Training Long-Context Vision-Language Models Effectively with Generalization Beyond 128K Context

Zhaowei Wang[1,2,*] **Lishu Luo**[2] **Haodong Duan**[2] **Weiwei Liu**[2] **Sijin Wu**[2]
Ji Luo[2] **Shen Yan**[2] **Shuai Peng**[2] **Sihang Yuan**[2] **Chaoyi Huang**[2]
Yi Lin[2,†] **Yangqiu Song**[1,†]
[1]CSE Department, HKUST [2]ByteDance Seed
{zwanggy, yqsong}@cse.ust.hk linyi.james@bytedance.com

Abstract

Long-context modeling is becoming a core capability of modern large vision-language models (LVLMs), enabling sustained context management across long-document understanding, video analysis, and multi-turn tool use in agentic workflows. Yet practical training recipes remain insufficiently explored, particularly for designing and balancing long-context data mixtures. In this work, we present a systematic study of *long-context continued pre-training* (LongPT) for LVLMs, extending a 7B model from 32K to 128K context with extensive ablations on long-document data. Our study yields three key findings: i) synthesizing long-document VQA data provides effective and diverse long-context supervision, covering tasks from information extraction to numerical reasoning; ii) retrieving relevant evidence remains the primary long-context bottleneck, favoring retrieval-heavy mixtures with a small amount of reasoning data to preserve task diversity; iii) surprisingly, pure long-document VQA data largely preserves short-context capabilities, suggesting that instruction-formatted long data lessens the need for short-data mixing. Based on these findings, we introduce **MMProLong**[1], obtained by long-context continued pre-training from Qwen2.5-VL-7B with only a 5B-token budget. MMProLong improves long-document VQA performance by 7.11% and generalizes beyond its 128K training window, maintaining strong performance at 256K and 512K context lengths without additional training. It further transfers to webpage-based multimodal needle-in-a-haystack tasks, long-context vision-text compression, and long-video understanding without task-specific supervision. Overall, our study provides a practical LongPT recipe and an empirical foundation for advancing the next generation of LVLMs.

1 Introduction

The ability to process extremely long context has unlocked a wide range of new capabilities for both large language models [LLMs; 1, 2] and large vision-language models [LVLMs; 3, 4]. For LVLMs in particular, long-context modeling enables multi-hop reasoning over large document collections [5, 6], capturing spatiotemporal dependencies from hour-long videos [7, 8], and maintaining context consistency in long-horizon agent tasks [9–11].

To support such capabilities, LVLMs' context windows have been rapidly scaled to 128K tokens and beyond, driven by both proprietary models (e.g., Gemini 3.1 [12] and GPT-5.4 [13]) and open-weight alternatives such as Qwen3-VL [3] and GLM-4.5V [14]. However, while recent technical reports [3, 14] discuss the use of long-document data, practical recipes for developing long-context

[*]Work done at ByteDance Seed. [†]Corresponding authors.
[1]Code: https://github.com/zhaowei-wang-nlp/MMProLong

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

vision-language models (LCVLMs) remain underexplored. It remains unclear which types of long-context data should be synthesized, how diverse long-context data should be mixed with short-context data, and how training choices such as length distributions affect the resulting model.

To bridge this gap, we present a systematic study of key design choices for *long-context continued pre-training* (LongPT) in LVLMs. Starting from Qwen2.5-VL-7B [15], we extend its context window from 32K to 128K and analyze how multimodal long-context training data should be constructed and mixed. We begin with long documents, which provide realistic, information-rich multimodal contexts with rich visual layouts and dense textual content. From these documents, we construct five training tasks spanning two supervision families: long-document VQA and OCR transcription. Comparing these tasks, we find that long-document VQA is substantially more effective than OCR transcription, as it supports a broader task spectrum from information extraction to complex numerical reasoning. Practically, high-quality QA pairs can be efficiently generated from only a few semantically coherent pages, avoiding full-document processing by the teacher model.

We then study practical data recipes for LongPT in LVLMs, covering sequence-length distribution, long-context task mixtures, and the role of short-context data. In these ablations, we observe three main findings: i) long-context ability is not acquired only at a specific target length such as 128K; instead, balanced sequence lengths help models learn generalizable key-information retrieval; ii) evidence retrieval is the primary bottleneck in long-context training, favoring retrieval-heavy mixtures with a small amount of reasoning data to preserve task diversity; and iii) contrary to common practice in long-context continued training for LLMs [16], pure long-document VQA data largely preserves short-context capabilities, suggesting that instruction-formatted long data reduces the need for short-context data. Taken together, these ablations yield our final LongPT recipe: balanced sequence lengths, a retrieval-heavy long-document VQA mixture, and no explicit short-context data mixing.

With this recipe, the resulting model **MMProLong** not only improves long-document VQA performance by 7.1% at 64K/128K lengths, but also maintains strong performance at 256K and 512K context lengths without additional training or adaptation, exceeding baselines by over 20%. Beyond this, our model generalizes to broader multimodal long-context scenarios, including webpage-based needle-in-a-haystack on MM-NIAH [6], long-context compression on VTCBench [17], and long-video understanding tasks [7, 8, 18]. Additionally, experiments on Qwen3-VL show similar improvements, suggesting that our training recipe provides a practical path toward building general long-context capability in LVLMs.

2 Related Work

Extending the context windows for LLMs. Extending context windows has become essential for processing long-context inputs, with recent LLMs supporting 128K tokens [1, 19] and even 1M tokens [20–22]. Existing approaches either enable lightweight extension through positional extrapolation [23–27] and attention modifications [28–32], or rely on long-context continued training to build more robust long-context capability [16, 33–35]. Our work follows the data-centric continued-training perspective, but studies it in multimodal settings where long contexts contain interleaved image and text tokens.

Long-context vision-language models. As LLM context windows have expanded, recent LVLMs such as Gemini 3.1 Pro [12], Claude Opus 4.7 [36], and Qwen3-VL [3] have also supported substantially longer contexts. However, their training recipes remain largely undisclosed, making it difficult to identify which design choices drive long-context capability. Concurrent work [37] studies long-document data construction for LVLMs, but mainly builds on backbones that already support 128K or longer contexts, such as Qwen3-VL [3] and Mistral 3.1 [38]; thus, its findings may reflect context alignment rather than true context extension. For example, they find 1B-token LongPT outperforming 10B-token LongPT, and LongSFT outperforming LongPT. In contrast, we study LongPT on Qwen2.5-VL [15], whose native context window is only 32K, allowing us to directly examine how to extend LVLMs to longer contexts. Another related line of work studies long-video understanding [39–45], but these methods are often specialized for temporal redundancy and video token reduction rather than general long-context LVLM training.

Multimodal long-context evaluation. Recent benchmarks evaluate multimodal long-context understanding from complementary perspectives, including long-document VQA [46, 47], multimodal

needle-in-a-haystack [48, 49], vision-text compression [17], and long-video understanding [8, 18, 50]. MMLongBench [5] provides a comprehensive evaluation across five task categories with standardized context lengths up to 128K. Our evaluation covers MMLongBench, VTCBench, and long-video benchmarks, demonstrating the broad generalization of our model **MMPROLONG**.

3 Experimental Setup

We conduct our LongPT experiments based on Qwen2.5-VL-7B [15], extending its original 32K context window to 128K. Following Dynamic-NTK [51], we scale the mRoPE base frequency from its original value of 1×10^6 to 4×10^6 by default. Unless otherwise specified, each LongPT run is trained with a fixed budget of 5B tokens, a maximum sequence length of 131,072 tokens, and a global batch size of 4M tokens. Throughout the paper, we use binary prefixes: $K = 2^{10}$, $M = 2^{20}$, and $B = 2^{30}$. We provide the full implementation details in Appendix A and the full evaluation details in Appendix B.

4 Multimodal Long-Context Data Curation

Recent studies [14, 52, 53] have identified data synthesis and mixture design as critical factors in pre-training, making data design a central focus of our study. For LVLMs, documents provide a natural source for synthesizing image-text data, as each page combines rich visual layout with dense textual content and can be rendered into long multimodal sequences. In this section, we first describe the preliminary step for constructing the document pool, which provides the raw visual-textual source for further data synthesis. Next, we discuss five approaches to synthesizing multimodal long-context data based on long documents, spanning long-document VQA and OCR transcription. Finally, we conduct experiments to evaluate which type of supervision provides the strongest LongPT signal.

4.1 Preliminaries: Document Pool Construction

To support scalable data synthesis, we construct a large-scale document pool comprising over 1.5 million PDF-formatted documents from multiple sources. The resulting pool spans a broad range of document types, including academic papers, books, and technical manuals, as well as diverse domains such as engineering, medicine, social sciences, and biology. The detailed statistics are shown in Appendix C.1. We further sample documents with 32 to 50 pages from this pool. Using the 2×2-pixel unshuffle in our Qwen2.5-VL backbone, these documents yield multimodal sequences ranging from 32K to 128K tokens. Meanwhile, we deduplicate those documents against all evaluation benchmarks by SHA-256 of the PDF content and by source-URL matching.

As LVLMs operate on images rather than PDF files, each PDF page is rendered to an image at $DPI = 144$ using PyMuPDF [2]. This resolution provides a practical trade-off between visual fidelity and storage cost. In addition, we use an OCR expert model fine-tuned from Seed 2.0 [4] to parse each rendered page into layout-aware blocks. These parsed blocks are further used in both synthesis families: title and section labels provide the section structure to sample coherent page segments for long-document VQA, while recognized text blocks serve as transcription targets for the OCR transcription families.

4.2 Long-document VQA Data Synthesis

Segment-level Synthesis Pipeline. We construct the long-document VQA training data using a short-to-long synthesis pipeline. The key idea is to first generate a QA pair from a short, semantically coherent document segment, and then place the QA pair back into its original full-document context to form long-context training instances.

Specifically, we first parse each document with our OCR expert model and identify its section structure using element labels, namely `title` and `section`. Based on the parsed structure, we randomly sample one or more consecutive sections whose total length spans 8–15 pages. This produces a coherent segment at the section level for QA generation.

[2] https://github.com/pymupdf/pymupdf

Next, we feed the page images of the sampled segment into Seed 2.0 [4], which serves as the QA-generation model. The model is prompted to generate a QA pair together with evidence descriptions and evidence pages. The detailed prompt is provided in Appendix D.3.

Finally, we recover the original full document corresponding to the sampled segment and combine it with the generated QA pair. This yields a single long-context VQA training instance, where the answer can be inferred from a localized short segment while the model must process the full long-document context.

Data Quality and Efficiency. Since we provide the QA-generation model with only 8–15 pages, the process relies on the model's strong short-context document understanding rather than forcing it to process the full long document. In this way, we find that the generated QA pairs are of high quality, and we also add a manual check in Appendix D.4. For efficiency, this pipeline substantially reduces the cost of generating large-scale data.

A key challenge in segment-level QA synthesis is ensuring that locally valid questions remain unambiguous when evaluated in the full-document context. Specifically, because QA pairs are generated from a short segment, the same question may have a different answer when placed back into the full document. For example, a question such as *"What is the reported revenue?"* may be answerable within the sampled section, but ambiguous in a full annual report where different sections report revenue for different departments or years. To avoid such global-context false positives, we require the QA-generation model to add explicit segment anchors to the question, such as *"in the Introduction section"* or *"on pages 20–25"*.

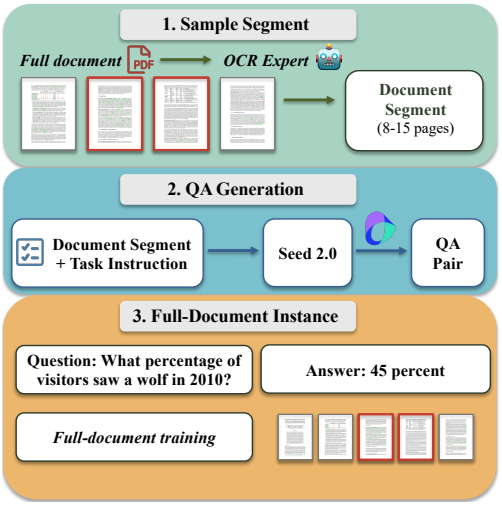


Figure 1: **Long-document VQA synthesis pipeline.** We first parse a full PDF with an OCR expert and sample a coherent document segment. Then, an LVLM generates a grounded QA pair from the sampled segment, which is finally placed back into the original full document to form a long-context training instance.

Data Types. With this segment-level QA synthesis pipeline, we synthesize three types of long-document VQA training data, each targeting a distinct capability. These data types are defined by the type and number of evidence pieces required to answer each question. Together, they span a spectrum from single-page extraction to multi-page extraction to reasoning over multiple pieces of evidence: (i) **single-page extraction** (`extract-single`) asks the model to retrieve factual information from a single page, e.g., "According to the Homemade Bitters recipe on Page 39, how long should the herbs soak in vodka?"; (ii) **multi-page extraction** (`extract-multi`) requires the model to aggregate factual information from multiple pages, e.g., "Based on Pages 6, 13, and 19, list all risk factors mentioned in the report."; and (iii) **reasoning** (`reasoning`) further requires numerical or logical operations over extracted evidence, such as summation, comparison, or counting across pages, e.g., "What is the difference between total consumption and total imports for rice production in 2020?"

Together, the first two types focus on locating and extracting relevant evidence from long documents, while the reasoning type further evaluates whether the model can derive answers by performing operations on the extracted evidence.

4.3 OCR transcription data synthesis

In addition to long-document VQA, another family of long-context data we build is document-level OCR transcription. This data family encourages LVLMs to capture long-distance image-text dependencies by requiring them to transcribe text elements across all pages of a long document.

Synthesis Pipeline. For each document, we first parse every page with our OCR expert model and retain text elements such as section titles, paragraphs, tables, and captions. We then construct an OCR transcription sequence by using the rendered page images as visual input and the parsed text elements as the target output. With this formulation, LVLMs must repeatedly attend back to the dense

Table 1: We compare long-document VQA data with OCR transcription data under the same setting. The models are evaluated on the document category of MMLongBench [5] at 64K and 128K, which contains three datasets: MMLongBench-Doc [46], LongDocURL [47], and SlideVQA [55]. We abbreviate them as MMLB-D, LD-URL, and SLIDE, respectively. **SFT** means an extra SFT stage.

	64K MMLongBench				128K MMLongBench				AVG.
Training data	MMLB-D	LD-URL	SLIDE	AVG.	MMLB-D	LD-URL	SLIDE	AVG.	
Qwen2.5-VL-7B	32.17	49.57	75.00	52.24	26.96	51.85	68.00	48.94	50.59
`extract-single`	33.85	59.73	77.00	56.86	30.89	55.69	77.00	54.53	55.69 +5.1
`extract-multi`	32.75	**64.32**	77.00	**58.02**	**31.50**	54.82	**81.00**	**55.77**	**56.90** +6.3
`reasoning`	32.67	60.34	**79.00**	57.33	29.23	**61.61**	76.00	55.62	56.47 +5.9
`OCR-full`	23.67	11.06	59.00	31.24	23.97	20.36	61.00	35.11	33.17 -17.4
`OCR-needle`	**40.07**	28.77	68.00	45.61	22.75	37.24	66.00	42.00	43.80 -6.8
`OCR-full` (SFT)	37.19	53.07	78.00	56.09	25.03	51.73	78.00	51.59	53.84 +3.2
`OCR-needle` (SFT)	33.79	50.38	78.00	54.06	26.65	49.84	76.00	50.83	52.44 +1.9

textual content in the rendered page images and transcribe it over long distances, thereby modeling long-distance image-text dependencies.

Data Types. Using this pipeline, we generate two types of OCR transcription training data. These types are defined by the scope of pages to be transcribed: (i) **full-document OCR** (`OCR-full`) requires the model to transcribe text elements from all pages of the document, encouraging dense image-text dependency modeling across the full context; and (ii) **needle-page OCR** (`OCR-needle`) selects only a small subset of pages (1–3 pages) for transcription and keeps the remaining pages as distractors, turning OCR transcription into a retrieval-style long-context training signal. Collectively, these two types encourage LVLMs to model long-distance image-text dependencies under both dense transcription and retrieval-style settings.

4.4 Comparing Long-Document VQA and OCR Transcription

We compare the five candidate synthesis tasks under a controlled 5B-token LongPT budget. For each task, we build a separate training set with at least 5B tokens and train Qwen2.5-VL-7B [15] using the hyperparameters in Section 3. Dataset statistics, such as token counts and sequence-length distributions, are provided in Appendices D.1 and E.1.

SFT for OCR Transcription Data. OCR transcription encourages long-distance image-text dependency modeling but is not naturally aligned with instruction-following evaluations. To favor OCR-based LongPT, we further apply a 5B-token SFT stage to the OCR-trained checkpoints using LLaVA-OneVision instruction data [54]; data details are in Appendix F.

Long-Document VQA Provides Stronger Supervision. The results are shown in Table 1. First, the 32K base model degrades substantially at 128K, with MMLongBench-Doc dropping from 32.17% to 26.96%. More importantly, OCR transcription yields poor downstream performance, especially full-document OCR, whose overall average drops to 33.17%. After adding the SFT stage to improve instruction-following ability, the OCR-trained checkpoints obtain moderate gains of 3.24% and 1.85% for full-document and needle-page OCR, respectively. In contrast, all three long-document VQA tasks consistently improve performance, with multi-page extraction achieving the best average of 56.90%. These results suggest that long-document VQA provides stronger and more compute-efficient supervision than OCR transcription for LongPT: it yields better downstream performance without requiring the additional 5B-token SFT stage. We therefore focus on long-document VQA in the remaining data-design experiments.

5 Data Mixture and Training Design

Having identified long-document VQA as an effective data source, we now study how to turn it into a practical LongPT recipe. Specifically, we examine three key design choices: the distribution of training instance lengths, the mixture of long-context data, and the preservation of short-context performance. We provide an additional ablation on the RoPE base frequency in Appendix G.3.

5.1 Training Sequence-Length Distribution

Table 2: Long-context data mixture test. We grid search the mix of information extraction vs. reasoning tasks from $0:10$ to $10:0$ under the fixed 5B-token budget. **Ratio** represents the extraction-to-reasoning ratio.

	64K MMLongBench				128K MMLongBench				AVG.
Ratio	MMLB-D	LD-URL	SLIDE	AVG.	MMLB-D	LD-URL	SLIDE	AVG.	
0:10	32.67	60.34	79.00	57.33	29.23	**61.61**	76.00	55.62	56.47
2:8	32.04	**65.02**	77.00	58.02	27.37	61.34	74.00	54.24	56.13
4:6	31.57	59.49	78.00	56.35	32.95	53.39	**79.00**	55.11	55.73
6:4	32.52	64.86	79.00	58.79	32.72	55.53	**79.00**	55.75	57.27
8:2	**36.00**	62.69	**80.00**	**59.56**	**34.19**	56.33	77.00	55.84	**57.70**
10:0	33.98	59.48	79.00	57.49	32.12	59.07	78.00	**56.40**	56.94

When extending the context window of LLMs, prior work [16, 34] often relies on books or code repositories from SlimPajama [56] or the Stack [57], whose sequence lengths are naturally distributed across 8K to 128K tokens. In contrast, our long-document pool contains a large number of documents ranging from 20 to 200 pages, providing sufficient coverage for constructing training instances at different target lengths. This raises a practical question: how should we choose the length distribution of synthesized training instances?

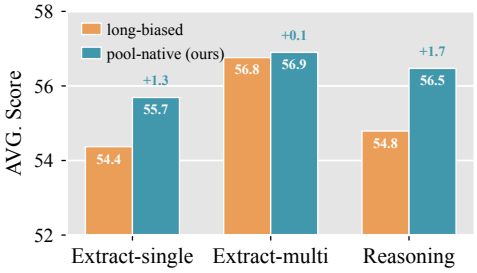


Figure 2: Comparison between different length distributions. We report overall average scores across 64K and 128K of the long-document VQA category from MMLongBench. Full results are provided in Appendix G.1.

Constructing data with different length distributions. Here, we study two length distributions, namely *pool-native* and *long-biased*. In the data-curation study (Section 4), we use the *pool-native* length distribution by default, as training instances are synthesized from documents naturally sampled within the 32–50 page range (32K-128K tokens), without additional length-based reweighting.

On the other hand, since we evaluate our models at a 128K context length, it is natural to ask whether allocating the token budget to longer examples leads to better LongPT results. We therefore construct a *long-biased* variant in which 83.9% of the examples contain at least 100K tokens, compared with only about 23.5%–23.6% in the *pool-native* distribution (See Table 13). This variant exposes the model more frequently to near-maximum-length contexts (128K), whereas the pool-native distribution covers a broader range of context lengths. The statistics of the *long-biased* data are summarized in Appendix D.2. As the *pool-native* set is used by default in Section 4, we further construct a *long-biased* training set for each type of long-document VQA data.

A diverse length distribution yields better long-context capability. The average performance of both length distributions is summarized in Figure 2, with full evaluation results provided in Appendix G.1. Overall, the *pool-native* distribution outperforms the *long-biased* distribution, yielding average gains of $+1.3$, $+0.1$, and $+1.7$ points for `extract-single`, `extract-multi`, and `reasoning`, respectively. Notably, despite evaluation with a 128K context window, the *pool-native* distribution consistently matches or outperforms the *long-biased* distribution while containing fewer near-128K training examples.

These empirical findings suggest that long-context ability is not a discrete capability acquired only at a specific target length, such as 128K. Instead, it requires continuous calibration across different absolute positions and relative image-text distances. In other words, LongPT should teach the model to retrieve key information in a way that generalizes across diverse long-context scenarios. Based on this observation, we adopt the *pool-native* distribution for all other experiments in this paper.

5.2 Multi-Task Long-Context Data Mixture

Data composition is critical for long-context extension in LLMs [16, 34]. So far, our studies have evaluated each type of long-document VQA data in isolation. We therefore study how to combine these three types of data into a single training mixture. Specifically, we group them into two

Table 3: **Short-context performance under different short-data mixing ratios.** We report the average over six short-context benchmarks together with per-benchmark scores. **Short Data** means the proportions of short-context data. The 0% means only using long-context data.

	General VQA		Multimodal Reasoning			Text Rec.	AVG.
Short Data	MMBench	RWQA	MMMU	MMMU-Pro	MathVista	OCRBench	
Qwen2.5-VL-7B	80.68	68.76	53.00	**37.80**	**70.50**	**88.10**	66.47
0%	80.00	**72.68**	49.33	36.65	68.70	85.50	65.48
20%	**81.82**	70.98	**54.33**	36.47	68.30	87.30	**66.53**
40%	81.25	70.46	52.67	37.05	68.30	87.10	66.14
60%	81.66	69.93	52.67	37.51	67.80	86.70	66.05
80%	81.14	69.67	52.11	37.40	69.30	87.40	66.17

categories: information extraction, which combines `extract-single` and `extract-multi` evenly, and reasoning, which corresponds to `reasoning`.

We grid search the extraction-to-reasoning ratio in 20% increments, ranging from all reasoning (0:10) to all extraction (10:0). As shown in Table 2, moderately extraction-heavy mixtures perform best, with 6:4 and 8:2 achieving the highest overall scores. The best performance is obtained with an extraction-to-reasoning ratio of 8:2, which also outperforms the best single-data setting in Table 1. This suggests that combining complementary data types is more effective than relying on any single type alone. It also indicates that retrieving key information from long-context inputs remains a major bottleneck when extending the context window, while retaining a small amount of reasoning data helps preserve task diversity. We therefore use the 8:2 mixture in subsequent experiments.

5.3 Short-context Performance Preservation

The degradation of short-context capabilities is a common concern in long-context continued pre-training. To examine this trade-off, we mix different proportions of short-context data into the LongPT stage while keeping the total token budget fixed. We obtain the short-context data from LLaVA-OneVision [54], with more details provided in Appendix F. We keep the 8:2 extraction-to-reasoning mixture fixed as the long-context component and vary the short-context data ratio from 0% to 80% in increments of 20%. We summarize the averages of long-document VQA performance in Figure 3, with full results provided in Appendix G.2. The short-context performance is reported in Table 3. In particular, short-context performance is evaluated on six benchmarks across three capabilities, as described in Appendix B.8.

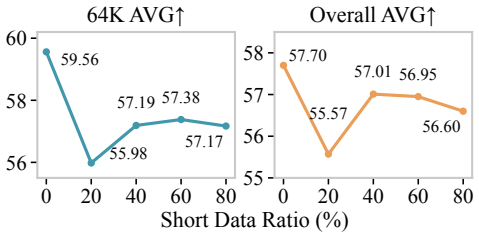


Figure 3: Long-document VQA performance under different short-data mixing ratios. We report 64K AVG and the overall AVG across 64K and 128K. Full results, including 128K scores, are provided in Appendix G.2.

Results and trade-off. A somewhat surprising observation is that pure long-context training does not severely degrade short-context capabilities. With 0% short-context data, the model achieves the best long-document VQA average of 57.70, while its short-context average only drops mildly from 66.47 to 65.48. This suggests that high-quality long-document VQA data can preserve much of the model's general short-context ability, possibly because its QA format still follows an instruction-following style despite the substantially longer input context.

Meanwhile, short-context mixing introduces a clear trade-off. Adding 20% short-context data yields the best short-context average of 66.53, but lowers the long-document VQA score to 55.57. In contrast, the 40% setting provides a better practical balance: it preserves long-document performance better, averaging 57.01, while keeping short-context performance close to the base model at 66.14.

Overall, since our goal is to maximize long-context capability without substantial degradation in short-context performance, we adopt pure long-context training without short-context data in the final recipe. The 40% setting can be viewed as a balanced alternative when stronger short-context preservation is required.

Table 4: **Final long-document VQA results.** We compare **MMPROLONG**, trained with the final LongPT recipe, against representative open-source and closed-source LVLMs at 64K and 128K.

Model	64K MMLongBench				128K MMLongBench				AVG.
	MMLB-D	LD-URL	SLIDE	AVG.	MMLB-D	LD-URL	SLIDE	AVG.	
Small open-source LVLMs (<15B)									
MMPROLONG (ours)	**36.00**	**62.69**	**80.00**	**59.56**	**34.19**	**56.33**	**77.00**	**55.84**	**57.70**
Qwen2.5-VL-7B	32.17	49.57	75.00	52.24	26.96	51.85	68.00	48.94	50.59
InternVL3-8B	27.72	52.84	70.00	50.19	25.89	48.44	58.00	44.11	47.15
InternVL3-14B	35.33	49.69	67.00	50.67	29.61	50.21	53.00	44.27	47.47
InternVL3.5-8B	29.26	47.94	38.00	38.40	18.28	31.89	0.00	16.72	27.56
InternVL3.5-14B	29.04	48.57	53.00	43.54	16.91	33.86	0.00	16.92	30.23
Gemma3-4B	25.00	33.33	57.00	38.44	26.44	33.38	53.00	37.61	38.03
Gemma3-12B	31.52	49.59	63.00	48.03	30.56	51.91	60.00	47.49	47.76
Gemma4-E2B	28.00	29.86	48.00	35.29	20.83	27.73	51.00	33.19	34.24
Gemma4-E4B	**36.00**	30.21	45.00	37.07	28.50	37.78	45.00	37.09	37.08
Large open-source LVLMs (≥ 15B)									
Qwen2.5-VL-32B	44.06	60.58	77.00	60.55	32.08	60.15	76.00	56.08	58.31
Qwen2.5-VL-72B	51.97	58.96	77.5	62.81	40.19	**62.45**	73.9	58.85	60.83
InternVL3-38B	36.00	49.32	74.00	53.11	29.65	50.33	54.00	44.66	48.88
InternVL3.5-38B	39.17	48.37	52.00	46.51	22.24	40.09	1.01	21.11	33.81
Gemma3-27B	38.00	51.72	69.00	52.91	27.94	61.13	68.00	52.35	52.63
Gemma4-26B-A4B	40.67	42.82	64.00	49.16	35.68	43.25	60.00	46.31	47.74
Gemma4-31B	**56.33**	**62.42**	**84.00**	**67.58**	**51.87**	62.00	**86.00**	**66.62**	**67.10**
Closed-source LVLMs									
GPT-5.4	61.56	72.37	87.10	73.68	52.96	73.06	N/A	63.01	69.41
GPT-5.5	**93.10**	**92.52**	**95.77**	**93.80**	**83.33**	**89.12**	N/A	**86.22**	**90.77**
Gemini-2.5-Pro	65.40	68.13	89.00	74.18	65.04	67.60	91.00	74.55	74.37
Gemini-3.1-Flash	74.43	79.32	91.00	81.58	69.63	76.38	90.00	78.67	80.13
Gemini-3.1-Pro	79.40	78.25	93.00	83.55	77.75	80.57	**93.00**	83.77	83.66

6 MMPROLONG Performance and Generalization

Based on the abovementioned design choices, we arrive at the final training recipe of **MMPROLONG**, with the full configuration in Appendix A. In this section, we first compare **MMPROLONG** with a wide range of LVLM baselines on long-document VQA. Besides this evaluation, we further show that **MMPROLONG** can generalize to (i) longer contexts up to 512K without further training or architectural modification; (ii) diverse long-context tasks without task-specific training, including MM-NIAH for webpage-based needle-in-a-haystack evaluation, VTCBench for long-context vision-text compression, and long-video understanding benchmarks. To further evaluate the generalizability of our training recipe, we also apply it to Qwen3-VL-8B in Appendix G.7.

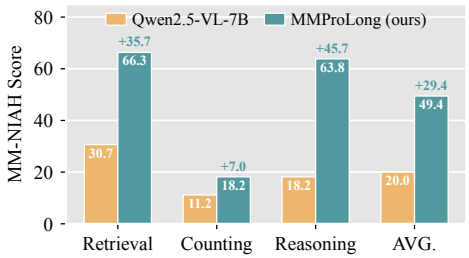


Figure 4: **MM-NIAH Scores.** We report the average of 64K and 128K scores for retrieval, counting, reasoning, and average. Additional baselines are provided in Appendix G.4.

6.1 MMPROLONG Compared with LVLM Baselines

We compare **MMPROLONG**, trained with the final recipe, against a wide range of open-source and closed-source LVLM baselines, as shown in Table 4. The full list of evaluated models is provided in Appendix B.3. Among open-source LVLMs below 15B parameters, **MMPROLONG** achieves the best overall average of 57.70, improving the Qwen2.5-VL-7B base model by 7.11 points from 50.59. The improvement is consistent across both context lengths, raising the average score from 52.24 to 59.56 at 64K and from 48.94 to 55.84 at 128K. Notably, **MMPROLONG** also outperforms several substantially larger open-source LVLMs, including InternVL3-38B and Gemma3-27B, achieving competitive long-context performance among open-source models.

6.2 Generalization Beyond the Training Setting

Generalization to Longer Contexts. In this experiment, we examine whether the final **MMPROLONG** recipe can extrapolate beyond its 128K training context. To this end, we further extend the

Table 5: We evaluate the final 128K **MMProLong** at 256K and 512K without additional training or adaptation. Each block reports MMLongBench-Doc (MMLB-D), LongDocURL (LD-URL), SlideVQA (SLIDE). Some baselines fail on 512K SlideVQA due to the large number of images.

Model	256K MMLongBench				512K MMLongBench				AVG.
	MMLB-D	LD-URL	SLIDE	AVG.	MMLB-D	LD-URL	SLIDE	AVG.	
MMProLong	29.69	**58.58**	**77.00**	**55.09**	**31.91**	**55.65**	**70.00**	**52.52**	**53.80**
Qwen2.5-VL-7B	25.47	35.88	53.00	38.12	13.44	24.61	20.41	19.49	28.80
Gemma3-4B	20.89	32.68	44.00	32.52	20.39	26.14	0.00	15.51	24.02
Gemma3-12B	**31.63**	47.47	63.00	47.37	24.18	46.37	0.00	23.51	35.44

long-document benchmarks to 256K and 512K contexts. Following MMLongBench [5], we construct these longer contexts by alternately padding the left and right sides with randomly sampled negative documents, while keeping the original questions and relevant evidence unchanged. Further details are provided in Appendix B.4. As shown in Table 5, **MMProLong** generalizes to these longer contexts without additional training or adaptation. While Qwen2.5-VL-7B degrades sharply as the context length increases, dropping from 38.12 at 256K to 19.49 at 512K, **MMProLong** maintains strong performance at both lengths, achieving 55.09 at 256K and 52.52 at 512K. This results in a significant overall advantage over the original Qwen2.5-VL-7B model, increasing the average score from 28.80 to 53.80.

Having shown that the final LongPT recipe improves long-document VQA performance and extrapolates to longer contexts, we next examine whether the learned long-context capability transfers to other multimodal tasks. Figures 4 and 5 report results on MM-NIAH and long-video understanding benchmarks, namely Video-MME [8], MLVU [18], and LongVideoBench [7]; full VTCBench results are provided in Appendix G.5.

MM-NIAH [6] is a multimodal needle-in-a-haystack benchmark that evaluates retrieval, counting, and reasoning tasks over webpage-based haystacks. As shown in Figure 4, **MMProLong** substantially improves over Qwen2.5-VL-7B, increasing the average score from 20.00 to 49.45 when averaging over 64K and 128K contexts. The gains are especially pronounced on retrieval and reasoning tasks, suggesting that long-document VQA training transfers to the ability to locate and use sparse evidence for reasoning in long multimodal contexts. Full results are shown in Appendix G.4.

We observe similar transfer on VTCBench and long-video understanding benchmarks. VTCBench evaluates LVLMs' ability to perform long-context vision-text compression across three tasks: retrieval, reasoning, and memory. As shown in Table 20 in Appendix G.5, **MMProLong** improves the overall score on VTCBench from 48.23 to 52.73, with gains on both reasoning and memory tasks while maintaining strong retrieval performance. On long-video benchmarks in Figure 5, **MMProLong** consistently improves over Qwen2.5-VL-7B on Video-MME, MLVU, and LongVideoBench, despite not using video-specific supervision; detailed scores are provided in Appendix G.6. Together, these results indicate that the proposed LongPT recipe learns a general long-context multimodal capability rather than overfitting to the document VQA format.

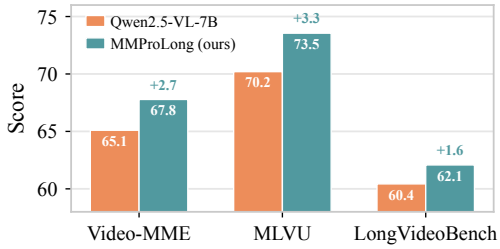


Figure 5: **Long-video generalization.** We report aggregate scores on Video-MME, MLVU, and LongVideoBench. Full results are provided in Appendix G.6.

7 Conclusion

In this work, we presented a systematic study of long-context continued pre-training for LVLMs, focusing on how to construct and mix effective multimodal long-context data. Our experiments show that long-document VQA is a strong and practical training task, as it provides diverse retrieval and reasoning supervision, preserves short-context capability, and supports context extension from 32K to 128K under a modest token budget. Instantiated as **MMProLong**, our recipe improves long-document VQA performance and generalizes beyond the training context window to 256K and 512K context lengths, as well as broader multimodal long-context tasks, such as long-video

understanding. We hope our study provides a practical foundation for building future LVLMs with reliable long-context capability.

Acknowledgments and Disclosure of Funding

The authors of this paper were supported by the National Key Research and Development Program of China (2025YFE0200500), the ITSP Platform Research Project (ITS/189/23FP) from ITC of Hong Kong, SAR, China, and the AoE (AoE/E-601/24-N), the CRF (No. C6004-25G), the RIF (R6021-20) and the GRF (16205322) from RGC of Hong Kong, SAR, China.

The authors declare no additional competing interests beyond the affiliations listed in this paper.

References

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

[2] Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, 2025. *URL https://ai.meta.com/blog/llama-4-multimodal-intelligence/*, 2025.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[4] Bytedance Seed. Seed2. 0 model card: Towards intelligence frontier for real-world complexity. Technical report, Technical report, Bytedance, 2025. URL https://lf3-static. bytednsdoc. com ....

[5] Zhaowei Wang, Wenhao Yu, Xiyu Ren, Jipeng Zhang, Yu Zhao, Rohit Saxena, Liang Cheng, Ginny Wong, Simon See, Pasquale Minervini, et al. Mmlongbench: Benchmarking long-context vision-language models effectively and thoroughly. *arXiv preprint arXiv:2505.10610*, 2025.

[6] Weiyun Wang, Shuibo Zhang, Yiming Ren, Yuchen Duan, Tiantong Li, Shuo Liu, Mengkang Hu, Zhe Chen, Kaipeng Zhang, Lewei Lu, et al. Needle in a multimodal haystack. *Advances in Neural Information Processing Systems*, 37:20540–20565, 2024.

[7] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

[8] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24108–24118, 2025.

[9] Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, et al. Webwatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025.

[10] Huanyao Zhang, Jiepeng Zhou, Bo Li, Bowen Zhou, Yanzhe Shan, Haishan Lu, Zhiyong Cao, Jiaoyang Chen, Yuqian Han, Zinan Sheng, et al. Browsecomp-v3: A visual, vertical, and verifiable benchmark for multimodal browsing agents. *arXiv preprint arXiv:2602.12876*, 2026.

[11] Yuxuan Zhang, Yubo Wang, Yipeng Zhu, Penghui Du, Junwen Miao, Xuan Lu, Wendong Xu, Yunzhuo Hao, Songcheng Cai, Xiaochen Wang, et al. Clawbench: Can ai agents complete everyday online tasks? *arXiv preprint arXiv:2604.08523*, 2026.

[12] The Gemini Team. Gemini 3.1 pro: A smarter model for your most complex tasks. `https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/`, Feb 2026. Google The Keyword Blog.

[13] OpenAI. Introducing GPT-5.4. `https://openai.com/index/introducing-gpt-5-4/`, Mar 2026. OpenAI Blog.

[14] Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

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

[16] Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*, 2024.

[17] Hongbo Zhao, Meng Wang, Fei Zhu, Wenzhuo Liu, Bolin Ni, Fanhu Zeng, Gaofeng Meng, and Zhaoxiang Zhang. Vtcbench: Can vision-language models understand long context with vision-text compression? *arXiv preprint arXiv:2512.15649*, 2025.

[18] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13691–13701, 2025.

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

[20] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[21] Google. Introducing gemini 2.0: our new ai model for the agentic era, 2024. URL `https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/#ceo-message`.

[22] Google. Gemini 2.5: Our most intelligent ai model. *URL https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/*, 2025.

[23] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

[24] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*.

[25] Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*, 2024.

[26] Yikai Zhang, Junlong Li, and Pengfei Liu. Extending llms' context window with 100 samples. *arXiv preprint arXiv:2401.07004*, 2024.

[27] Dawei Zhu, Nan Yang, Liang Wang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. Pose: Efficient context window extension of llms via positional skip-wise training. *arXiv preprint arXiv:2309.10400*, 2023.

[28] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*, 2023.

[29] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[30] Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Song Han, and Maosong Sun. Infllm: Unveiling the intrinsic capacity of llms for understanding extremely long sequences with training-free memory. *arXiv preprint arXiv:2402.04617*, 3(7), 2024.

[31] Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew Gormley. Unlimiformer: Long-range transformers with unlimited length input. *Advances in Neural Information Processing Systems*, 36:35522–35543, 2023.

[32] Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. Llm maybe longlm: Self-extend llm context window without tuning. *arXiv preprint arXiv:2401.01325*, 2024.

[33] Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. Effective long-context scaling of foundation models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4643–4663, 2024.

[34] Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128K context. *arXiv preprint arXiv:2402.10171*, 2024.

[35] An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyang Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. Qwen2.5-1m technical report. *ArXiv*, abs/2501.15383, 2025. URL `https://api.semanticscholar.org/CorpusID:275921951`.

[36] Anthropic. Introducing Claude Opus 4.7. `https://www.anthropic.com/news/claude-opus-4-7`, Apr 2026. Anthropic News.

[37] Austin Veselka. How to train your long-context visual document model. *arXiv preprint arXiv:2602.15257*, 2026.

[38] Mistral AI. Mistral small 3.1. `https://mistral.ai/news/mistral-small-3-1/`, Mar 2025. Mistral AI Research.

[39] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.

[40] Yunhang Shen, Chaoyou Fu, Shaoqi Dong, Xiong Wang, Yi-Fan Zhang, Peixian Chen, Mengdan Zhang, Haoyu Cao, Ke Li, Shaohui Lin, et al. Long-vita: Scaling large multi-modal models to 1 million tokens with leading short-context accuracy. *arXiv preprint arXiv:2502.05177*, 2025.

[41] Te Yang, Xiangyu Zhu, Bo Wang, Quan Chen, Peng Jiang, and Zhen Lei. Eea: Exploration-exploitation agent for long video understanding. *arXiv preprint arXiv:2512.03500*, 2025.

[42] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024.

[43] Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, et al. Longvt: Incentivizing" thinking with long videos" via native tool calling. *arXiv preprint arXiv:2511.20785*, 2025.

[44] Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. Bolt: Boost large vision-language model without training for long-form video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3318–3327, 2025.

[45] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

[46] Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[47] Chao Deng, Jiale Yuan, Pi Bu, Peijie Wang, Zhong-Zhi Li, Jian Xu, Xiao-Hui Li, Yuan Gao, Jun Song, Bo Zheng, et al. Longdocurl: a comprehensive multimodal long document benchmark integrating understanding, reasoning, and locating. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1135–1159, 2025.

[48] Hengyi Wang, Haizhou Shi, Shiwei Tan, Weiyi Qin, Wenyuan Wang, Tunyu Zhang, Akshay Nambi, Tanuja Ganu, and Hao Wang. Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models. *arXiv preprint arXiv:2406.11230*, 2024.

[49] Tsung-Han Wu, Giscard Biamby, Jerome Quenum, Ritwik Gupta, Joseph E Gonzalez, Trevor Darrell, and David M Chan. Visual haystacks: A vision-centric needle-in-a-haystack benchmark. *arXiv preprint arXiv:2407.13766*, 2024.

[50] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967, 2025.

[51] Emozilla. Dynamically scaled rope further increases performance of long context llama with zero fine-tuning, 2023. URL `https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/`.

[52] Gemma Team Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ram'e, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gael Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Róbert Istvan Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, Andr'as Gyorgy, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Boxi Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Pluci'nska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, J. Michael Wieting, Jonathan Lai, Jordi Orbay, Joe Fernandez, Joshua Newlan, Junsong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stańczyk, Pouya Dehghani Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Ardeshir Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vladimir Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab S. Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam M. Shazeer, Oriol Vinyals, Jeffrey Dean, Demis Hassabis, Koray Kavukcuoglu, Clément Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and L'eonard Hussenot. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[53] Google DeepMind. Gemma 4: Our most intelligent open models. *URL https://deepmind.google/models/gemma/gemma-4/*, 2026.

[54] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[55] Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: A dataset for document visual question answering on multiple images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13636–13645, 2023.

[56] Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. Slimpajama: A 627b token cleaned and deduplicated version of redpajama, 2023.

[57] Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, et al. The stack: 3 tb of permissively licensed source code. *arXiv preprint arXiv:2211.15533*, 2022.

[58] Qianli Ma, Yaowei Zheng, Zhelun Shi, Zhongkai Zhao, Bin Jia, Ziyue Huang, Zhiqi Lin, Youjie Li, Jiacheng Yang, Yanghua Peng, et al. Veomni: Scaling any modality model training with model-centric distributed recipe zoo. *arXiv preprint arXiv:2508.02317*, 2025.

[59] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.

[60] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 11198–11201, 2024.

[61] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Yue Cao, Yangzhou Liu, Weiye Xu, Hao Li, Jiahao Wang, Han Lv, Dengnian Chen, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Cong He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Ying Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Lijun Wu, Kai Zhang, Hui Deng, Jiaye Ge, Kaiming Chen, Limin Wang, Mingsong Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

[62] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[63] OpenAI. Introducing GPT-5.5. `https://openai.com/index/introducing-gpt-5-5/`, Apr 2026. OpenAI Blog.

[64] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36:71683–71702, 2023.

[65] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024.

[66] X.AI. Grok-1.5 vision preview. `https://x.ai/blog/grok-1.5v`, 2024.

[67] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9556–9567, 2024.

[68] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186, 2025.

[69] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

[70] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.

[71] Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24838–24848, June 2025.

[72] bloc97. NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation., 2023. URL `https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/`.

[73] bloc97. Add NTK-Aware interpolation "by parts" correction, 2023. URL `https://github.com/jquesnelle/scaled-rope/pull/1`.

[74] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

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

A Final LongPT Recipe and Implementation Details

We summarize the final LongPT recipe used for the main results in Section 6. The recipe is derived from the design choices studied in Sections 4 and 5: we use long-document VQA as the primary source of synthesizing data, naturally sample training sequences over the target length range, and use an $8:2$ extraction-to-reasoning mixture.

Here we list the full configuration of the final LongPT recipe:

Table 6: **Final training recipe for MMPROLONG**, which is selected from the ablation studies in Sections 4 and 5.

Long-Context Continued Pre-Training (LongPT)		
Long Data	Data Synthesis	`extract-single`, `extract-multi`, and `reasoning`
	Distribution	*Pool-native*: natural sampling from the document pool over the target length range $[32\text{K}, 128\text{K}]$
	Mixture	8:2 extraction-to-reasoning ratio. The three task ratio is 40%`extract-single`, 40%`extract-multi`, 20%`reasoning`
Short Data	Default	None (pure long-context data)
	Alternative	60% long-context data, 40% LLaVA-OneVision data for better short-context preservation
Model	Initialization	Qwen2.5-VL-7B-Instruct (original mRoPE base freq. 1×10^6)
	mRoPE for 128K	4×10^6
	Maximum length	131,072 (128K tokens)
Optim.	Token Budget	5B tokens (2.9K H20 hours)
	Optimizer	AdamW (weight decay = 0.1, β_1 = 0.9, β_2 = 0.95)
	LR schedule	1×10^{-5} with 10% warmup and cosine decay to 1×10^{-6}
	Batch size	4M tokens (32 sequences)
	Framework	VeOmni with FlashAttention
	Parallelism	Sequence parallelism size 2 and FSDP size 4

A.1 Full Implementation Details

We conduct our LongPT experiments based on Qwen2.5-VL-7B [15], whose original context window is 32K, and extend it to 128K. For RoPE base frequency, we follow Dynamic-NTK [51] and set the mRoPE base frequency from the original value of 1×10^6 to 4×10^6 by default. We present the ablation study of the base frequency in Appendix G.3. Then, we further apply the final training recipe to Qwen3-VL-8B [3] to test whether the recipe transfers across backbone models. We train both models with VeOmni [58], a scalable framework for multimodal pre-training. The optimizer used is AdamW, using a peak learning rate of 1×10^{-5}, cosine decay to 1×10^{-6}, and 10% linear warmup. Unless otherwise specified, each LongPT run is trained for a fixed budget of $5B$ tokens, with a maximum sequence length of $131{,}072$ tokens (128K) and a global batch size of 4M tokens, which contains 32 sequences per update. Throughout the paper, we use binary prefixes: $K = 2^{10}$, $M = 2^{20}$, and $B = 2^{30}$.

To speed up, we use FlashAttention [59] for efficient attention computation in long-context data. In addition, Ulysses sequence parallelism size 2 and FSDP size 4 are used to fit the 128K training configuration on a single 8-GPU NVIDIA H20 node; in practice, we train with 8 H20 nodes (64 GPUs in total) to improve throughput.

B Full Evaluation Details

We describe the details of our evaluation. Here, for tasks from MMLongBench [5], including long-document VQA and MM-NIAH tasks, we follow their evaluation and use their released code v1.1 [3]. Meanwhile, we use the VLMEvalKit [60] to evaluate LVLMs on other tasks, including VTCBench, long video benchmarks, and short-context benchmarks.

[3] `https://github.com/EdinburghNLP/MMLongBench`

B.1 Long-Document VQA

For the main long-document evaluation, we use the document category of MMLongBench [5], which contains MMLongBench-Doc [46], LongDocURL [47], and SlideVQA [55]. All examples in this category are instantiated at five standardized context lengths: 8K, 16K, 32K, 64K, and 128K tokens. Unless otherwise specified, we evaluate each model at 64K and 128K context lengths.

For grading, we follow the MMLongBench v1.1 evaluation protocol and report the official LLM-judged document QA score for each dataset. Specifically, this v1.1 protocol introduces LLM-based judging for the long-document VQA tasks, which handles different answer formats separately. For simple answer formats, such as string, integer, float answers, and "not answerable," the judge assigns a binary score indicating whether the predicted answer matches the reference answer as shown in Table 7. For list-style answers, the evaluator first extracts the predicted list and then computes an F1 score based on the overlap between the predicted list and the reference list. We show this prompt in Table 8. For each context length, the AVG. column is the macro average over the three datasets, and the overall AVG. is the macro average over the 64K and 128K results.

Table 7: Prompt for the binary answer judge in the long-document VQA category of MMLongBench. Gray highlighted spans denote placeholders filled at evaluation time.

```
Now your role is a grading teacher. Your task is to review and score student answers
based on reference standard answers. You need to notice the following key points:
- First, extract the final answer from the student's solution, then analyze and judge
whether the answer is correct.
- Scoring should only refer to the final answer obtained by the student; there is no
need to examine whether the intermediate problem-solving steps are correct.
- When analyzing and judging whether the answer is correct, you need to write down the
scoring rationale, organize it into clear statements that follow the logical flow.
The summary of the scoring rationale should be placed at the end, using the following
format: "In summary, the student's answer deserves x points" (where x represents the
student's specific score).
- Keep the whole process concise, within 150 words.
- Provide the score based on your analysis and display it in a code block in "JSON"
format.
- An item is covered if it is strictly mentioned or unambiguously implied by a
semantic equivalence. This includes numerical equivalence (e.g., 10% and 0.1),
synonyms (e.g., UK and United Kingdom), and plural/singular forms (e.g., "apple" and
"apples"). However, do not accept loosely related concepts.
Your output format is:
[Scoring Rationale]:
[Score]: x points
[JSON]:
{"answer_score": <integer_value>}

Below is the grading rubric:
[Scores]:
The scoring scale consists of 2 levels in total, from highest to lowest: 1 point, 0
points (the minimum is 0 points; if a situation arises where points need to be
deducted beyond 0, simply assign 0 points).
[Tier Details]:
1 point: Assign 1 point if the student's final answer matches the standard answer.
If the question has multiple sub-questions, all sub-questions must be answered to
assign 1 point.
0 points: Assign 0 points if the student's final answer does not match the standard
answer.

<in-context exemplars_1>

<in-context exemplars_2>

<in-context exemplars_3>

<test_case_1>
```

Table 8: Prompt for the list-style answer judge in the long-document VQA category of MMLongBench.

```
Now your role is a grading teacher.  Your task is to review and score student answers
for LIST-style questions, where the standard answer is a list of required items.
- First, extract the specific list of items from the <Student Answer>.  Ignore
conversational filler (e.g., "The answer is...").
- Then, compare the [Extracted List] against the <Standard Answer> (Ground Truth).
Here are some extra key points:
- The standard answer is a JSON-like list of items with each item as one required
element.  Determine whether each item is covered by the student's answer list.
- An item is covered if it is strictly mentioned or unambiguously implied by a
semantic equivalence.  This includes numerical equivalence (e.g., 10% and 0.1),
synonyms (e.g., UK and United Kingdom), and plural/singular forms (e.g., "apple" and
"apples").  However, do not accept loosely related concepts.
- You need to write down the extraction and comparing rationale, organize it into
clear statements that follow the logical flow.  The summary of the rationale should be
placed at the end, using the following format:  "In summary, the student's answer list
has X items, covering Y items from the reference list."
- Keep the whole process concise, within 200 words.
- Provide the student's answer item count and covered item count in a code block in
"JSON" format.
Your output format is:
[Rationale]:
[JSON]:
{
   "student_answer_count":  <integer_value>,
   "covered_count":  <integer_value>
}

<in-context exemplars_1>

<in-context exemplars_2>

<in-context exemplars_3>

<test_case_1>
```

B.2 Question-Type Composition

We summarize the composition of the long-document VQA evaluation from two perspectives in Table 9. LongDocURL provides official `task_tag` labels for understanding, reasoning, and locating questions. We also partition each dataset by annotated evidence pages and answer format: after separating unanswerable questions, we group answerable questions with multiple evidence pages as cross-page, and divide the remaining single-page questions into text, numeric, and list answers. The percentages below are reported for the 128K evaluation set. The 256K and 512K variants reuse the same questions and relevant evidence, so their question-type composition is unchanged by the negative-document padding.

B.3 Details of Long-document VQA Evaluation with the Final Recipe

In evaluating **MMProLong** trained with the final recipe, we compare it with a wide range of open- and closed-source LVLMs. For open-source models, we evaluate Qwen2.5-VL (7B, 32B, 72B) [15], InternVL3 (8B, 14B, 38B) [61], InternVL3.5 (8B, 14B, 38B) [62], Gemma3 (4B, 12B, 27B) [52], and Gemma4 (E2B, E4B, 26B-A4B, 31B) [53]. For closed-source models, we include GPT-5.4 [13], GPT-5.5 [63], Gemini-2.5-Pro [22], Gemini-3.1-Flash [12], and Gemini-3.1-Pro [12] with a high reasoning budget.

B.4 Longer-Context Evaluation up to 512K

To test whether the 128K recipe extrapolates to longer contexts, we further evaluate the same three long-document VQA datasets at 256K and 512K. The examples at 256K and 512K context lengths

Table 9: **Question-type composition of the long-document VQA evaluation.** (a) Official LongDocURL task labels. (b) Mutually exclusive groups based on evidence scope and answer format. Values are percentages of questions at 128K; each dataset sums to 100%. MMLB-D, LD-URL, and SLIDE denote MMLongBench-Doc, LongDocURL, and SlideVQA, respectively.

(a) LongDocURL official task labels

Understanding	Reasoning	Locating
63%	30%	7%

(b) Evidence scope and answer format

Question group	MMLB-D	LD-URL	SLIDE
Single-page text	17%	15%	50%
Single-page numeric	27%	25%	37%
Single-page list	6%	8%	0%
Cross-page evidence	28%	51%	13%
Unanswerable	22%	1%	0%

are obtained following MMLongBench [5]: we alternately pad the left and right sides with randomly sampled negative documents until the required length is reached. No additional training or inference-time adaptation is applied to tested models when evaluating at these longer lengths. We use the same official LLM-judged score and macro-average computation as in the main long-document evaluation.

B.5 MM-NIAH: Evaluation on Webpage Haystacks

MM-NIAH [6] is a multimodal needle-in-a-haystack benchmark that builds on the webpages from OBELICS [64]. In particular, the benchmark contains three task families: retrieval, counting, and reasoning, each with two variants: text-needle and image-needle. In our evaluation, we use the standardized version provided by MMLongBench [5], where each example is instantiated at five context lengths: 8K, 16K, 32K, 64K, and 128K tokens. In our study, we report results at 64K and 128K context lengths as we extend the context window from 32K to 128K. For reporting, we first average the text and image variants within each task family. Then, the average (AVG.) score is the macro average over the three task-family scores. We use the metrics provided by MMLongBench [5] for each task variant, including exact match, soft accuracy, and multiple-choice accuracy, depending on the specific subtask definition.

B.6 VTCBench: Evaluation on Vision-Text Compression

VTCBench [17] is another multimodal long-context benchmark that evaluates whether a model can preserve visual-text information under compressed visual context. We follow the VTCBench-Wild setting reported in the original paper and report all three task scores: Retrieval, Reasoning, and Memory. We follow the same grading rules as the original paper. Retrieval and Reasoning are measured by the official `containsAll` accuracy, which checks whether all ground-truth answers are contained in the model prediction. Memory is measured by the official LLM accuracy, where `gpt-4o-mini` judges answer correctness. The AVG. score follows the Overall column of VTCBench-Wild and is the sample-count weighted average over the three splits, with 800 retrieval examples, 800 reasoning examples, and 600 memory examples.

B.7 Long-Video Understanding Evaluation

For long-video understanding, we report the performance on Video-MME [8], MLVU [18], and LongVideoBench [7]. We evaluate these benchmarks using the 1fps variants and cap the maximum number of sampled frames per video at 768, with the total number of video tokens not exceeding 24,576. This configuration is fully aligned with Qwen2.5-VL technical report [15]. For Video-MME and LongVideoBench, we report overall multiple-choice accuracy. For MLVU, we evaluate the multiple-choice subset and report its average accuracy.

B.8 Short-Context Evaluation

To monitor whether **MMProLong** suffers degradation in general VLM ability, we evaluate short-context performance across three capabilities and six benchmarks. These include general VQA (MMBench-V1.1 [65] and RealWorldQA [66]), multimodal reasoning (MMMU [67], MMMU-Pro [68], and MathVista [69]), and text recognition (OCRBench [70]).

For MMMU, we use `MMMU_DEV_VAL` in VLMEvalKit, which combines the official MMMU development and validation splits. For MMMU-Pro, we use `MMMU_Pro_10c`, the standard 10-choice variant. For MathVista, we use the testmini subset. We report the official score of each benchmark and compute the short-context AVG. as the macro average over the six benchmark scores.

C Details of Preliminaries

C.1 Document Pool Statistics

We show the corpus scale, average page length, and language distribution of our document pool in Table 11. Meanwhile, the distribution of the document domain is shown in Figure 6.

C.2 OCR Expert

We use an OCR expert model fine-tuned from Seed 2.0 [4] to preprocess the rendered PDF pages. For each page image, the OCR expert parses layout-aware text blocks and assigns structural labels such as title, section heading, paragraph, table, figure caption, header, and footer.

These parsed blocks serve two purposes in our pipeline: (i) the title and section labels provide lightweight document-structure signals for sampling semantically coherent page spans when generating long-document VQA data; (ii) the recognized text blocks provide the target text for constructing OCR transcription baselines.

OCR capability. We evaluate the document-parsing capability of our OCR expert on OmniDocBench [71]. Table 10 reports Overall Edit for English and Chinese documents, where lower scores are better; our OCR expert achieves 0.129 and 0.209, respectively.

Table 10: **OCR model evaluation on OmniDocBench.** Lower Overall Edit scores are better. Our OCR expert supplies the transcription targets used in data synthesis.

Model	Overall Edit (EN) ↓	Overall Edit (ZH) ↓
Pipeline tools		
PaddleOCR-StructureV3	0.145	**0.206**
MinerU2.1.1-pipeline	0.162	0.244
Expert VLMs		
MinerU2.0-2505-0.9B	0.133	0.238
MonkeyOCR-pro-1.2B	0.146	0.221
OLMOCR-sglang	0.326	0.469
Our OCR expert	**0.129**	0.209

Table 11: Statistics of the document pool used for data synthesis.

Category	Statistic	Value
Corpus Scale	Number of documents	1,537,504
	Page-count range	[20, 200]
	Average Pages	23.80
	Total pages	36,592,809
Lang.	English documents	1,479,370 (96.22%)
	Chinese documents	55,202 (3.59%)
	Other languages	2,932 (0.19%)

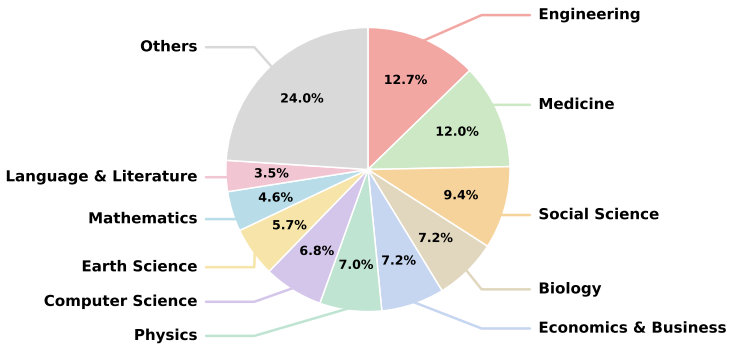


Figure 6: The document domain distribution of the document pool used for data synthesis.

D Long-Document VQA Training Data Details

D.1 Data Statistics for *Pool-Native* Distribution (Default)

We summarize the statistics of the long-document VQA training data in Table 12a, with the corresponding token-length distribution shown in Figure 7a. This data corresponds to the *pool-native* length distribution used by our final LongPT recipe, covering samples from 32 to 50 rendered PDF pages and approximately 32K–128K multimodal tokens.

Table 12: Statistics of the long-context training data used in our data-design study.

(a) Long-document VQA data in the *pool-native* Distribution (default).

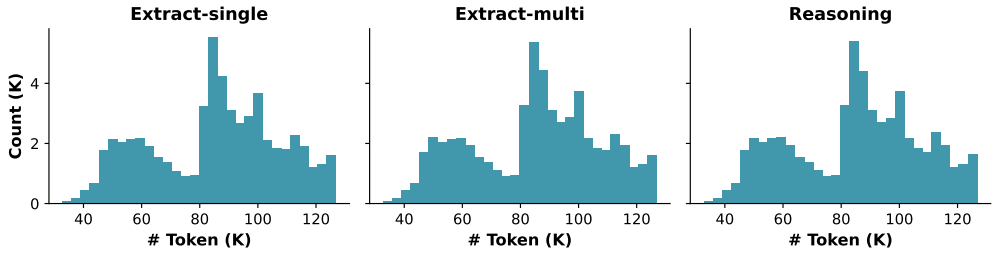
Task	# Samples	Avg. Pages	Avg. Tokens (K)	Total Tokens (B)	Page Range	Token Range (K)
`extract-single`	59,055	38.4	85.3	5.04	[32, 50]	[32.8, 126.8]
`extract-multi`	59,316	38.4	85.3	5.06	[32, 50]	[32.8, 126.8]
`reasoning`	59,403	38.4	85.3	5.06	[32, 50]	[32.8, 126.7]
Total	**177,774**	38.4	85.3	**15.16**	[32, 50]	[32.8, 126.8]

(b) Long-document VQA data in the *long-biased* distribution.

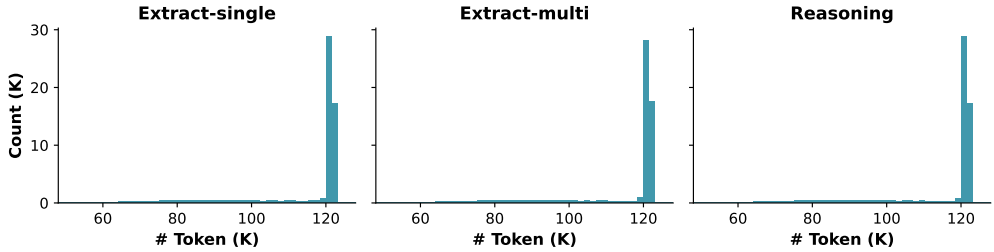
Task	# Samples	Avg. Pages	Avg. Tokens (K)	Total Tokens (B)	Page Range	Token Range (K)
`extract-single`	59,736	66.8	114.3	6.83	[50, 100]	[32, 125.3]
`extract-multi`	59,624	66.8	114.3	6.81	[50, 100]	[32, 124.0]
`reasoning`	59,770	66.8	114.3	6.83	[50, 100]	[32, 126.7]
Total	**179,130**	66.8	114.3	**20.47**	[50, 100]	[32, 126.7]

(c) OCR transcription data.

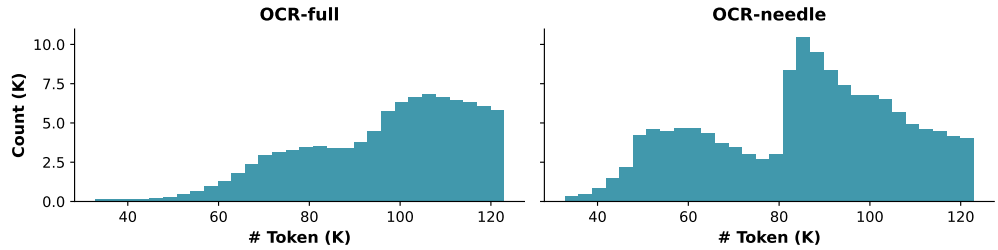
Task	# Samples	Avg. Pages	Avg. Tokens (K)	Total Tokens (B)	Page Range	Token Range (K)
`OCR-full`	97,336	39.6	96.4	9.38	[32, 50]	[32.8, 122.9]
`OCR-needle`	140,655	42.9	85.4	12.01	[32, 50]	[32.8, 122.9]
Total	**237,991**	41.6	89.9	**21.39**	[32, 50]	[32.8, 122.9]



(a) Long-document VQA data in the *pool-native* Distribution (default).



(b) Long-document VQA data in the *long-biased* distribution.



(c) OCR transcription data.

Figure 7: Token-length distributions of the long-context training data used in our data-design study.

Table 13: Percentage of training samples with token length at least 100K for both *pool-native* and *long-biased* distribution.

Distribution	`extract-single`	`extract-multi`	`reasoning`
pool-native	23.6%	23.6%	23.5%
long-biased	83.9%	83.9%	83.9%

D.2 Data Statistics for *Long-Biased* Distribution

We summarize the statistics of the *long-biased* long-document VQA training data in Table 12b. The corresponding token-length distribution is shown in Figure 7b.

Compared with the default *pool-native* distribution, this data is sampled from longer documents with 50–100 rendered PDF pages and concentrates more training mass near the upper end of the target context window. As shown in Table 13, 83.9% of the samples in the *long-biased* distribution contain at least 100K tokens across all three VQA tasks. In contrast, only about 23.5%–23.6% of the samples in the *pool-native* distribution exceed this threshold.

D.3 Prompt Template for QA Pair Generation

For each source document, we first use OCR block labels to identify title and section boundaries, then sample a semantically coherent span of 8–15 consecutive pages. The sampled page images are sent to the teacher LVLM together with a prompt that asks it to synthesize one document-grounded QA pair.

By default, we include two in-context exemplars sampled from a small manually curated exemplar pool. We instantiate the same core prompt with different task descriptions and page constraints for three data types: single-page extraction, multi-page extraction, and reasoning.

More concretely, Table 14 shows the prompt template used in our data sourcing process, and Table 15 summarizes the task description and extra restriction inserted for each task type.

Table 14: The prompt template used for long-document QA pair generation. Gray highlighted spans denote placeholders filled dynamically for each sampled document span.

```
[System]
You are an expert in synthesizing document question-answering dialogue. In this task,
I will provide you the images of one or more sections from a document and you need to
generate a question-answering pair based on the given sections. You need to notice
the following key points:

[Task Definition & Requirements]
The current document QA task is task description.

[General Restrictions]
- Questions must be confidently answerable and document-dependent, relying solely on
the provided visual content without external knowledge.
- If the question asks about missing information, the absence of such content must be
definitively verifiable from the provided image.
- Scope Constraint: Frame questions specifically within the scope of the provided
section/pages, strictly avoiding broad references like "across the document" or "the
whole file."
- Page Indexing Rule: When writing the evidence description or the "evidence_pages"
field in JSON, strictly use the index provided in the prompt (e.g., use '10' for "Page
10"), regardless of the page number inside images.
extra_restriction

[Response Format]
Please generate the response in two parts:

Part 1: Evidence Description
Provide a detailed description first, citing specific text or visual elements (using
the four evidence types defined below) to support your reasoning. You may use
multiple paragraphs if necessary.

When describing evidence, explicitly categorize the text and visual elements into one
of the following four types:
- Text: pure texts, such as paragraphs.
- Layout: text elements with special layout meaning (generalized text), such as
titles, headers, footers, table names, and figure names.
- Figure: including charts and general images.
- Table: structured data in rows and columns.

Part 2: JSON Output
Provide the final question, answer, and metadata in a strict JSON format.

Your answer must strictly fall into one of the following four categories. You must
also indicate the type in the JSON output:
1. String: General text, names, short sentences, or short phrases found in the
document.
- Example: "John Doe", "Financial Report", "The project was completed on time."
2. Integer: Whole numbers representing counts, years, page numbers, etc.
- Example: "42", "2023", "100".
3. Float: Numbers with decimal points, including currency, percentages, or
scientific measurements.
- Example: "12.5", "$45.20", "98.5%".
4. List: A collection of multiple items, names, or values.
- Example: ["Apple", "Banana", "Orange"], ["Item A", "Item B"].

Your output format is:
[Evidence Description]:
[JSON]:
{"question": <question>, "answer": <answer>, "answer_format": <answer_format>,
"evidence_pages": [<page_index>], "evidence_sources": [<evidence_sources>]}
<>

<in-context exemplars_1>

<in-context exemplars_2>

[Current Case]
The following images represent pages start_page to end_page of the document.
Generate the question-answering pair strictly based on the visual content below,
ensuring the question scope is limited to phrases like "the Introduction section",
"Pages 20, 21, and 25", or "Page 10."
current_case_image_sequence
```

Table 15: The task descriptions and extra restriction for the three long-document VQA tasks. We insert these parts into the prompt template to synthesize data.

Task	Task Description	Extra Restriction
`extract-single`	information extraction. You need to generate questions that focus on extracting specific, explicit information directly from the provided document images. Your objective is to create questions that ask for precise facts, entities (such as names, dates, locations, or authors), numerical values, lists of items, or specific steps in a procedure. The answers must be visually present and directly retrievable from the text, tables, or layout elements.	Single-Page Focus: Although multiple pages are provided for context, you must generate a question that is **strictly self-contained within a single page**. Select one specific page from the provided sequence and generate a question based solely on its content. The answer must be fully derivable from that single page without requiring cross-referencing or synthesizing information from other pages.
`extract-multi`	information extraction. You need to generate questions that focus on extracting specific, explicit information directly from the provided document images. Your objective is to create questions that ask for precise facts, entities (such as names, dates, locations, or authors), numerical values, lists of items, or specific steps in a procedure. The answers must be visually present and directly retrievable from the text, tables, or layout elements.	Multi-Page Priority: When multiple pages are provided, **prioritize** generating questions that require synthesizing information across different pages (>= 2 pages). For example, link a table header from one page to a row on the next, or aggregate data points scattered across multiple pages. You can also come up with other formats of multi-page questions. **Only fall back to single-page questions if no meaningful cross-page connections exist.**
`reasoning`	quantitative reasoning. You need to generate questions that require performing calculation, comparison, or counting. Your objective is to create questions involving one of the following: 1. Calculate: Performing arithmetic operations on data found in the document. 2. Compare: Comparing values to identify trends, maximums, minimums, or relative sizes. 3. Count: Counting the frequency of specific items, keywords, or visual elements that satisfy a condition. The answer must require the aforementioned processing step rather than being directly visible as a single contiguous string.	- Non-Trivial Synthesis: The question must require **aggregating multiple data points**. For example, ask "What is the difference in revenue between Q1 and Q2?" or "Calculate the total sum of items listed in Table 3." The key is that the user must find at least two pieces of information and combine them to get the answer.

D.4 Human Verification of QA Pairs

We randomly sample 100 generated QA pairs from the three task types and manually verify their answer correctness and evidence consistency. Among these inspected examples, 97 are fully correct: two contain incorrect answers, and one has an inaccurate evidence annotation. This suggests that the synthesis pipeline produces high-quality document-grounded supervision overall, while also indicating that the generated data may still contain a small amount of noise.

E OCR Transcription Details

In addition to VQA-based synthesis, we construct OCR transcription data as a contrasting family of long-context supervision. Each training example places rendered document pages as visual input and asks the model to generate the corresponding OCR text parsed from those pages. We consider two variants. Full-document OCR requires transcribing text elements from all pages of the document, creating dense image-text alignment across the full context. Needle-page OCR keeps the long visual context but only asks the model to transcribe text from a small subset of selected pages, making the task closer to retrieval over many distractor pages.

E.1 Data statistics

We summarize the statistics of the OCR transcription data in Table 12c. The corresponding token-length distribution is shown in Figure 7c.

F Short-Context Data Details

We use short-context data in two sets of experiments. First, in the comparison between long-document VQA and OCR transcription data (Section 4.4), we additionally apply supervised fine-tuning (SFT) with short-context instruction data after training the model on OCR transcription data. Second, in the short-data mixture experiment (Section 5.3), we mix short-context data with long-context data to study how much short-context supervision should be included in LongPT.

For both settings, we use publicly released short-context instruction data from LLaVA-OneVision [54]. We use all of these short-context instruction data in the short-data mixture experiment, while only the SFT portion is used in the additional SFT stage after OCR transcription training. We download the single-image and multi-image subsets from the LLaVA-OneVision Hugging Face repositories[4][5].

G Complementary Experimental Results

G.1 Training Sequence-Length Distribution

We provide the full dataset-level results for the sequence-length distribution ablation in Table 16.

Table 16: We compare the *pool-native* and *long-biased* distributions under each of the three VQA-based training tasks. We abbreviate these three datasets as MMLB-D, LD-URL, and SLIDE, respectively.

		64K MMLongBench				128K MMLongBench				AVG.
Training data	**Length**	MMLB-D	LD-URL	SLIDE	AVG.	MMLB-D	LD-URL	SLIDE	AVG.	
`extract-single`	pool-native	33.85	59.73	77.00	56.86	30.89	55.69	77.00	54.53	55.69 +1.3
	long-biased	31.47	55.88	75.00	54.12	29.48	59.40	75.00	54.63	54.37
`extract-multi`	pool-native	32.75	64.32	77.00	58.02	31.50	54.82	81.00	55.77	56.90 +0.1
	long-biased	36.29	56.75	80.00	57.68	30.72	55.79	81.00	55.84	56.76
`reasoning`	pool-native	32.67	60.34	79.00	57.33	29.23	61.61	76.00	55.62	56.47 +1.7
	long-biased	32.00	55.79	79.00	55.60	28.74	58.22	75.00	53.98	54.79

G.2 Short-Data Mixing for Long-Document VQA

We provide the full per-dataset long-document VQA results for the short-data mixing ablation in Table 17, complementing the AVG-only summary plotted in Figure 3.

Table 17: Short-context data test. We mix in 0% to 100% short-context data during LongPT under the fixed $5B$-token budget. The long-context data the 8:2 extraction-to-reasoning long-context mixture. **Short Data** means the proportions of short-context data. The 0% means only using long-context data.

Short Data	64K MMLongBench				128K MMLongBench				**AVG.**
	MMLB-D	LD-URL	SLIDE	AVG.	MMLB-D	LD-URL	SLIDE	AVG.	
0%	36.00	62.69	80.00	**59.56**	34.19	56.33	77.00	55.84	**57.70**
20%	32.33	57.61	78.00	55.98	31.49	56.00	78.00	55.16	55.57
40%	34.90	56.65	80.00	57.19	30.19	61.30	79.00	**56.83**	57.01
60%	33.62	57.53	81.00	57.38	31.93	58.61	79.00	56.52	56.95
80%	33.90	57.60	80.00	57.17	31.93	55.17	81.00	56.03	56.60

[4] `https://huggingface.co/datasets/lmms-lab/LLaVA-OneVision-Data`
[5] `https://huggingface.co/datasets/lmms-lab/M4-Instruct-Data`

G.3 mRoPE Base Frequency

Existing studies [24, 33, 72, 73] have shown that increasing the RoPE [74] frequency base during long-context continued pre-training or inference can improve long-context performance. Dynamic-NTK [51] suggests scaling the frequency base by $t^{\frac{d}{d-2}}$, where t denotes the context-window expansion factor and d is the attention head dimension.

However, LVLMs often use more structured positional encodings than plain 1-D RoPE. In our experiments, Qwen2.5-VL-7B adopts mRoPE [75], which decomposes rotary embeddings into temporal, height, and width components. As a result, visual position indices grow more slowly than a flattened 1-D sequence. It is therefore unclear whether the RoPE-scaling heuristic from Dynamic-NTK, originally developed for LLMs, directly applies to LVLMs with mRoPE.

We conduct an ablation study over the mRoPE frequency base for LVLM long-context training. By default, we follow Dynamic-NTK and scale the frequency base of the trained model from 1×10^6 to 4×10^6 when extending the context window from 32K to 128K tokens. We also evaluate two alternative bases, 2×10^6 and 8×10^6, to examine whether decreasing or increasing the default 4×10^6 base improves LongPT performance.

The evaluation results on long-document VQA tasks are shown in Table 18. Overall, moderately increasing the mRoPE base improves or maintains long-context performance compared with using a smaller base. Across the evaluated tasks, 2×10^6 and the Dynamic-NTK-scaled base 4×10^6 achieve comparable overall performance, with 4×10^6 slightly outperforming 2×10^6 on `extract-multi` and `reasoning`. Further increasing the base to 8×10^6 improves some individual metrics, but does not yield consistent gains across tasks and can degrade performance on `extract-multi` and `reasoning`. These results suggest that moderate mRoPE-base scaling is sufficient for extending LVLMs to longer contexts, while overly aggressive scaling is unnecessary. Based on this observation and to maintain consistency with the Dynamic-NTK heuristic, we set 4×10^6 as the mRoPE base in our main experiments.

Table 18: Ablation on the mRoPE frequency base for 128K LongPT. We grid search the mRoPE base in $\{2 \times 10^6, 4 \times 10^6, 8 \times 10^6\}$, starting from the original base of 1×10^6 used for the 32K context. The Dynamic-NTK heuristic [51] suggests 4×10^6 as the default base when extending the context window from 32K to 128K. Overall, 2×10^6 and 4×10^6 achieve comparable performance, while further increasing the base to 8×10^6 does not yield consistent gains across tasks.

		64K MMLongBench				128K MMLongBench				AVG.
Training data	**Freq.**	MMLB-D	LD-URL	SLIDE	AVG.	MMLB-D	LD-URL	SLIDE	AVG.	
`extract-single`	2×10^6	**35.52**	57.07	78.00	**56.86**	28.87	**58.65**	78.00	55.17	**56.02**
	4×10^6	33.85	**59.73**	77.00	56.86	**30.89**	55.69	77.00	54.53	55.69
	8×10^6	33.63	52.73	**79.00**	55.12	30.83	58.10	**79.00**	**55.98**	55.55
`extract-multi`	2×10^6	31.86	63.01	76.00	56.95	**31.60**	52.31	79.00	54.30	55.63
	4×10^6	32.75	**64.32**	77.00	**58.02**	31.50	**54.82**	**81.00**	**55.77**	**56.90**
	8×10^6	**33.36**	58.91	**79.00**	57.09	29.25	54.00	**81.00**	54.75	55.92
`reasoning`	2×10^6	33.04	57.20	77.00	55.75	33.44	58.66	**77.00**	**56.36**	56.05
	4×10^6	32.67	**60.34**	**79.00**	**57.33**	29.23	**61.61**	76.00	55.62	**56.47**
	8×10^6	**34.33**	57.86	73.00	55.07	**34.26**	54.61	72.00	53.62	54.34

G.4 MM-NIAH Generalization

We provide the full 64K and 128K MM-NIAH results in Table 19, complementing the averaged main-text summary in Figure 4.

Table 19: **MM-NIAH generalization.** Ret., Count, and Reas. average text and image needles.

(a) **64K MM-NIAH**

Model	Ret.	Count	Reas.	AVG.
MMProLong	**74.83**	**27.67**	**67.33**	**56.61**
Qwen2.5-VL-7B	50.00	6.00	27.50	27.83
InternVL3-8B	67.67	14.33	59.33	47.11
InternVL3.5-8B	56.12	4.45	57.23	39.26
Gemma3-4B	43.25	4.33	31.50	26.36
Gemma3-12B	65.58	27.00	42.25	44.94

(b) **128K MM-NIAH**

Model	Ret.	Count	Reas.	AVG.
MMProLong	**57.83**	8.67	**60.33**	**42.28**
Qwen2.5-VL-7B	11.33	16.33	8.83	12.17
InternVL3-8B	52.33	7.33	50.00	36.56
InternVL3.5-8B	3.45	0.00	2.27	1.91
Gemma3-4B	29.83	1.83	28.75	20.14
Gemma3-12B	48.83	**19.67**	33.75	34.08

G.5 VTCBench Generalization

We provide the full VTCBench-Wild results in Table 20. We report retrieval, reasoning, memory, and the sample-count weighted overall score following VTCBench-Wild.

Table 20: VTCBench generalization. We report retrieval, reasoning, memory, and the sample-count weighted overall score following VTCBench-Wild.

Model	Ret.	Reas.	Mem.	AVG.
MMProLong	**91.75**	**22.88**	**40.50**	**52.73**
Qwen2.5-VL-7B	91.63	15.63	33.83	48.23
Qwen3-VL-8B	89.00	11.50	33.67	45.73
InternVL3.5-8B	51.38	7.63	17.33	26.18
InternVL3.5-38B	45.81	8.75	22.33	25.93
Gemma3-27B	49.38	3.75	17.33	24.05

G.6 Long-Video Generalization

We provide the numeric long-video generalization results in Table 21, complementing the main-text summary in Figure 5.

Table 21: **Long-video generalization.** We report aggregate scores on Video-MME, MLVU, and LongVideoBench.

Model	Video-MME	MLVU	LongVideoBench
Qwen2.5-VL-7B	65.1	70.2	60.43
MMProLong	**67.78**	**73.55**	**62.08**

Table 22: Long-document VQA transfer across backbone models. We apply LongPT variants to Qwen3-VL-8B and report results on LongDocURL and SlideVQA. Since Qwen3-VL is already a 256K-context model trained with large-scale long-context pre-training, SFT, and RL, we focus on these two held-out document benchmarks.

	64K MMLongBench			128K MMLongBench			AVG.
Model	LD-URL	SLIDE	AVG.	LD-URL	SLIDE	AVG.	
Qwen3-VL-8B	56.36	74.00	65.18	60.11	72.00	66.05	65.62
+ **MMProLong** Recipe	62.22	73.00	67.61	62.81	72.00	67.41	67.51 +1.9

G.7 Generalization Across Backbones

We further apply our LongPT recipe to Qwen3-VL-8B [3] to test whether the training recipe transfers beyond Qwen2.5-VL. However, the Qwen3-VL series already includes 256K-context models trained with 100B tokens in long-context continued pre-training, and Qwen3-VL has undergone additional SFT and RL optimization for long-document tasks [3]. Therefore, this experiment is not intended as a strict study on context window extension. Instead, we use it as a diagnostic to examine whether the

Table 23: MM-NIAH transfer across backbone models. We report Retrieval, Counting, Reasoning, and average scores at 64K and 128K context lengths. Our recipe achieves better overall MM-NIAH score on Qwen3-VL-8B.

	64K MM-NIAH				128K MM-NIAH				AVG.
Model	Ret.	Count	Reas.	AVG.	Ret.	Count	Reas.	AVG.	
Qwen3-VL-8B	77.83	25.33	62.33	55.17	61.67	17.00	56.00	44.89	50.03
+ **MMPROLONG** Recipe	**83.00**	**45.67**	**70.83**	**66.50**	**74.17**	**35.00**	**61.83**	**57.00**	**61.75** +11.7

observed behavior transfers across backbones. We report the performance on long-document VQA tasks and MM-NIAH in Tables 22 and 23, respectively.

The results show that the final **MMPROLONG** recipe remains effective even on this stronger long-context backbone. On long-document VQA, **MMPROLONG** improves the average score from 65.62 to 67.51. The gains are more pronounced on MM-NIAH, where the average score increases from 50.03 to 61.75, with consistent improvements across retrieval, counting, and reasoning at both context lengths. Together with the diagnostic nature of this experiment, these results suggest that the proposed training recipe is not specific to Qwen2.5-VL-7B, but may also improve the long-context behavior of a newer backbone that already incorporates native long-context training.

H Limitations

For the training scale, our systematic study is primarily conducted on 7B/8B-scale LVLMs. This choice enables controlled comparisons across data recipes, context lengths, and training budgets, but it also leaves open how the observed trends scale to substantially larger models. Extending the same study to 30B or 70B-scale LVLMs, or to even longer context windows, like 512K or 1M, would require significantly higher computational cost, since long-context continued pre-training is expensive both in model size and sequence length. Although our transfer experiment on Qwen3-VL-8B provides initial evidence that the proposed recipe is not tied to a single backbone, a more comprehensive scaling study across larger model families remains an important direction for future work.

For evaluation, our long-document VQA experiments rely on model-based judging to assess answer correctness. Compared with lexical-overlap metrics, such judges can better handle semantically equivalent answers and free-form generation, but they also introduce additional API cost. This cost becomes substantial when evaluating many checkpoints, context lengths, and model variants, which limits the frequency and scale of evaluation during long-context training. Developing more efficient, reliable, and low-cost evaluation protocols for multimodal long-context models is, therefore, an important direction for future research.

I Broader Impact

Reliable long-context capability is important for deploying LVLMs in real-world scenarios that require understanding and reasoning over large multimodal inputs, such as long documents, webpages, videos, and agentic workflows. This work studies how to build this capability through long-context continued pre-training, with a particular focus on constructing effective multimodal long-context data under a modest training budget. Our findings suggest that carefully designed long-document VQA data can provide useful supervision for evidence retrieval and reasoning over long visual-textual contexts, and can transfer beyond documents to broader multimodal long-context tasks. We hope these results can help the community develop more data-efficient training recipes for LVLMs, reducing the need for excessive token budgets while improving models' ability to use long multimodal context.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, we made claims about building an effective training recipe to extend the context windows for LVLMs. The experiments in Sections 4 to 6 support those claims clearly.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We add a complete discussion of the limitations in Appendix H about trained models size and evaluation methods.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: Our work is about empirically finding a training recipe for extending the context window of LVLMs. Thus, our work does not have theory assumptions or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide complete details about training recipe Appendix A, evaluation details Appendix B, and data statistics Appendix D. More importantly, we will open-source all data and code.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the code and data samples in our submission, and will release them on acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide complete details about training recipe Appendix A, evaluation details Appendix B, and data statistics Appendix D. More importantly, we will open-source all data and code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report statistical significance based on multiple independent training runs. Such analysis would require repeating long-context continued pre-training under each data recipe and context setting, which is prohibitively expensive at 128K sequence length. Instead, we mitigate this limitation by conducting controlled comparisons under the same backbone, training budget, optimizer, and evaluation protocol, and by evaluating each trained model across multiple context lengths and downstream long-context tasks.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the details of computer resources and training hours in Table 6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All evaluation datasets used in this paper are based on previously open-sourced datasets, and all licenses are publicly available. Furthermore, our dataset collection and experiments ***do not*** involve any human subjects or participants. Our research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We include the broader impact in Appendix I.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: Our work studies long-context continued pre-training for LVLMs and may release related artifacts upon acceptance. For any release, we provide usage terms, documentation, and safety disclaimers covering intended research use, known limitations, and potential risks such as privacy leakage and incorrect long-context.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All evaluation datasets used in this paper are based on previously open-sourced datasets, and all licenses are publicly available.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will provide detailed documentation of our data and code in our GitHub Repository and HuggingFace Dataset on acceptance of the submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We synthesize data using LVLMs. Thus, there is no crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: Our work doesn't involve any crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We use Seed 2.0 to generate document-grounded QA pairs and an OCR expert fine-tuned from Seed 2.0 for document preprocessing. These uses are described in Section 4 and Appendix C.2, and the QA-generation prompts are provided in Appendix D.3.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.