently expensive, since extending context windows to hundreds of thousands or even millions of tokens incurs prohibitive training and inference cost in both computation and memory, severely limiting practical deployment.

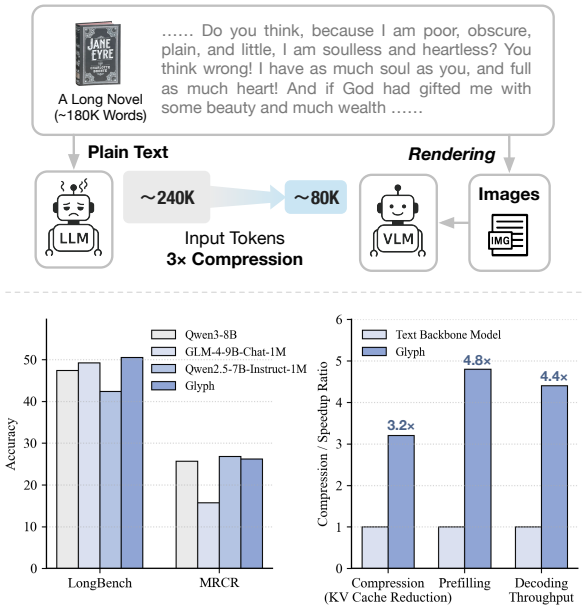


Figure 1: (Upper) Comparison of two paradigms for long-context modeling: conventional approaches process plain text directly with LLMs, whereas the proposed visual-based paradigm, Glyph, renders text into compact images to achieve substantial input-token compression. (Lower) Glyph achieves competitive performance on LongBench and MRCR, meanwhile providing significant compression and inference speedup over its text backbone model on 128K-token inputs.

Most existing approaches to long-context modeling tackle this challenge by making it easier for models to handle more tokens, either by supporting longer token sequences or by reducing the cost of processing them. One line of work extends positional encodings, such as YaRN (Peng et al., 2023), enabling pretrained models to accept longer inputs without additional long-context training. However,

* Core contributors.
† Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 37145–37158
July 2-7, 2026

such methods neither reduce inference cost nor reliably preserve performance when extrapolated to much longer sequences (Wu et al., 2024). Another line of work improves efficiency by modifying the attention mechanism, including sparse or linear attention (Huang et al., 2023; Yang et al., 2024; Peng et al., 2025; Chen et al., 2025a). Yet these methods do not reduce the number of input tokens required to represent the original context, so the overall computational burden remains substantial as inputs grow longer. Retrieval-augmented approaches (Laban et al., 2024; Yu et al., 2025a) reduce the amount of text processed at inference time through external retrieval, but may miss globally relevant information and can introduce additional latency. Taken together, these directions mainly improve how long token sequences are processed or selected, rather than changing the amount of original context each input token can carry.

In this work, we explore a complementary direction for long-context modeling by redesigning the input representation to encode more original text within the same input-token budget. To this end, we propose Glyph, which renders long documents into compact visual pages for processing by a vision-language model (VLM). Compared with standard text tokenization, this representation exposes the much higher compression that VLMs can offer for long-text modeling. As a result, a fixed-context VLM can process substantially more original material than a text-only LLM with the same input budget, without extending the token window or relying on external retrieval. For example, a conventional 128K-context LLM cannot accommodate a full novel such as *Jane Eyre* (≈240K text tokens) without truncation, which can undermine questions requiring global coverage. In contrast, Glyph can render the same book into compact visual pages (e.g., ≈80K visual tokens), allowing a 128K-context VLM to process the full text and answer such questions more reliably.

However, realizing this compression advantage in practice requires more than rendering text as images. The central challenge is to preserve enough textual fidelity for downstream reasoning while still achieving substantial compression. To meet this challenge, Glyph combines three stages: continual pre-training on rendered long-text data to transfer long-context capability to compressed visual inputs, an LLM-driven genetic search to identify rendering configurations that balance compression and task performance, and post-training with supervised fine-tuning and reinforcement learning to further align the model with such inputs. We additionally incorporate an auxiliary OCR objective to improve fine-grained text recognition and tighten the alignment between visual and textual representations. All these components turn visual-text compression into a trainable and scalable approach to long-context modeling.

Extensive experiments show that Glyph demonstrates the high compression ratio that VLMs can offer for language modeling: substantially more original text can be encoded within the same input-token budget while preserving competitive performance. Across multiple long-context benchmarks, Glyph achieves 3–4× token compression while remaining comparable to strong text-only LLMs such as Qwen3-8B. This higher compression also yields clear practical benefits, including up to 4.8× faster prefilling, 4.4× faster decoding, and around 2× faster supervised fine-tuning. Under more aggressive compression, a 128K-context VLM can further handle tasks whose original text spans up to 1M tokens. Moreover, training on rendered text improves performance on real-world multimodal long-context tasks such as document understanding. These findings point to an orthogonal scaling direction for long-context modeling: usable context can be expanded not only by processing more tokens, but also by encoding more original text within the same token budget.

2 Related Work

2.1 Long-Context Modeling

Most research on long-context modeling has focused on scaling the length of token-based context representations through architectural and training advances. On the architectural side, prior work has explored sparse and hierarchical attention (Yang et al., 2016; Beltagy et al., 2020; Huang et al., 2023; Yang et al., 2024; Peng et al., 2025; Chen et al., 2025a), positional interpolation and extrapolation (Su et al., 2021; Press et al., 2021; Sun et al., 2022; Peng et al., 2023), and content-aware positional encodings (Chen et al., 2025b; Zhu et al., 2024). On the training side, LongAlign (Zhang et al., 2024) improves long-context instruction tuning through dedicated data construction and loss weighting, LongLoRA (Chen et al., 2024) combines shifted sparse attention with parameter-efficient fine-tuning, LongRecipe (Wang et al., 2024b) improves long-context efficiency through

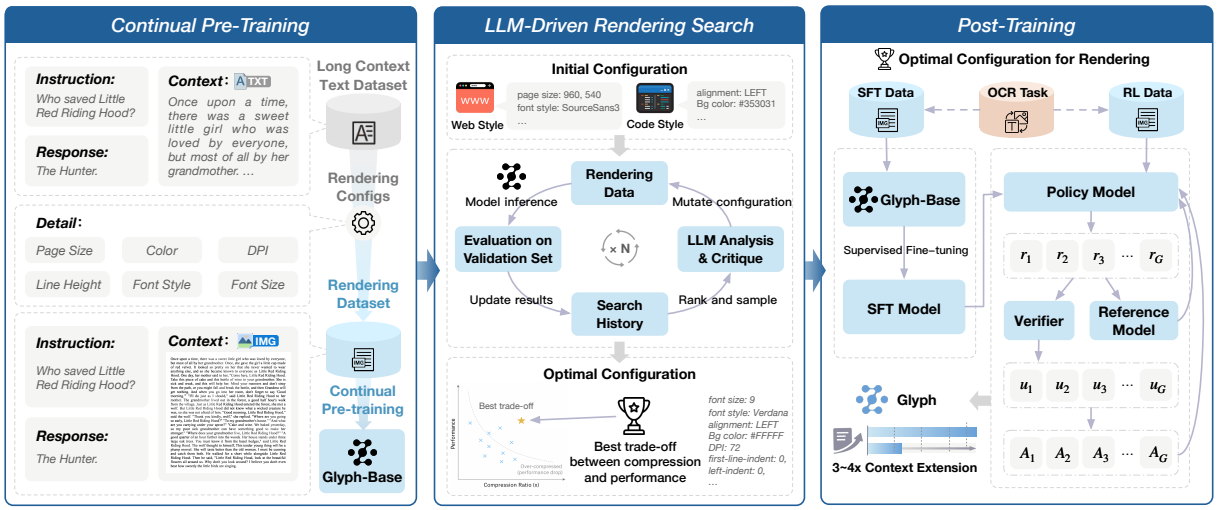


Figure 2: Glyph consists of three stages: continual pre-training on rendered long-text data, LLM-driven genetic search for optimal rendering configurations, and post-training with SFT and RL. These stages enable efficient long-context modeling through visual-text compression.

token analysis, index transformation, and optimization, and ProLong (Liu et al., 2024b) adopts a data-centric strategy by selecting samples with long-range dependencies.

Despite their differences, these methods mainly extend effective token length or reduce the cost of processing long token sequences. Our work instead explores a complementary direction: increasing how much original text can be encoded within the same input-token budget through visual-text compression. A concurrent work, DeepSeek-OCR (Wei et al., 2025), also suggests that visual representations can act as compressed carriers of textual content. However, it is not developed as a general language modeling approach. In contrast, we study whether visual-text compression can serve as a general high-compression mechanism complementary to existing token-length scaling methods.

2.2 Vision-Language Models

Vision-language models extend language models to jointly process textual and visual inputs. Representative systems such as PALI (Chen et al., 2022), LLaVA (Liu et al., 2023), and CogVLM (Wang et al., 2024a) show how visual encoders can be effectively integrated with strong language backbones. Subsequent work further improves VLMs through stronger base LLMs and large-scale vision-language pre-training (Hong et al., 2024a; Bai et al., 2025), while also expanding related multimodal systems to additional modalities such as video and audio (Hurst et al., 2024).

Particularly relevant to our work, recent VLMs have shown strong capabilities in text-rich image understanding and optical character recognition (OCR) (Hong et al., 2024b; Liu et al., 2024a). These results suggest that visual tokens can serve not only as carriers of natural-image semantics, but also as an alternative representation of text itself, often in a more compact form. Building on this capability, we investigate whether visual representations can offer high compression for general long-context modeling.

3 Method

We present Glyph, a complementary approach to long-context modeling that scales usable context through visual-text compression. Instead of extending token-based context windows, Glyph renders ultra-long textual inputs into compact visual pages and processes them with a VLM. This redesign of the input representation allows substantially more original text to be encoded within the same input-token budget, revealing the compression advantage of VLMs for long-text modeling. To make this compression effective in practice, Glyph further introduces an LLM-driven genetic search to identify rendering configurations that best balance compression and downstream performance.

3.1 Overall Framework

As illustrated in Figure 2, Glyph consists of three tightly coupled stages: (1) Continual Pre-Training on rendered long-text data, which teaches the VLM to read and reason over compressed textual inputs under diverse visual styles; (2) LLM-Driven Rendering Search, which identifies task-appropriate rendering configurations that balance compression

and performance; and (3) Post-Training, including SFT and RL under the selected configuration to further adapt the model to downstream long-context tasks. These stages make high-compression long-context modeling with VLMs effective in practice.

3.2 Task Definition

Task Formulation. We formalize the standard long-context instruction following task as a triple $(\mathcal{I}, \mathcal{C}, \mathcal{R})$, where $\mathcal{I}$ is a concise user instruction specifying the core goal, $\mathcal{C}$ is an ultra-long textual context, and $\mathcal{R}$ is the target response. The conventional learning objective is to maximize

$$P(\mathcal{R} \mid \mathcal{I}, \mathcal{C}),$$

i.e., to generate an accurate response conditioned on both the instruction and the long textual context.

Scaling this token-based formulation to million-token contexts, however, incurs prohibitive memory and computation costs. To overcome these limitations, we reformulate the input representation through *visual compression*. Instead of directly feeding $\mathcal{C}$ as text tokens, we render it into a sequence of visual pages $\mathcal{V} = \{v_1, \ldots, v_n\}$, where each v_i denotes one rendered page. This allows the model to reason over a compressed visual representation of the original context:

$$P(\mathcal{R} \mid \mathcal{I}, \mathcal{V}).$$

Each training instance is thus represented as $(\mathcal{I}, \mathcal{V}, \mathcal{R})$.

Parameterized Visual Compression. The rendering process defines how the original text is compressed into visual inputs before being consumed by the model. Each rendering is specified by a configuration vector:

$$\boldsymbol{\theta} = \big(\texttt{dpi}, \texttt{page\_size}, \texttt{font\_family}, \texttt{font\_size}, \texttt{line\_height}, \texttt{alignment}, \texttt{indent}, \texttt{spacing}, \texttt{h\_scale}, \texttt{colors}, \texttt{borders}, \ldots\big),$$

which controls the typography, layout, and visual style of the rendered pages. Given the context $\mathcal{C}$ and configuration $\boldsymbol{\theta}$, the rendering pipeline produces a sequence of visual pages $\mathcal{V}$ that serves as the VLM input.

To quantify compression, we define the compression ratio as

$$\rho(\boldsymbol{\theta}) = \frac{|\mathcal{C}|}{\sum_{i=1}^{n} \tau(v_i)},$$

where $|\mathcal{C}|$ denotes the number of text tokens in the original context and $\tau(v_i)$ is the number of visual tokens consumed by page v_i. A higher $\rho(\boldsymbol{\theta})$ means that more original text is encoded within the same visual-token budget.

In practice, $\boldsymbol{\theta}$ controls both information density (e.g., through font size and DPI) and visual readability (e.g., through layout and spacing). By varying $\boldsymbol{\theta}$, we can continuously adjust the trade-off between compression and readability, which is later optimized for downstream performance.

3.3 Continual Pre-Training on Rendered Text

The goal of continual pre-training is to teach the VLM to read visually compressed long text and reason over it, with rendered pages acting as compact carriers of textual content. We therefore expose it to diverse rendered long-text data and training tasks, enabling long-context capability to transfer from text-token inputs to compressed visual inputs.

Data Construction. To improve robustness under different compression layouts, we render large-scale long-text corpora using diverse configurations. These configurations vary typography, density, and page layout, allowing the model to encounter a broad range of visual realizations of the same underlying text. To avoid degenerate renderings, we impose a set of validity constraints on parameter combinations, excluding cases with poor readability (e.g., line height smaller than font size). We further introduce several manually designed style themes, including *document_style*, *web_style*, *dark_mode*, *code_style*, and *artistic_pixel*, so that the model learns to handle diverse text appearances that are also aligned with visual patterns commonly seen during VLM pre-training.

To make the model robust beyond plain transcription, we construct three families of continual pre-training tasks:

- **OCR Tasks**: the model reconstructs all text on one or multiple rendered pages.
- **Interleaved Language Modeling**: some text spans are rendered as images while the remaining context stays in text, and the model is trained to perform language modeling over such interleaved samples.
- **Generation Tasks**: given partial rendered pages (e.g., the beginning or end of a document), the model predicts the missing continuation.

Through these tasks, the model learns not only

to recognize rendered text, but also to reason and generate under compressed visual contexts.

Loss Function. We optimize the standard cross-entropy objective

$$\mathcal{L}_{\text{CPT}} = -\mathbb{E}_{(\mathcal{I}^*,\mathcal{V},\mathcal{R})} \sum_t \log P_\phi(r_t \mid \mathcal{I}^*, \mathcal{V}, r_{<t}), \tag{1}$$

where r_t denotes the t-th token in the target response $\mathcal{R}$, $r_{<t}$ denotes all preceding response tokens, $\mathcal{I}^*$ is an optional instruction (e.g., absent in interleaved language modeling tasks), and ϕ is initialized from the base VLM. This stage yields Glyph-Base, a model that can reliably read rendered long text and serves as the foundation for subsequent rendering search and post-training.

3.4 LLM-Driven Rendering Search

High compression alone does not guarantee strong downstream performance. The effectiveness of visual-text compression depends critically on the rendering configuration, which determines the trade-off between information density and visual readability. We therefore perform an LLM-driven rendering search after continual pre-training to identify the configuration $\boldsymbol{\theta}^*$ that best balances compression and downstream performance.

Search Procedure. Starting from an initial population of candidate configurations $\{\boldsymbol{\theta}_k\}$ sampled from the pre-training distribution, we iteratively perform the following steps:

1. **Rendering**: render the validation set under each configuration $\boldsymbol{\theta}_k$ to obtain compressed visual inputs.
2. **Evaluation**: run inference with Glyph-Base on the rendered validation set and record both task performance and compression ratio.
3. **LLM-Guided Mutation and Crossover**: use an LLM to analyze the current population and propose promising mutations and crossovers based on past configurations and validation results.
4. **Selection**: update the search history, rank candidate configurations, and sample promising ones for the next generation.

The search continues until convergence, i.e., when no further improvement in performance or compression is observed for a predefined number of generations. The resulting configuration $\boldsymbol{\theta}^*$ is then fixed for post-training. Details of the rendering parameters and the selected configuration are provided in Appendix A.

3.5 Post-Training

With the optimal rendering configuration $\boldsymbol{\theta}^*$ fixed, we further adapt Glyph-Base to downstream long-context tasks under compressed visual inputs. This stage consists of supervised fine-tuning and reinforcement learning, together with an auxiliary OCR alignment objective that stabilizes fine-grained text recognition. While continual pre-training teaches the model to read rendered long text, post-training focuses on improving task-level reasoning under the selected high-compression input configuration.

Supervised Fine-Tuning. We first perform supervised fine-tuning on a high-quality long-context instruction-following corpus, whose contexts are rendered using the selected configuration $\boldsymbol{\theta}^*$. Each target response follows a thinking-style format with explicit reasoning traces (e.g., "`<think>...</think>`"), encouraging the model to perform step-by-step reasoning under compressed visual inputs.

The supervised fine-tuning objective is

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(\mathcal{I},\mathcal{V},\mathcal{R})} \sum_t \log P_\phi(r_t \mid \mathcal{I}, \mathcal{V}, r_{<t}), \tag{2}$$

where r_t denotes the t-th token in the target response $\mathcal{R}$, $r_{<t}$ denotes all preceding response tokens, and ϕ is initialized from Glyph-Base. This stage provides a strong initialization for subsequent reinforcement learning.

Reinforcement Learning. Starting from the SFT model, we further optimize the policy using Group Relative Policy Optimization (GRPO). For each input $(\mathcal{I}, \mathcal{V})$, we sample a group of candidate responses $\{y_1, \ldots, y_G\}$ from the old policy $\pi_{\phi_{\text{old}}}$ and define the importance weight

$$w_i = \frac{\pi_\phi(y_i \mid \mathcal{I}, \mathcal{V})}{\pi_{\phi_{\text{old}}}(y_i \mid \mathcal{I}, \mathcal{V})}. \tag{3}$$

Each sampled response y_i receives a reward $u(y_i) \in \{0, 1\}$ that combines:

- **Verifiable rewards**, obtained from a reference-based LLM judge that scores answer correctness against the ground-truth response.
- **Format rewards**, which encourage the model to follow the required reasoning format.

Model	Single-Doc QA		Multi-Doc QA		Summarization		Few-shot		Synthetic		Code		Avg
	QP	NQA	HQA	2QA	QSUM	GovRep	TREC	TriQA	PR Zh	PR En	RB	LCC	
GPT-4.1	51.60	35.73	69.10	74.15	23.50	33.36	77.00	93.36	100.00	100.00	67.94	68.43	56.03
LLaMA-3.1-8B-Instruct	44.56	26.34	56.88	46.67	**23.28**	**32.36**	19.25	89.12	62.20	99.50	42.81	46.35	41.34
Qwen2.5-7B-Instruct-1M	**45.29**	25.61	60.70	40.51	22.95	29.97	59.37	86.93	98.5	**100.00**	29.80	21.72	42.42
Qwen3-8B	44.67	26.13	65.83	**73.92**	19.60	26.85	70.50	87.98	**100.00**	97.26	40.89	44.87	47.46
GLM-4-9B-Chat-1M	43.75	26.72	58.98	50.89	22.84	27.60	61.50	**90.07**	**100.00**	99.50	55.64	**59.54**	49.27
Glyph	40.64	**28.45**	**66.42**	72.98	19.78	25.53	**82.62**	88.54	89.03	99.50	**60.80**	48.85	**50.56**

Table 1: Performance comparison of Glyph with leading LLMs on LongBench (%). Our model achieves competitive results in the overall average score. Best results are **bolded**, and second-best are underlined. Refer to Table 10 for the rest of the results.

Model	4 Needle						8 Needle					
	0k-8k	8k-16k	16k-32k	32k-64k	64k-128k	Avg	0k-8k	8k-16k	16k-32k	32k-64k	64k-128k	Avg
GPT-4.1	50	38	29	42	38	39.4	33	26	17	22	19	23.4
LLaMA-3.1-8B-Instruct	33.42	25.97	22.73	**26.97**	12.68	24.35	23.80	17.69	**19.85**	17.72	11.79	**18.17**
Qwen2.5-7B-Instruct-1M	25.96	20.13	19.93	24.25	17.29	21.51	17.64	19.48	12.41	14.80	14.24	15.71
Qwen3-8B	29.34	22.67	20.34	23.63	**19.11**	23.02	18.75	19.69	16.81	**17.86**	**15.00**	17.62
GLM-4-9B-Chat-1M	15.17	13.78	9.18	20.27	15.05	14.69	14.55	9.65	9.34	9.47	8.97	10.40
Glyph	**35.44**	**26.82**	**24.15**	25.69	16.37	**25.81**	**25.12**	**21.22**	16.43	13.91	13.51	18.14

Table 2: Performance comparison of our model against leading LLMs on the 4-needle and 8-needle sub-tasks of the MRCR benchmark (%). Our method consistently ranks first or second across most settings while preserving about 3× compression ratio. Performance on the 2-needle task is deferred to the Appendix.

The group-normalized advantage is

$$A_i = \frac{u(y_i) - \text{mean}(\{u(y_j)\}_{j=1}^{G})}{\text{std}(\{u(y_j)\}_{j=1}^{G})}, \quad (4)$$

and the GRPO objective is

$$\mathcal{J}_{\text{GRPO}}(\phi) = \mathbb{E}_{\substack{(\mathcal{I},\mathcal{V})\sim P \\ \{y_i\}_{i=1}^{G}\sim\pi_{\phi_{\text{old}}}}} \Bigg[\frac{1}{G}\sum_{i=1}^{G} \Big(\min\big(w_i A_i, \text{clip}(w_i,\, 1-\epsilon_l,\, 1+\epsilon_h)\, A_i\big) - \beta\, D_{\text{KL}}\big(\pi_\phi \,\|\, \pi_{\text{SFT}}\big)\Big)\Bigg]. \quad (5)$$

where ϵ_l, ϵ_h, and β are hyperparameters.

Auxiliary OCR Alignment. Fine-grained text recognition remains critical under aggressive visual compression. We therefore maintain an auxiliary OCR alignment objective throughout both SFT and RL, encouraging the model to faithfully recover low-level textual details from rendered pages. The OCR task has the same form as in continual pre-training. During RL, its supervision is incorporated through a reward based on Levenshtein distance.

By combining SFT on thinking traces, reinforcement learning, and continuous OCR-aware alignment, post-training improves both high-level reasoning and low-level text fidelity under compressed visual inputs.

4 Experiments

4.1 Experimental Setup

We evaluate Glyph from multiple perspectives to understand the effectiveness and potential of visual-text compression for long-context modeling. Our analysis covers long-context performance, usable-context expansion under a fixed input budget, efficiency gains in training and inference, generalization to real multimodal documents, and behavior under more aggressive compression settings.

Our main evaluation is conducted on LongBench, MRCR, Ruler, and MMLongBench-Doc, comparing against strong text-only LLM baselines of similar size. Additional implementation details, benchmark descriptions, and extended analyses are deferred to Appendix B and Appendix C.

4.2 Main Results on LongBench and MRCR

We first evaluate whether Glyph can preserve strong long-context performance under visual-text compression. Tables 1 and 2 show that Glyph remains competitive with, and in some cases surpasses, strong text-only LLMs of similar size, including Qwen3-8B and GLM-4-9B-Chat-1M. These results suggest that visual-text compression can expand usable context without sacrificing overall task performance.

Figure 3 further illustrates the scaling behavior of Glyph. For LongBench, we report results under

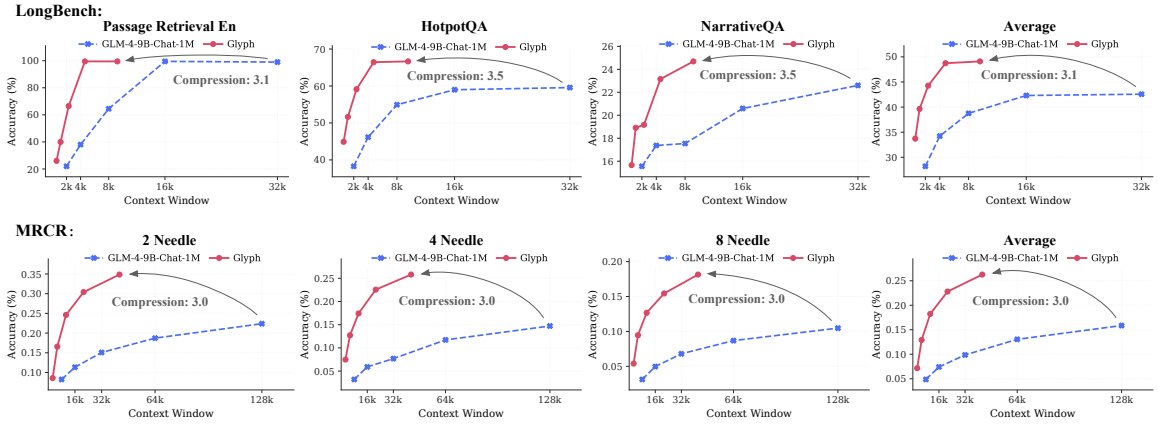


Figure 3: Performance comparison of Glyph and the baseline across different context windows, demonstrating that Glyph achieves performance equivalent to longer contexts with substantially shorter context windows.

Model	Niah-S1	Niah-S2	Niah-M1	Niah-M2	Niah-V	Niah-Q	VT	CWE	FWE	QA-1	QA-2	Avg
GPT-4.1	100.0	98.85	100.0	100.0	99.67	100.0	100.0	97.87	98.66	86.82	77.47	96.30
LLaMA-3.1-8B-Instruct	99.33	99.33	99.33	**99.00**	98.17	99.67	87.07	57.30	81.85	**84.00**	58.00	87.55
Qwen2.5-7B-Instruct-1M	**100.00**	99.67	99.67	**99.00**	93.83	98.75	85.40	72.10	85.67	80.00	60.67	88.61
Qwen3-8B	**100.00**	**100.00**	95.33	84.67	97.42	99.33	98.47	74.67	86.67	70.33	53.33	87.29
GLM-4-9B-Chat-1M	**100.00**	**100.00**	92.67	**99.00**	95.00	**100.00**	98.20	49.50	83.22	72.67	56.67	86.08
DPI: 72 / Compression rate: average 4.0, up to 7.7												
Glyph	73.33	64.67	67.33	56.00	73.42	71.42	77.93	94.40	92.67	59.33	63.33	72.17
DPI: 96 / Compression rate: average 2.2, up to 4.4												
Glyph	98.00	95.33	95.67	85.00	96.33	95.83	94.93	94.80	98.00	79.00	70.67	91.23
DPI: 120 / Compression rate: average 1.2, up to 2.8												
Glyph	99.67	99.00	**100.00**	93.67	**99.00**	99.58	**99.33**	**98.97**	**99.11**	79.00	**74.00**	**94.67**

Table 3: Performance on the Ruler benchmark (%). We demonstrate the impact of different DPI settings on our model's performance and the resulting compression ratios. For each configuration, the table includes both the average compression ratio across all sub-tasks and the maximum compression achieved for specific sub-task types.

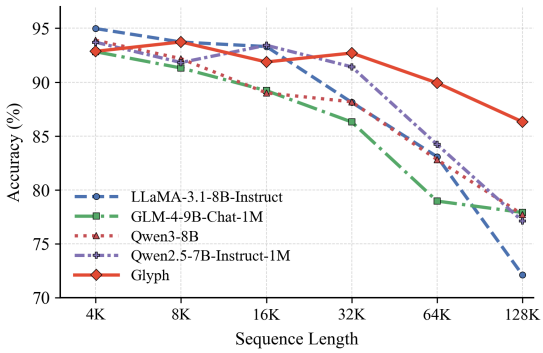


Figure 4: Model performance degradation across different sequence lengths on the Ruler benchmark.

truncated contexts; for MRCR, we use the original dataset splits. On LongBench, Glyph achieves an average effective compression ratio of 3.3×, with some tasks reaching about 5×, while on MRCR the average compression ratio is 3.0×. As a result, under the same model input budget, Glyph can encode and utilize several times more original text than text-only baselines. This advantage further grows as the input budget increases.

4.3 Results on Ruler

On the Ruler benchmark, Glyph also achieves performance comparable to leading LLMs across most categories (Table 3). We exclude the UUID task because it remains particularly challenging for current VLMs, likely due to the difficulty of faithfully recognizing rare and order-sensitive character sequences under visual encoding.

Beyond the standard benchmark setting, Ruler also reveals an important property of our approach: its performance can be improved through test-time scaling on the visual side. When we increase the rendering resolution (DPI) at inference time, Glyph shows consistent gains, and at higher DPI settings it even surpasses strong text-only baselines. This suggests that the performance of visual-text compression still has meaningful headroom, and that better visual fidelity can be directly translated into stronger long-context performance.

We further analyze performance across different sequence lengths in Figure 4. In shorter contexts,

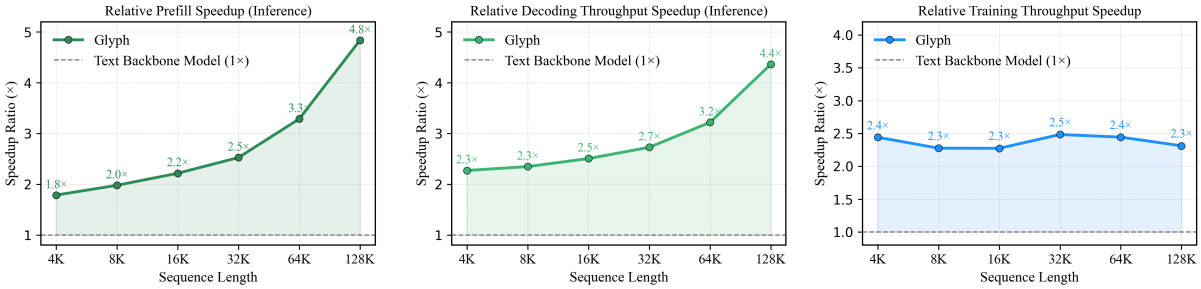


Figure 5: Speedup ratios of Glyph over the text backbone model for prefill, decoding, and training across different sequence lengths.

Model	SP	CP	UA	Acc	F1
GLM-4.1V-9B-Base	36.76	23.41	21.52	29.18	28.78
Glyph-Base	47.91	22.24	14.80	32.48	34.44
Glyph	**57.73**	**39.75**	**27.80**	**45.57**	**46.32**

Table 4: Results on MMLongBench-Doc (%). SP, CP, UA, and Acc denote Single-page, Cross-page, Unanswerable, and Overall Accuracy, respectively.

Configuration	LongBench	MRCR	Ruler	Avg.
Random Config	41.78	15.82	65.13	40.91
Manual Config	**43.45**	19.33	68.09	43.62
Search-based Config	**43.45**	**22.10**	**71.24**	**45.60**

Table 5: Ablation study comparing randomly combined, manually designed, and search-based configurations on three benchmarks under SFT setting. The search-based configuration achieves the best overall performance.

Model	LongBench	MRCR	Ruler
Glyph	50.56	26.27	72.17
– w/o OCR (in RL)	-1.40	-2.00	-0.35
– w/o RL	-7.11	-4.17	-0.93
– w/o OCR (in SFT)	-8.12	-8.42	-1.23

Table 6: Ablation study showing the performance drop (%) relative to the final Glyph model when components are progressively removed.

Model	2 Needle	4 Needle	8 Needle
GLM-4-9B-Chat-1M	10.08	6.19	2.26
Qwen2.5-7B-Instruct-1M	**11.36**	7.34	**7.77**
Glyph	9.36	**7.62**	7.64

Table 7: Average MRCR performance (%) across 128K–1M context lengths under different needle counts.

text-only models such as LLaMA-3.1-8B-Instruct maintain a slight edge. However, as the input length grows, Glyph degrades more slowly. This is consistent with our observations on LongBench and MRCR: due to compression, the same amount of original text requires fewer model input tokens for Glyph. Therefore, its accuracy remains more stable as context grows.

4.4 Efficiency Evaluation

We further evaluate the efficiency of Glyph in both training and inference by comparing it with the text backbone model. As shown in Figure 5, Glyph consistently provides clear speedups in prefilling, decoding, and SFT training, with the gains becoming larger at longer sequence lengths.

These improvements come from the reduced number of model input tokens under visual-text compression. Although the visual pipeline introduces additional computation from the vision encoder, this overhead is small relative to the 9B language backbone. In return, the shorter effective inputs reduce attention cost and KV-cache usage, leading to clear efficiency gains in both inference and training.

As the sequence length grows from 8k to 128k, Glyph exhibits markedly better scalability, achieving stable SFT training speedup together with increasing inference speedup. Detailed evaluation settings are provided in Appendix B.

4.5 Cross-Modal Generalization

Although Glyph is trained primarily on rendered text images rather than natural multimodal documents, it still transfers effectively to real-world document understanding. To evaluate this generalization ability, we test Glyph on MMLongBench-Doc, which contains 130 long PDF documents with diverse layouts and embedded images. As shown in Table 4, Glyph achieves clear improvements over the backbone model GLM-4.1V-9B-Base.

These results suggest that training on visually compressed text does not merely improve performance on synthetic rendered inputs. Instead, it also strengthens the model's ability to process real

long-form documents, indicating that the learned representation transfers beyond the training distribution.

4.6 Ablation Study and Analysis

We conduct a series of ablations and further studies to better understand Glyph. These results clarify which design choices matter most for performance and how much headroom remains in the visual-text compression paradigm. Additional comparisons, robustness evaluations, and computational trade-off analyses are provided in Appendix C.

Configuration Search. We compare three types of rendering configurations for SFT: (i) randomly sampled configurations from the pre-training set, (ii) manually designed settings based on prior knowledge, and (iii) the configuration obtained from our search procedure. While these settings achieve similar compression ratios, Table 5 shows that the searched configuration consistently performs best on average and across most individual tasks. This result highlights the importance of systematically optimizing the rendering strategy, rather than relying on manual design alone.

OCR Auxiliary Tasks. We further study the effect of adding OCR auxiliary tasks during both SFT and RL training. As shown in Table 6, including OCR supervision yields consistent gains across benchmarks. This suggests that improving low-level text recognition also strengthens the model's higher-level long-context understanding under compressed visual inputs.

Extreme Compression Exploration. To examine the limits of our approach, we further explore a more aggressive setting with an effective compression ratio of $8\times$. We apply this configuration during post-training and evaluate the resulting model on MRCR with sequence lengths extended from 128k to 1024k. As shown in Table 7, Glyph remains on par with GLM-4-9B-Chat-1M and Qwen2.5-1M under this setting. This result suggests that visual-text compression can be pushed to more extreme regimes while still retaining non-trivial long-context capability.

5 Conclusion

In this work, we present Glyph, a long-context modeling framework that renders long texts into compact images and processes them with vision-language models. Through continual pre-training on rendered text, LLM-driven rendering search, and targeted post-training, Glyph achieves 3–4$\times$ token compression while maintaining competitive performance with strong text-only LLMs of similar size, such as Qwen3-8B. Extensive experiments further show clear efficiency gains in both training and inference, demonstrating that visual-text compression can serve as an effective representation for long-context language modeling. Taken together, these findings suggest that redesigning the input representation offers a promising new paradigm for scaling long-context modeling, complementary to existing token-based approaches.

6 Limitations and Future Directions

Our results suggest that visual-text compression is a promising representation strategy for long-context language modeling, but the current study still has several important limitations.

Sensitivity to rendering choices. Glyph relies on rendering long text into images before processing, and its performance remains sensitive to rendering choices such as resolution, font, and spacing. While our search procedure identifies configurations that perform well in practice, improving robustness across a broader range of rendering styles remains an important challenge.

OCR and fine-grained text fidelity. As reflected in the Ruler benchmark, rare and order-sensitive strings such as UUIDs remain challenging for current VLMs. Errors in these cases often take the form of character substitutions or reordering, pointing to limitations in fine-grained text recognition under aggressive compression. While such failures do not dominate most tasks in this paper, they likely bound the current upper limit of visual-text compression for language modeling.

Task scope. Our evaluation mainly focuses on long-context understanding benchmarks. These tasks provide a strong proof of concept, but they do not fully capture broader settings such as reasoning-heavy or agentic workflows, where models may need to selectively revisit earlier context, inspect local details, or combine compressed context with tool use. We also observe that, compared with text-only models, the visual-text model generalizes less stably across different tasks.

Future directions. A natural next step is to further explore the compression potential of visual rep-

resentations for general language modeling. More broadly, this line of work suggests that scaling language models may benefit not only from processing more tokens, but also from advances in how the original input is represented and compressed. One important question along this direction is whether representations beyond page-level rendering, such as higher-dimensional or more continuous latent representations, can preserve more information under the same budget while remaining useful for downstream tasks. Another promising direction is to combine compressed context with adaptive tool-use mechanisms, so that a model or agent can first reason over a compressed global view and then selectively zoom in, retrieve, or revisit local regions when finer-grained evidence is needed.

7 Acknowledgments

This work was supported by the National Science Foundation for Distinguished Young Scholars (with No. 62125604). This work was also supported by the Natural Science Foundation of China (No. 62536008). We would also like to thank Zhipu AI for sponsoring GPU computing and API costs consumed in this study.

References

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*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. In *Proceedings of ACL*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, and 1 others. 2025a. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*.

Xi Chen, Xiao Wang, Soravit Changpinyo, Anthony J Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, and 1 others. 2022. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2024. Longlora: Efficient fine-tuning of long-context large language models. In *The International Conference on Learning Representations (ICLR)*.

Yutao Chen, Yiren Wang, and 1 others. 2025b. Cope: Complex positional encoding for long context extrapolation. *arXiv preprint arXiv:2508.18308*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, and 38 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, and 1 others. 2024a. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and 1 others. 2024b. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290.

Yunpeng Huang, Jingwei Xu, Junyu Lai, Zixu Jiang, Taolue Chen, Zenan Li, Yuan Yao, Xiaoxing Ma, Lijuan Yang, Hao Chen, and 1 others. 2023. Advancing transformer architecture in long-context large language models: A comprehensive survey. *arXiv preprint arXiv:2311.12351*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Philippe Laban, Alexander Richard Fabbri, Caiming Xiong, and Chien-Sheng Wu. 2024. Summary of a haystack: A challenge to long-context llms and rag systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9885–9903.

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

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

Yang Liu, Wei Gao, and 1 others. 2024b. Long context is not long at all: A prospector of long-dependency data for large language models. *arXiv preprint arXiv:2407.11234*.

Baolin Peng, Zhuohan Li, and 1 others. 2023. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, and 1 others. 2025. Rwkv-7" goose" with expressive dynamic state evolution. *arXiv preprint arXiv:2503.14456*.

Ofir Press, Noah A. Smith, and Mike Levy. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Zhen Sun, Peng Cheng, Wei He, and 1 others. 2022. Xpos: Improving position interpolation with extrapolation. *arXiv preprint arXiv:2212.10554*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Kiran Vodrahalli, Santiago Ontanon, Nilesh Tripuraneni, Kelvin Xu, Sanil Jain, Rakesh Shivanna, Jeffrey Hui, Nishanth Dikkala, Mehran Kazemi, Bahare Fatemi, and 1 others. 2024. Michelangelo: Long context evaluations beyond haystacks via latent structure queries. *arXiv preprint arXiv:2409.12640*.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, and 1 others. 2024a. Cogvlm: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*, 37:121475–121499.

Yizhi Wang, Fan Yang, and 1 others. 2024b. Longrecipe: Recipe for efficient long context generalization in large language models. *arXiv preprint arXiv:2406.12345*.

Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.

Yingsheng Wu, Yuxuan Gu, Xiaocheng Feng, Weihong Zhong, Dongliang Xu, Qing Yang, Hongtao Liu, and Bing Qin. 2024. Extending context window of large language models from a distributional perspective. *arXiv preprint arXiv:2410.01490*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. 2024. Gated linear attention transformers with hardware-efficient training. In *International Conference on Machine Learning*, pages 56501–56523. PMLR.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL*.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and 1 others. 2025a. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025b. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Tianle Zhang, Zhuohan Li, and 1 others. 2024. Longalign: Instruction-tuning long-context llms. *arXiv preprint arXiv:2401.10968*.

Wei Zhu, Ziheng Wang, and 1 others. 2024. Data-adaptive positional encoding for length generalization. *NeurIPS*.

A Rendering Parameters

Our rendering parameters are summarized in Table 8. The best configuration found by our LLM-driven rendering search is shown in Figure 6, together with its rendered output.

Factor	Specification / Sampling Strategy
dpi	Mixture of sets: *lowest* (45–59), *low* (60–71), *medium* (72–119), *normal* ({72,80,96,100,110,120,144,150,300}), *high* (over 300); favor normal/medium with small probability spikes to extremes.
page_size	(i) Fixed paper sizes (A4, Letter, Legal, A5, B5, A3, B4, Tabloid) with priors; (ii) common aspect ratios (e.g., 1.414, 1.333, 1.5, 1.778); (iii) fully random aspect via piecewise distribution (narrow → tall).
font_family	Pooled and deduplicated families across serif/sans/mono/pixel; italics sampled by filename heuristics (suffixes, italic/oblique).
font_size	$\{7, 7.5, 8, 9, 9.5, 10, 11, 12, 14\}$; line_height tied as font_size + $\{0, \ldots, 3\}$.
alignment	LEFT/JUSTIFY (dominant) with small-prob. RIGHT/CENTER.
margins	Three patterns: all-equal; vertical-larger; horizontal-larger; values in 10–40pt ranges.
indent	Modes: none; first-line indent (≈1–2.5 em); block/hanging with left/right indents.
spacing	space-before/space-after use a multi-mode prior (none, small, large).
h_scale	Horizontal glyph scaling (0.75–1.0) with decaying probabilities.
colors	Page/background/font palettes for light/dark themes; document/web/-code styles inherit coherent triplets (page, paragraph, font).
borders	Optional paragraph borders with width/padding; disabled by default.
newline_markup	With small probability, explicit markers (e.g., \n, tags, or tokens) inserted to preserve structure.
auto_crop	Optional white-margin cropping and last-page trimming.

Table 8: Controllable factors in the rendering pipeline and their sampling strategies. The mixture design yields broad yet realistic typography/layout coverage and tunable compression $\rho(\boldsymbol{\theta})$.

Model	2 Needle					
	0k-8k	8k-16k	16k-32k	32k-64k	64k-128k	Avg
GPT-4.1	83	72	67	62	59	68.6
LLaMA-3.1-8B-Instruct	54.27	**53.21**	**51.05**	29.81	24.98	42.66
Qwen3-8B	**58.95**	41.18	36.18	24.99	20.89	36.44
GLM-4-9B-Chat-1M	39.77	15.87	18.42	18.63	18.42	22.22
Qwen2.5-7B-Instruct-1M	45.92	51.07	46.97	**34.67**	**37.57**	**43.24**
Glyph	41.51	40.78	39.58	29.67	22.41	34.85

Table 9: Performance of various models on the MRCR task (%) with the 2 Needle setting across different context length intervals (0k–8k, 8k–16k, 16k–32k, 32k–64k, 64k–128k) and the average score.

B Implementation Details

Training Details For continual pre-training of the 9B long-context backbone, the model is initialized from the released GLM-4.1V-9B-Base checkpoint and trained on a diverse mixture of rendered long-context data and vision-language corpora (e.g., OCR task) within 128k context length. The training uses a global batch size of 170 and a learning rate of 2e-6 with cosine decay for around 4000 steps.

For rendering search, we run 5 rounds with 200 steps each to identify the configuration that maximizes compression while maintaining strong performance.

After this, we conduct further SFT and RL training. For SFT, we train for 1.5k steps with a batch size of 32. The Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.95$) is used with cosine decay and 160 warm-up steps, where the learning rate decays from 5e-6 to 2e-6. For reinforcement learning, we adopt the GRPO algorithm. Each training group samples 16 candidate responses, and degenerate samples with all-zero or all-one rewards are discarded. We apply the clip-higher trick from DAPO (Yu et al., 2025b) with ϵ_l being 0.2 and ϵ_h being 0.28. Training runs for 500 iterations with a batch size of 32. We also use the Adam optimizer with a constant learning rate of 1e-6.

Baselines. We compare Glyph with leading open-sourced LLMs of similar size:

- **Qwen3-8B** achieves state-of-the-art performance across reasoning and a wide range of tasks.
- **Qwen2.5-7B-Instruct-1M** excels at long-context understanding and achieves strong performance across diverse benchmarks.
- **LLaMA-3.1-8B-Instruct** is a widely used model with strong instruction-following and multilingual capabilities.
- **GLM-4-9B-Chat-1M** delivers strong long-context performance across a broad range of tasks.

Backbone Model. Our method relies on a strong VLM to process long-context tasks. Given the strong performance of GLM-4.1V-9B, especially on OCR and long-document tasks, we choose GLM-4.1V-9B-Base as our backbone model.

Model	Single-Doc QA		Multi-Doc QA		Summarization		Few-shot		Synthetic
	QA Zh	QA En	Mus	Dur	News	VcSum	Sam	Lsht	Pa C
GPT-4.1	63.90	51.27	55.63	24.58	23.70	14.66	41.25	50.00	26.5
LLaMA-3.1-8B-Instruct	62.20	**54.98**	31.61	**33.75**	**24.21**	**16.23**	7.61	0.00	7.13
Qwen3-8B	60.98	49.78	45.54	16.69	18.55	12.08	36.47	42.00	12.81
GLM-4-9B-Chat-1M	**63.17**	52.88	39.14	28.27	23.90	16.21	36.15	**47.38**	2.39
Qwen2.5-7B-Instruct-1M	62.98	53.62	34.72	21.85	21.02	12.20	**39.17**	28.68	3.50
Glyph	37.23	45.89	**56.18**	26.87	21.52	12.43	32.49	44.43	**30.50**

Table 10: The rest of the results on LongBench benchmark (%), which encompasses Single-Document QA, Multi-Document QA, Summarization, Few-shot Learning, and Synthetic task.

Evaluation Benchmarks. To conduct a comprehensive analysis of the long-context performance, we have adopted three popular benchmarks, including LongBench, MRCR, and Ruler. LongBench consists of 21 datasets in total in 6 categories, covering diverse long-context tasks. MRCR is a task proposed by Vodrahalli et al. (2024). We use the OpenAI version, which consists of multi-turn conversations about writing, asking models to recall one of the contexts in dialogue history. Ruler is a widely used synthetic benchmark with 11 NIAH tasks. To validate cross-modal benefits, we choose the MMLongBench-Doc, which involves 130 lengthy PDFs with diverse layouts and images, and 1062 questions.

Efficiency Evaluation Setting. For training, we focus on SFT since RL involves rollout time, making it difficult to compare fairly. Moreover, running RL at very long context lengths (e.g., 64k or beyond) requires excessive memory resources. This again highlights the advantage of our approach: through compression, we can conduct RL training at 32k context length while effectively covering over 100k tokens of raw text input, where RL is prohibitively difficult for LLMs due to memory and computation demands. For SFT, we measure per-sample training time under the same number of training data using 8×80G H100 GPUs. For inference, we deploy both models on a single 80G H100 and measure efficiency along two axes: (i) prefill latency at batch size 1, and (ii) per-sample inference time at the maximum feasible batch size with output length set to 256 tokens. We omit separate KV-cache measurements because KV-cache usage scales linearly with sequence length, so compression translates almost directly into about 67% memory savings.

Models	Average
Qwen3-8B (YaRN)	47.46
- Selective Context	42.41
- RAG	37.17
Glyph	50.56

Table 11: Comparison with additional long-context baselines on LongBench. Retrieval- and compression-based methods are evaluated under similar compression ratios.

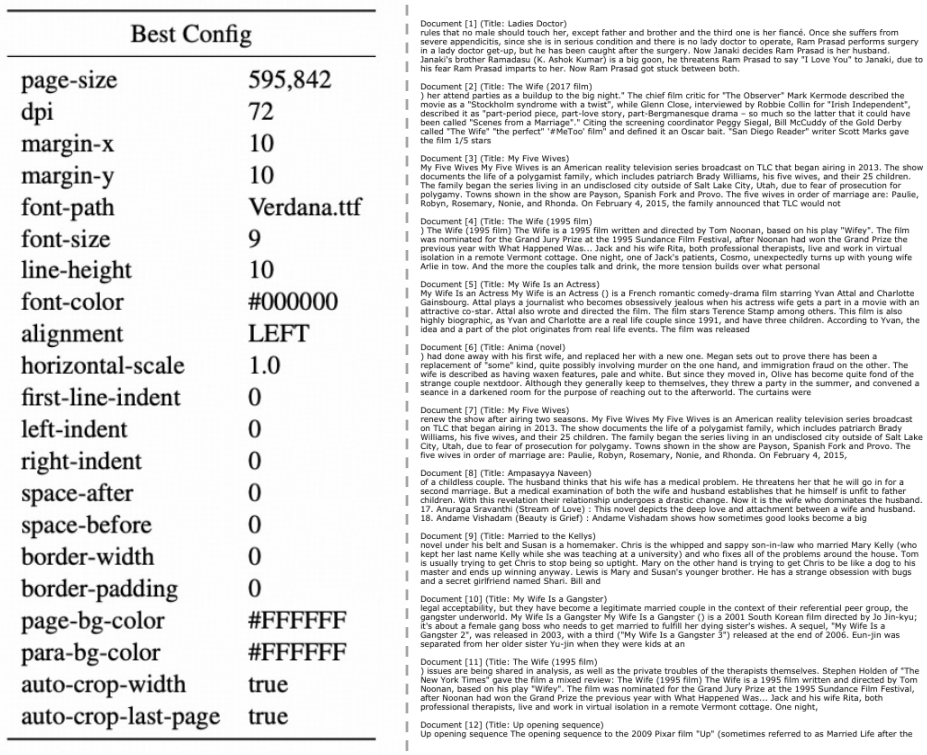
C Additional Comparisons and Analyses

Comparison with Additional Baselines. To compare Glyph with commonly used alternatives for long context modeling, we additionally evaluate retrieval- and compression-based methods on LongBench under similar compression ratios. We note that Qwen3-8B, our main text baseline, already incorporates YaRN extrapolation for long-context extension. As shown in Table 11, Glyph outperforms these baselines, providing further evidence for the effectiveness of visual-text compression under similar compression ratios.

Comparison with a Matched Text-Only Baseline. We also include a matched text-only baseline obtained by applying the same post-training procedure from Glyph-Base in a text-only setting. As shown in Table 12, Glyph achieves comparable performance on LongBench, MRCR, and Ruler while using 3–4× fewer input tokens. Although this comparison does not fully align the continual pre-training stage between the text-only and visual-text settings, it still provides additional evidence that the benefit of Glyph is not solely attributable to post-training recipe differences.

Computational Trade-offs of the Visual Pipeline. Visual-text compression introduces additional computation from the vision encoder, but this overhead is small relative to the language backbone in Glyph

Best Config	
page-size	595,842
dpi	72
margin-x	10
margin-y	10
font-path	Verdana.ttf
font-size	9
line-height	10
font-color	#000000
alignment	LEFT
horizontal-scale	1.0
first-line-indent	0
left-indent	0
right-indent	0
space-after	0
space-before	0
border-width	0
border-padding	0
page-bg-color	#FFFFFF
para-bg-color	#FFFFFF
auto-crop-width	true
auto-crop-last-page	true



Document [1] (Title: Ladies Doctor)
rules that no male should touch her, except father and brother and the third one is her fiancé. Once she suffers from severe appendicitis, since she is in serious condition and there is no lady doctor to operate, Ram Prasad performs surgery in a lady doctor get-up, but he has been caught after the surgery. Now Janaki decides Ram Prasad is her husband. Janaki's brother Ramadasu (K. Ashok Kumar) is a big goon, he threatens Ram Prasad to say "I Love You" to Janaki, due to his fear Ram Prasad imparts to her. Now Ram Prasad got stuck between both.

Document [2] (Title: The Wife (2017 film))
) her attend parties as a buildup to the big night." The chief film critic for "The Observer" Mark Kermode described the movie as a "Stockholm syndrome with a twist", while Glenn Close, interviewed by Robbie Collin for "Irish Independent", described it as "part-period piece, part-love story, part-Bergmanesque drama – so much so the latter that it could have been called "Scenes from a Marriage"." Citing the screening coordinator Peggy Siegal, Bill McCuddy of the Gold Derby called "The Wife" "the perfect" '#MeToo' film" and defined it an Oscar bait. "San Diego Reader" writer Scott Marks gave the film 1/5 stars

Document [3] (Title: My Five Wives)
My Five Wives My Five Wives is an American reality television series broadcast on TLC that began airing in 2013. The show documents the life of a polygamist family, which includes patriarch Brady Williams, his five wives, and their 25 children. The family began the series living in an undisclosed city outside of Salt Lake City, Utah, due to fear of prosecution for polygamy. Towns shown in the show are Payson, Spanish Fork and Provo. The five wives in order of marriage are: Paulie, Robyn, Rosemary, Nonie, and Rhonda. On February 4, 2015, the family announced that TLC would not

Document [4] (Title: The Wife (1995 film))
) The Wife (1995 film) The Wife is a 1995 film written and directed by Tom Noonan, based on his play "Wifey". The film was nominated for the Grand Jury Prize at the 1995 Sundance Film Festival, after Noonan had won the Grand Prize the previous year with What Happened Was... Jack and his wife Rita, both professional therapists, live and work in virtual isolation in a remote Vermont cottage. One night, one of Jack's patients, Cosmo, unexpectedly turns up with young wife Arlie in tow. And the more the couples talk and drink, the more tension builds over what personal

Document [5] (Title: My Wife Is an Actress)
My Wife Is an Actress My Wife is an Actress () is a French romantic comedy-drama film starring Yvan Attal and Charlotte Gainsbourg. Attal plays a journalist who becomes obsessively jealous when his actress wife gets a part in a movie with an attractive co-star. Attal also wrote and directed the film. The film stars Terence Stamp among others. This film is also highly biographic, as Yvan and Charlotte are a real life couple since 1991, and have three children. According to Yvan, the idea and a part of the plot originates from real life events. The film was released

Document [6] (Title: Anima (novel))
) had done away with his first wife, and replaced her with a new one. Megan sets out to prove there has been a replacement of "some" kind, quite possibly involving murder on the one hand, and immigration fraud on the other. The wife is described as having waxen features, pale and white. But since they moved in, Olive has become quite fond of the strange couple nextdoor. Although they generally keep to themselves, they threw a party in the summer, and convened a seance in a darkened room for the purpose of reaching out to the afterworld. The curtains were

Document [7] (Title: My Five Wives)
renew the show after airing two seasons. My Five Wives My Five Wives is an American reality television series broadcast on TLC that began airing in 2013. The show documents the life of a polygamist family, which includes patriarch Brady Williams, his five wives, and their 25 children. The family began the series living in an undisclosed city outside of Salt Lake City, Utah, due to fear of prosecution for polygamy. Towns shown in the show are Payson, Spanish Fork and Provo. The five wives in order of marriage are: Paulie, Robyn, Rosemary, Nonie, and Rhonda. On February 4, 2015,

Document [8] (Title: Ampasayya Naveen)
of a childless couple. The husband thinks that his wife has a medical problem. He threatens her that he will go in for a second marriage. But a medical examination of both the wife and husband establishes that he himself is unfit to father children. With this revelation their relationship undergoes a drastic change. Now it is the wife who dominates the husband. 17. Anuraga Sravanthi (Stream of Love) : This novel depicts the deep love and attachment between a wife and husband. 18. Andame Vishadam (Beauty is Grief) : Andame Vishadam shows how sometimes good looks become a big

Document [9] (Title: Married to the Kellys)
novel under his belt and Susan is a homemaker. Chris is the whipped and sappy son-in-law who married Mary Kelly (who kept her last name Kelly while she was teaching at a university) and who fixes all of the problems around the house. Tom is usually trying to get Chris to stop being so uptight. Mary on the other hand is trying to get Chris to be like a dog to his master and ends up winning anyway. Lewis is Mary and Susan's younger brother. He has a strange obsession with bugs and a secret girlfriend named Shari. Bill and

Document [10] (Title: My Wife Is a Gangster)
legal acceptability, but they have become a legitimate married couple in the context of their referential peer group, the gangster underworld. My Wife Is a Gangster My Wife Is a Gangster () is a 2001 South Korean film directed by Jo Jin-kyu; it's about a female gang boss who needs to get married to fulfill her dying sister's wishes. A sequel, "My Wife Is a Gangster 2", was released in 2003, with a third ("My Wife Is a Gangster 3") released at the end of 2006. Eun-jin was separated from her older sister Yu-jin when they were kids at an

Document [11] (Title: The Wife (1995 film))
) issues are being shared in analysis, as well as the private troubles of the therapists themselves. Stephen Holden of "The New York Times" gave the film a mixed review: The Wife (1995 film) The Wife is a 1995 film written and directed by Tom Noonan, based on his play "Wifey". The film was nominated for the Grand Jury Prize at the 1995 Sundance Film Festival, after Noonan had won the Grand Prize the previous year with What Happened Was... Jack and his wife Rita, both professional therapists, live and work in virtual isolation in a remote Vermont cottage. One night,

Document [12] (Title: Up opening sequence)
Up opening sequence The opening sequence to the 2009 Pixar film "Up" (sometimes referred to as Married Life after the

Figure 6: The optimal parameter setting. The left column lists the values for page layout, font, and spacing, while the right column provides an example of the rendered text.

Models	LongBench	MRCR	Ruler
Text Baseline	50.48	27.67	92.89
Glyph	50.56	26.27	91.23

Table 12: Comparison with a matched text-only baseline trained with the same recipe as Glyph.

Models	Indic-BN	Indic-HI	Indic-OR	Arabic
Qwen3-8B	64.15	68.96	65.73	64.71
GLM-4-9B-Chat-1M	44.70	47.50	42.10	58.01
Glyph	59.30	61.10	54.40	60.29

Table 13: Robustness across scripts on Indic and Arabic MMLU.

(0.6B vs. 9B parameters). In return, the reduced number of model input tokens lowers attention cost and KV-cache usage. Consequently, as shown in Figure 5, Glyph achieves more than 4× inference speedup and more than 2× supervised fine-tuning speedup at 128K context length.

Robustness Across Scripts and Rendering Variations. We further evaluate Glyph on Indic and Arabic MMLU to examine its robustness on languages and scripts beyond the primary training distribution. For each setting, we use rendering search to select suitable fonts and rendering configurations. As shown in Table 13, Glyph maintains stable performance across these settings, suggesting that the visual-text compression pipeline can generalize beyond the scripts seen during training.