

VTCBench: Can Vision-Language Models Understand Long Contexts with Vision-Text Compression?

Hongbo Zhao^{1,3*} Meng Wang^{2*} Fei Zhu^{2†} Wenzhuo Liu^{1,3} Bolin Ni^{1,3}
Fanhu Zeng^{1,3} Gaofeng Meng^{1,2,3} Zhaoxiang Zhang^{1,3†} ChengLin Liu^{1,3}

¹State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences

²Centre for Artificial Intelligence and Robotics, Hong Kong Institute of Science & Innovation, Chinese Academy of Sciences

³University of Chinese Academy of Sciences

zhaohongbo2022@ia.ac.cn meng.wang@cair-cas.org.hk

Abstract

The computational and memory overheads associated with expanding the context window of LLMs severely limit their scalability. A noteworthy solution is vision-text compression (VTC), exemplified by frameworks like DeepSeek-OCR and Glyph, which convert long texts into dense 2D visual representations, thereby achieving token compression ratios of 2x–10x. However, the impact of this high information density on the core long-context capabilities of vision-language models (VLMs) remains under-investigated. To address this gap, we introduce the first benchmark for VTC and systematically assess the performance of VLMs across three long-context understanding settings: VTC-Retrieval, VTC-Reasoning, and VTC-Memory. We further introduce VTCBench-Wild to evaluate robustness across diverse rendering configurations. We comprehensively evaluate leading open-weight and proprietary models on our benchmarks. The results indicate that several VLMs can retrieve information from rendered text in short-to-moderate contexts, but this ability degrades at longer lengths, and the degradation is especially pronounced on tasks requiring binding and integrating dispersed evidence. This study provides diagnostic evidence of VTC and serves as a foundation for designing more efficient and scalable VLMs. Code and data are available at <https://moenupa.github.io/VTCBench/>.

1 Introduction

The dramatic success of large language models (LLMs) has been accompanied by a persistent challenge: scaling to long contexts. Recently, vision-text compression (VTC), also known as context optical compression, has emerged as a new paradigm to address this bottleneck by rendering text into images, as illustrated in Figure 2a.

*Equal contribution.

†Corresponding author.

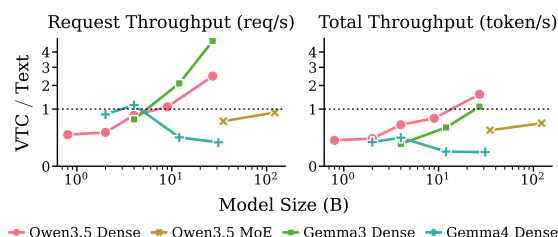


Figure 1: VTC versus pure-text efficiency comparison.

Recent works (Wei et al., 2025; Cheng et al., 2026; Xing et al., 2025; Wang et al., 2024a; Li et al., 2025) like DeepSeek-OCR (Wei et al., 2025) and Glyph (Cheng et al., 2026) propose rendering long text documents into compressed two-dimensional image sequences, leveraging the high information density of the visual modality. This cross-modal approach can reduce input-token usage and increase request throughput (Figure 1) by representing text as images. Beyond general text compression, recent work has explored VTC in many scenarios, including code understanding (Shi et al., 2026b), agent memory (Shi et al., 2026a), and self-compressing agent histories (Feng et al., 2026). This underscores its broad applicability and the growing need to understand how VLMs perform under this paradigm. By transforming the input space from a token sequence to a pixel grid, VTC shifts the burden of information management from sequential attention to spatial and visual reasoning.

As a new long context modeling paradigm, VTC has shown strong compression efficiency and promising performance on OCR-oriented (Ouyang et al., 2025) and long-context tasks (Bai et al., 2024b; Hsieh et al., 2024). However, many existing evaluations rely, explicitly or implicitly, on exact textual correspondence between the query and the context for success (Modarressi et al., 2025). This raises a central question:

Can vision-language models really understand long context with vision-text compression?

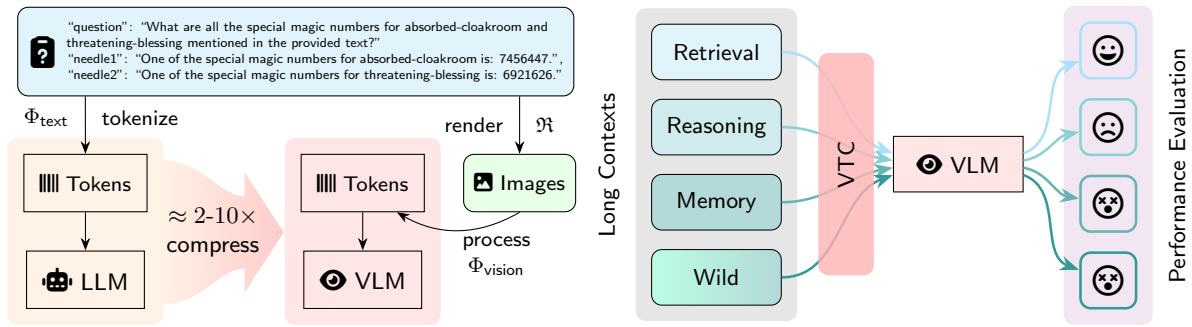


Figure 2: (a) Vision-text compression offers a more efficient alternative for long-context tasks: instead of feeding plain text directly to LLMs, it renders text as compact images for a VLM, achieving substantial input-token compression. (b) We propose VTCBench and VTCBench-Wild to comprehensively evaluate the long context understanding ability of VLMs within the vision-text compression framework.

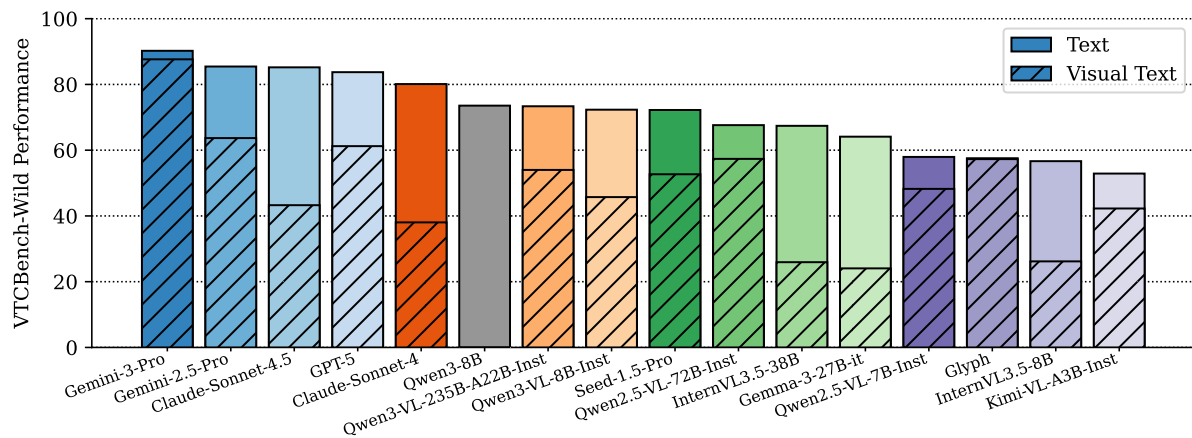


Figure 3: Performance of 16 VLMs & LLMs on VTCBench-Wild with (hatched) and without (solid) vision-text compression. Most models show substantial drops when the standard text-input pipeline is replaced by the end-to-end VTC input pipeline, while Gemini-3-Pro remains close to its text-input baseline.

In this work, we present *VTCBench* (Figure 2b), the first benchmark designed to evaluate long-context comprehension under the VTC framework. It covers three tasks: VTC-Retrieval, which tests locating and aggregating information in long compressed contexts; VTC-Reasoning, which tests associative reasoning with minimal lexical overlap between queries and context; and VTC-Memory, which evaluates long-range dialogue memory under vision-text compression. We further introduce VTCBench-Wild, a lightweight benchmark variant covering diverse rendering configurations, to evaluate whether VTC performance remains robust across the rendering settings encountered during deployment. In summary, these VTC-oriented long-context comprehension benchmarks establish a foundation to understand vision-text compression, offering crucial guidance for future design of highly efficient and capable next-generation long-context VLMs.

We conduct a comprehensive evaluation of leading open-weight and proprietary VLMs on VTCBench and VTCBench-Wild, yielding the following findings:

- Existing VLMs exhibit substantially weaker long-context comprehension under VTC than text-only LLMs, and this gap persists in same-VLM comparisons between VTC and the pure-text input pipeline.
- While VLMs remain relatively competent on simple retrieval, performance drops sharply on associative reasoning and long-term memory, indicating that VTC preserves basic text perception more than deeper long-context understanding.
- Notably, Gemini-3-Pro achieves near parity with its text-only baselines on VTCBench-Wild (Figure 3), demonstrating that strong

long-context understanding under VTC is achievable.

2 VTCBench

VTC introduces a new paradigm that uses visual modality as an efficient compression medium for textual information processing in LLMs. Formally, given a long text $\mathcal{T}$ with N_T tokens, a rendering operator $\mathfrak{R}$ converts it into m images $\mathcal{I} = \mathfrak{R}(\mathcal{T})$, which a VLM Φ_{VLM} encodes into N_I visual tokens. We define the VTC ratio as $r_{\text{VTC}} = \frac{N_T}{N_I}$.

It is important to distinguish VTC from visual document understanding. Document understanding tasks (e.g., DocVQA (Mathew et al., 2021)) process pre-existing visual artifacts, scanned pages, PDFs, or screenshots, whose layouts are fixed and given. In contrast, VTC starts from plain text and applies a fully controllable rendering operator r_{VTC} ; the visual representation is a design choice, not a property of the input. VTCBench evaluates this text-to-image compression paradigm, not document perception.

Because different VLMs convert the same image into different numbers of visual tokens, VTCBench includes two evaluation settings: predefined VTC ratio and predefined VTC rendering. Each setting covers three tasks that probe complementary long-context abilities: retrieval, reasoning, and memory. Examples can be found in Figure S3 in the Supplementary Material.

2.1 VTC Tasks

2.1.1 VTC-Retrieval

NIAH (Hsieh et al., 2024; Kamradt, 2023; Vodrahalli et al., 2024) test is a standard evaluation in natural language processing that measures a model’s ability to locate and retrieve target information from long inputs. These benchmarks are often characterized by retrieval-based tasks, where the query’s solution has a direct, verbatim correspondence within the source context. VTC-Retrieval is a visual NIAH task that provides dense-text images as context. Following RULER (Hsieh et al., 2024), we design 4 sub-tasks (Single NIAH, Multi-keys NIAH, Multi-values NIAH, Multi-queries NIAH) with different settings of “needle” (key-value pairs) inserted into a “haystack” (long distractor texts, i.e., Paul Graham essays (Kamradt, 2023)) to formulate context. Details of each task is in the Supplementary Material.

2.1.2 VTC-Reasoning

Literal matches make it easier for models to locate the relevant information and answer correctly (Modarressi et al., 2025), possibly *without* understanding the meaning of the text. Therefore, VTC-Reasoning adds associative reasoning to minimize literal overlap between retrieval questions and placed needle keys. Questions and needles are linked through associative relations, such as commonsense or real-world knowledge.

2.1.3 VTC-Memory

To complement retrieval-style tasks, we include a long-term memory setting. VTC-Memory evaluates whether models can distill, retain, and retrieve key information from long multi-turn dialogues. We adapt the LoCoMo (Maharana et al., 2024) corpus to construct dialogue contexts paired with questions, and cover four sub-tasks: single-hop, multi-hop, temporal, and open-domain. Additional details are provided in the Supplementary Material.

2.2 VTC settings

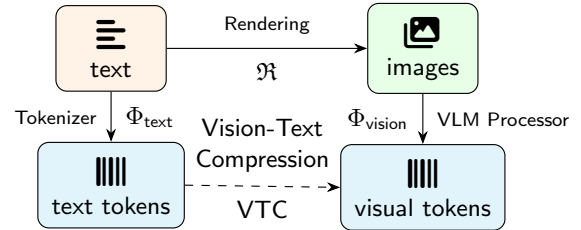


Figure 4: Illustration of vision-text compression as a composite function $\text{VTC} = \Phi_{\text{vision}}^{-1} \circ \mathfrak{R} \circ \Phi_{\text{text}}$, where its compression ratio r_{VTC} is a factor of the VLM Φ and the rendering operator $\mathfrak{R}$.

To systematically evaluate how VLMs perform under the VTC paradigm, it is crucial to control the experimental conditions. As illustrated in Figure 4, Vision-Text compression ratio r_{VTC} depends jointly on the rendering operator $\mathfrak{R}$ and the VLM Φ . We therefore define two complementary evaluation protocols, each answering a distinct question: (1) **Predefined VTC Ratio**: fix r_{VTC} by adapting $\mathfrak{R}$ for a given Φ , enabling comparison across models under the same compression budget; (2) **Predefined VTC Rendering**: fix $\mathfrak{R}$ and compare different Φ , ensuring all models process identical visual inputs so that performance differences reflect architectural capabilities rather than rendering variations.

2.2.1 Predefined VTC Ratio

To test the VTC paradigm across various VLMs while mitigating architectural differences, we adapt rendering configurations to VLM architecture (as detailed in Supplementary Material Section D) and control r_{VTC} to a predefined target value (e.g., $r_{\text{VTC}} = 2$). Specifically, we dynamically adjust the rendering operator $\mathfrak{R}$, i.e., the font size, according to the approximation below: for a given VLM and number of words n_{word} rendered into n_{img} images of size $(H_{\text{img}}, W_{\text{img}})$: $\text{fontsize}^2 \propto H_{\text{ch}} \cdot W_{\text{ch}} \propto \frac{n_{\text{img}} H_{\text{img}} W_{\text{img}}}{n_{\text{char_per_word}} \cdot n_{\text{word}}} \propto \frac{N_I}{n_{\text{char_per_word}} \cdot N_T} = \frac{1}{n_{\text{char_per_word}} \cdot r_{\text{VTC}}}$, where H_{ch} and W_{ch} denote the pixel height and width for a character, respectively; spacing such as kerning, line spacing, indentation, and margins are ignored; the number of characters per word $n_{\text{char_per_word}}$ is statistically constant. Therefore, we can estimate the font size of each model by $\text{fontsize} \propto \sqrt{\frac{1}{r_{\text{VTC}}}}$. To minimize performance variations caused by model-specific image preprocessing, we standardize the image size to 896×896 pixels, a dimension divisible by common patch sizes (e.g., 14, 16) and tile sizes (e.g., 448).

2.2.2 Predefined VTC Rendering

This setting evaluates a standardized rendering deployment. We fix rendering operator $\mathfrak{R}$ using a consistent preset across all models, making input images identical while allowing compression ratio r_{VTC} to vary by VLM architecture. We adopt plain-text preset $\mathfrak{R}_{\text{plain}} = \{\text{dpi} = 96, \text{font} = \text{Helvetica}, \text{size} = 12, \text{line-height} = 1.2, \text{margins} = 0, \text{img-size} = (896, 896)\}$, balancing four goals: standard typesetting conventions, vision-encoder optimization, fast evaluation, and improved vision-text compression. The 12-pt Helvetica at 96-dpi is chosen to maintain readability with moderate glyph density, while minimal spacing increases the amount of text per page. For VTC-Memory, we use $\mathfrak{R}_{\text{chat}}$ to mimic messaging app layouts, coloring one speaker’s utterances in green backgrounds and the other in white for clear speaker identification.

3 Experiments

3.1 VTCBench Results

We evaluate a range of VLMs on VTCBench, focusing on three core long-context understanding tasks. The results, detailed in Table 1, are discussed below.

3.1.1 Models and Inference Setup

We select thirteen VLMs: eleven open-weight models and two proprietary models. These model details are discussed in Supplementary Material Section D, spanning diverse sizes (3B to 235B) and architectures (dense or MoE). We also test Qwen3-8B (Yang et al., 2025), Qwen3-VL-8B (Team, 2025), and GPT-5 (OpenAI, 2025) as strong LLM baselines to show the performance gap between VLMs and LLMs. To ensure a fair apple-to-apple comparison, we further evaluate each VLM under both its visual-text and pure-text modes on VTCBench-Wild (Table 2), where the performance gap can be attributed to VTC compression rather than model capability differences. We evaluate all open-weight models with vLLM (Kwon et al., 2023) (model configurations in Supplementary Material Section E), running in BFloat16 on Nvidia A100 GPUs.

3.1.2 Task configurations

We test all models on two settings (predefined VTC ratio and predefined rendering) with the VTC-Retrieval, VTC-Reasoning, and VTC-Memory tasks. For VTC-Retrieval and VTC-Reasoning, we evaluate the models on the corpus from the series (1k, 2k, 4k, 8k, 16k, 32k). The evaluation is conducted by inserting a single needle at a specific position within the context. This process is repeated 11 times, with the needle placed at depths of 0%, 10%, 20%, ..., up to 100% of the context length. We found that most VLMs perform poorly when the corpus size reaches 32k, and therefore we do not consider longer texts.

3.1.3 VTC-Retrieval

As shown in Table 1, several advanced VLMs achieve high retrieval accuracy under short visual-text contexts, with models such as Qwen3-VL-235B and Glyph achieving competitive performance at 1k–4k lengths under the predefined compression ratio setting. However, all VLMs exhibit clear performance degradation as context length increases. In contrast, the text-only Qwen3-8B baseline maintains consistently high accuracy above 95% even at 32k tokens, highlighting a persistent gap between visual and textual long-context retrieval.

3.1.4 VTC-Reasoning

The performance gap becomes substantially larger in VTC-Reasoning. Although several proprietary

Model	r_{VTC}	VTC-Retrieval (containsAll $\uparrow$)						VTC-Reasoning (containsAll $\uparrow$)						VTC-Memory (ROUGE-L $\uparrow$)			
		1k	2k	4k	8k	16k	32k	1k	2k	4k	8k	16k	32k	SH	MH	TP	OD
Qwen3-8B (LLM)	–	98.86	98.86	96.82	97.16	96.59	95.57	94.18	81.45	66.18	49.82	36.18	17.45	52.55	30.23	27.83	17.96
Qwen3-VL-8B (LLM)	–	99.43	98.75	98.75	99.55	99.21	99.43	64.18	58.00	44.00	28.18	16.73	6.73	57.64	33.34	19.07	13.74
GPT-5 (LLM)	–	99.77	98.64	98.41	94.09	90.91	96.82	98.36	91.45	88.36	80.18	75.82	62.18	67.75	40.34	25.02	26.67
Predefined Compression Ratio $r_{\text{VTC}} = 2.00 \pm 0.02$ (and accordingly variable font size)																	
Kimi-VL-A3B	2.00	87.05	57.61	48.98	44.21	44.32	42.96	26.73	11.45	9.27	6.73	3.45	2.18	54.21	34.87	19.11	22.03
Gemini-2.5-Pro	1.98	100.0	53.64	54.54	54.55	38.64	40.57	96.18	92.51	32.36	15.82	10.18	OOM	43.28	26.18	14.69	17.53
Gemma-3-27B-it	2.00	88.86	84.89	82.27	74.43	OOM	OOM	19.09	16.91	12.18	7.64	5.27	OOM	26.31	19.08	10.76	7.20
GLM-4.1V-9B-Thinking	2.00	17.16	11.48	7.73	8.64	3.87	2.50	14.55	9.09	4.18	2.18	1.64	1.27	1.37	1.45	0.52	2.22
Glyph	2.00	91.48	81.37	84.89	78.53	80.46	75.68	40.09	22.07	17.65	11.76	9.12	5.05	4.10	2.43	1.69	2.16
GPT-5	2.00	81.93	74.55	71.14	62.73	60.68	57.16	58.73	41.82	25.64	26.82	25.00	22.73	50.41	30.81	27.36	24.38
InternVL3.5-8B	2.00	15.00	16.59	9.89	11.71	8.52	7.96	3.09	3.27	3.09	2.36	0.36	0.00	33.19	23.55	24.49	12.35
InternVL3.5-38B	2.00	20.11	14.66	11.82	12.96	9.78	9.89	4.18	5.82	3.27	3.09	1.09	0.18	40.96	27.92	18.54	17.19
Qwen2.5-VL-7B	2.00	85.23	75.57	74.77	76.82	76.59	69.21	32.60	15.52	11.13	6.71	3.86	1.60	52.22	32.60	23.76	18.30
Qwen2.5-VL-72B	2.00	88.98	84.55	87.73	86.59	82.27	76.48	52.79	36.55	33.29	24.04	15.61	9.52	55.55	31.61	20.96	18.75
Qwen3-VL-8B	2.00	95.23	88.41	87.39	88.98	86.93	79.59	23.27	21.45	18.18	9.27	6.73	4.18	46.09	29.31	23.63	15.43
Qwen3-VL-235B-A22B	2.00	97.16	88.52	94.77	92.16	91.21	81.34	23.27	20.55	19.82	17.09	11.82	8.91	45.10	32.62	26.77	19.71
Predefined rendering (font size)		$\mathfrak{R}_{\text{plain}}$ (12pt)													$\mathfrak{R}_{\text{chat}}$ (12pt)		
DeepSeek-OCR	3.12	18.18	21.82	14.66	0.34	0.00	OOM	1.69	0.38	0.06	0.00	0.00	OOM	OOM	OOM	OOM	OOM
Kimi-VL-A3B	2.00	87.05	57.61	48.98	44.21	44.32	42.96	26.73	11.45	9.27	6.73	3.45	2.18	54.21	34.87	19.11	22.03
Gemini-2.5-Pro	1.98	100.0	53.64	54.54	54.55	38.64	40.57	96.18	92.51	32.36	15.82	10.18	OOM	43.28	26.18	14.69	17.53
Gemma-3-27B-it	8.00	22.05	17.96	22.50	18.52	14.43	13.86	2.73	2.73	1.64	1.82	1.64	1.64	18.12	15.34	10.56	8.45
GLM-4.1V-9B-Thinking	2.00	17.16	11.48	7.73	8.64	3.87	2.50	14.55	9.09	4.18	2.18	1.64	1.27	1.37	1.45	0.52	2.22
Glyph	2.00	91.48	81.37	84.89	78.53	80.46	75.68	40.09	22.07	17.65	11.76	9.12	5.05	4.10	2.43	1.69	2.16
GPT-5	3.17	45.57	38.64	32.05	32.96	29.09	25.34	32.51	10.47	11.22	10.85	11.07	8.12	38.94	27.86	23.69	26.66
InternVL3.5-8B	1.56	23.64	21.82	23.64	20.80	16.02	12.50	7.45	6.00	3.64	2.91	3.09	0.73	22.29	18.30	21.38	15.65
InternVL3.5-38B	1.56	21.59	21.25	21.82	21.25	17.73	18.98	6.36	6.00	4.91	5.45	3.64	2.55	24.53	19.84	18.47	17.20
Qwen2.5-VL-7B	2.00	85.23	75.57	74.77	76.82	76.59	69.21	32.60	15.52	11.13	6.71	3.86	1.60	52.22	32.60	23.76	18.30
Qwen2.5-VL-72B	2.00	88.98	84.55	87.73	86.59	82.27	76.48	52.79	36.55	33.29	24.04	15.61	9.52	55.55	31.61	20.96	18.75
Qwen3-VL-8B	2.55	94.21	72.39	66.48	61.48	49.43	53.30	17.45	3.45	2.73	0.91	0.91	0.91	37.03	27.83	22.37	15.40
Qwen3-VL-235B-A22B	2.55	97.73	86.93	85.57	79.89	63.64	58.54	22.55	8.00	5.27	3.64	2.00	2.36	45.28	30.91	27.92	19.27

Table 1: VTCBench performance (%) across 3 tasks: retrieval, reasoning, and memory. r_{VTC} denotes vision-text compression ratio; for VTC-Retrieval and VTC-Reasoning, 1k, 2k, ..., 32k denotes equivalent text-form context length in text tokens; for VTC-Memory, SH, MH, TP, and OD denotes Single-Hop, Multi-Hop, Temporal, and Open-Domain subtasks, respectively. LLM baselines serve as upper-bound references. OOM indicates the model could not process the input at the given context length due to memory constraints, API input limits, etc.

or large-scale VLMs achieve strong results on short contexts, accuracy drops rapidly beyond 4k tokens, with most open-weight models approaching near-random performance at longer contexts. Compared with retrieval, reasoning tasks introduce minimal lexical overlap and require associative inference over visually compressed information, placing additional demands on evidence localization and semantic binding across rendered contexts.

3.1.5 VTC-Memory

Unlike the synthetic settings of previous tasks, VTC-Memory evaluates long-context understanding through cohesive dialogues, providing a more practical assessment scenario. As shown in Table 1, most VLMs still lag behind the text-only baseline, particularly on multi-hop (MH), tempo-

ral (TP), and open-domain (OD) problems that require information aggregation across turns. Notably, the Qwen2.5-VL series demonstrates strong robustness, achieving results competitive with or even surpassing the Qwen3-8B LLM baseline (e.g., Qwen2.5-VL-72B reaches 55.55% ROUGE-L on Single-Hop versus 52.55%). These results suggest that visually compressed high-density inputs can still support memory retrieval for some models and settings. Nevertheless, the sharp performance decline observed for most models on complex memory subtasks highlights that integrating interconnected evidence under visually compressed contexts remains difficult.

Crucially, Table 1 includes Qwen3-VL-8B and GPT-5 evaluated under both text-only and VTC modes, providing direct same-model controls. The

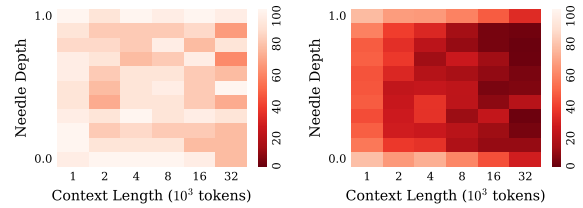
gaps are consistent with the cross-model trends above: at 4k context, Qwen3-VL-8B drops from 98.75% to 87.39% on retrieval and from 44.00% to 18.18% on reasoning; GPT-5 drops from 98.41% to 71.14% on retrieval and from 88.36% to 25.64% on reasoning. The observed gap is primarily attributable to the input modality, with VTC serving as the corresponding input-processing pipeline. Moreover, the larger drops on reasoning than on retrieval suggest that VTC places greater stress on evidence integration than on basic text access. A broader pure-text versus VTC analysis across all evaluated VLMs is presented in Section 3.2.

3.2 VTCBench-Wild

The above experiments isolate individual factors under controlled conditions. However, deploying VTC in practice involves choosing a rendering configuration where font, size, and spacing vary jointly, a combinatorial space that single-variable evaluation cannot cover. To this end, we introduce VTCBench-Wild, which samples from 99 rendering configurations to evaluate whether VTC performance is robust across diverse rendering choices. This benchmark comprises 2,200 samples selected from over two million unique combinations of VTCBench problems (28,600) and rendering configurations (99). Statistics and the derivation are provided in Table 3 and Section C, respectively.

We evaluated 15 VLMs and 1 LLM on VTCBench-Wild. The results in Table 2 reveal a clear hierarchy of capabilities and highlight architectural differences in processing visually diverse, compressed textual information. Beyond probing rendering robustness, VTCBench-Wild enables a controlled comparison by evaluating each model under both its vision-text (VT) and text-only (Text) modes. This controls for underlying model capability differences by comparing each model against its own text-input mode. The resulting gap reflects the effect of replacing the standard text-input pipeline with the end-to-end VTC input pipeline.

The text-only LLM baseline achieves consistently high performance across all three subsets, while most VLMs under VTC suffer a marked drop, particularly on reasoning and memory, with VT-Text gaps exceeding 20 points for the majority of models. A notable exception is Gemini-3-Pro, which maintains near-parity with its text-only mode (overall gap of only -2.59), demonstrating that the performance degradation observed in other models is not an inherent limitation of the



(a) VTC-Retrieval MQ-NIAH. (b) VTC-Reasoning.

Figure 5: Performance (%) on VTC-Retrieval and VTC-Reasoning with respect to needle placement depth using Qwen2.5-VL-72B-Instruct.

VTC paradigm. Other proprietary models, such as GPT-5 and Claude-Sonnet-4.5, despite competitive retrieval, still exhibit substantial reasoning gaps (>20 points), indicating that strong retrieval (perception) capability alone is insufficient for long-context reasoning under VTC. Among open-weight models, Qwen2.5-VL-72B shows balanced robustness, whereas InternVL3.5-38B and Glyph exhibit a recurring dissociation between textual perception and higher-level cognition, maintaining competitive retrieval while struggling on reasoning and memory. Overall, these findings suggest that parameter scale alone does not explain VTC robustness; architectures and training strategies likely play an important role in connecting spatial text perception with long-range semantic reasoning.

3.3 Analysis

3.3.1 Analysis of Rendering Configuration

To comprehensively assess the impact of the rendering operator $\mathfrak{R}$ on VLMs’ capabilities beyond OCR, we conducted ablation studies on VTC-Retrieval, VTC-Reasoning, and VTC-Memory tasks. All experiments employed the Qwen2.5-VL-7B and InternVL3.5-8B. We established a baseline using $\mathfrak{R}_{\text{plain-16}}$ for VTC-Retrieval and VTC-Reasoning, and $\mathfrak{R}_{\text{chat}}$ for VTC-Memory, and introduced perturbations to exactly one rendering attribute, selected from font size, font family, text/background color, and line height.

Table 4 shows that font size is the strongest observed factor of model performance across the three tasks. This is likely attributable to increased character legibility, which facilitates more accurate OCR by the model’s vision encoder. However, this improvement comes at the cost of a lower compression ratio. In contrast, the model demonstrates impressive robustness to stylistic changes: variations in font family and text or background colors

Model	Retrieval (containsAll $\uparrow$)			Reasoning (containsAll $\uparrow$)			Memory (LLMJudge $\uparrow$)			Overall		
	VT	Text	Gap	VT	Text	Gap	VT	Text	Gap	VT	Text	Gap
Qwen3-8B (LLM)	-	99.38	-	-	62.25	-	-	54.17	-	-	73.55	-
Gemini-3-Pro	99.88	100.00	$\downarrow 0.12$	97.63	99.64	$\downarrow 2.01$	58.00	64.67	$\downarrow 6.67$	87.64	90.23	$\downarrow 2.59$
Gemini-2.5-Pro	87.38	100.00	$\downarrow 12.62$	58.75	89.75	$\downarrow 31.00$	38.67	60.33	$\downarrow 21.66$	63.68	85.45	$\downarrow 21.77$
Claude-Sonnet-4.5	76.00	99.81	$\downarrow 23.81$	10.88	81.62	$\downarrow 70.74$	42.83	70.50	$\downarrow 27.67$	43.27	85.20	$\downarrow 41.93$
GPT-5	72.63	100.00	$\downarrow 27.37$	62.88	81.25	$\downarrow 18.37$	43.83	65.33	$\downarrow 21.50$	61.23	83.73	$\downarrow 22.50$
Claude-Sonnet-4	68.25	99.69	$\downarrow 31.44$	8.75	71.25	$\downarrow 62.50$	36.83	65.83	$\downarrow 29.00$	38.04	80.11	$\downarrow 42.07$
Qwen3-VL-235B-A22B	97.50	100.00	$\downarrow 2.50$	14.63	52.12	$\downarrow 37.49$	48.50	66.17	$\downarrow 17.67$	54.00	73.36	$\downarrow 19.36$
Qwen3-VL-8B	89.00	100.00	$\downarrow 11.00$	11.50	58.00	$\downarrow 46.50$	33.67	56.17	$\downarrow 22.50$	45.73	72.77	$\downarrow 27.05$
Seed1.5-VL	76.88	100.00	$\downarrow 23.12$	35.88	54.25	$\downarrow 18.37$	42.83	59.17	$\downarrow 16.34$	52.68	72.23	$\downarrow 19.54$
Qwen2.5-VL-72B	91.50	99.69	$\downarrow 8.19$	37.50	46.38	$\downarrow 8.88$	38.33	53.17	$\downarrow 14.84$	57.36	67.62	$\downarrow 10.25$
InternVL3.5-38B	45.81	98.50	$\downarrow 52.69$	8.75	47.88	$\downarrow 39.13$	22.33	52.00	$\downarrow 29.67$	25.93	67.41	$\downarrow 41.48$
Gemma-3-27B-it	49.38	98.44	$\downarrow 49.06$	3.75	45.62	$\downarrow 41.87$	17.33	43.00	$\downarrow 25.67$	24.05	64.11	$\downarrow 40.07$
Qwen2.5-VL-7B	91.63	94.12	$\downarrow 2.49$	15.63	27.12	$\downarrow 11.49$	33.83	50.83	$\downarrow 17.00$	48.23	57.95	$\downarrow 9.72$
Glyph	93.63	65.75	$\uparrow 27.88$	36.75	52.38	$\downarrow 15.63$	36.17	53.50	$\downarrow 17.33$	57.28	57.55	$\downarrow 0.27$
InternVL3.5-8B	51.38	96.31	$\downarrow 44.93$	7.63	22.62	$\downarrow 14.99$	17.33	49.17	$\downarrow 31.84$	26.18	56.66	$\downarrow 30.47$
Kimi-VL-A3B-Instruct	82.50	92.94	$\downarrow 10.44$	8.75	14.50	$\downarrow 5.75$	33.33	50.67	$\downarrow 17.34$	42.27	52.89	$\downarrow 10.62$

Table 2: VTCBench-Wild performance (%) with two input modalities: vision-text (VT) and text-only (Text). LLM-as-a-Judge uses gpt-4o-mini (Hurst et al., 2024).

# of tokens	Retrieval (800)	Reasoning (800)	Memory (600)
context (text)	11714.75	10848.24	35537.07
context (visual)	8581.86	8113.42	12436.85
problem	28.27	8.17	12.02
answer	9.54	1.91	7.23

Table 3: VTCBench-Wild Statistics. Tokens are calculated using Qwen3-VL-8B.

exert only minor effects on performance across all tasks, provided sufficient contrast is maintained. This suggests that the tested models are less sensitive to these stylistic changes than to readability-related factors. These findings suggest that the tested models are relatively robust to stylistic rendering changes and that readability appears to be a more important factor than stylistic variation in our tested VTC scenarios.

3.3.2 Analysis of Needle Position

Positional bias is known as a factor in text NIAH tasks (Hsieh et al., 2024; Modarressi et al., 2025). To examine this under the VTC paradigm, we visualize model accuracy subject to context length and needle position (Figure 5, more in Figures S5 to S8) on VTC-Retrieval and VTC-Reasoning.

The results show a pronounced “lost in the middle” phenomenon. Taking VTC-Reasoning as an example, accuracy forms a clear U-shaped curve: performance is highest when the needle is at the

very start (0% depth) or end (100% depth) of the context, and drops sharply in the central region. At 4k context length, accuracy falls to only 12.1% for middle-positioned needles. This positional bias is severely amplified as context length increases. With 16k context, the model still achieves over 40% accuracy for edge positions, but performance in the middle collapses to near-zero (e.g., 3.4%), making the central context nearly inaccessible for complex reasoning. These findings resemble the known lost-in-the-middle (Liu et al., 2024) pattern in text-based NIAH tasks, suggesting that VTC inputs can exhibit similar positional sensitivity.

3.3.3 Accuracy-Efficiency Pareto Trade-off

We further analyze the impact of the vision-text compression ratio r_{VTC} under fixed text lengths for both retrieval and reasoning. Increasing r_{VTC} is associated with lower performance in both tasks, revealing a clear accuracy-efficiency Pareto trade-off. Retrieval remains relatively robust under mild compression, maintaining high accuracy over short and medium contexts, but begins to degrade noticeably as compression becomes more aggressive, especially at longer lengths. Reasoning is substantially more sensitive: its performance is already limited even at low compression and declines steadily as either context length or r_{VTC} increases, approaching near-failure in the high-compression, long-context

Rendering Config	Qwen2.5-VL-7B		InternVL3.5-8B	
	Retrieval	Reasoning	Retrieval	Reasoning
$\mathcal{R}_{\text{plain-16}}$	98.79	36.73	71.82	17.82
$\cup \mathcal{R}(\text{font-size} = 10)$	73.64	20.00	14.24	5.27
$\cup \mathcal{R}(\text{font-size} = 12)$	86.06	27.64	36.06	10.18
$\cup \mathcal{R}(\text{font-size} = 14)$	99.39	31.27	71.82	20.55
$\cup \mathcal{R}(\text{font-size} = 16)$	98.79	36.73	71.82	17.82
$\cup \mathcal{R}(\text{font-size} = 18)$	100.00	39.09	72.73	15.45
$\cup \mathcal{R}(\text{font-size} = 20)$	100.00	46.36	65.45	24.73
$\cup \mathcal{R}(\text{font=Arial})$	99.09	35.82	36.06	10.18
$\cup \mathcal{R}(\text{font=CourierNew})$	98.79	38.18	36.97	10.73
$\cup \mathcal{R}(\text{font=TimesNewRoman})$	99.09	36.91	36.67	12.36
$\cup \mathcal{R}(\text{bg-color=red})$	98.79	41.82	24.24	8.00
$\cup \mathcal{R}(\text{bg-color=green})$	99.70	34.73	29.09	6.73
$\cup \mathcal{R}(\text{bg-color=gray})$	99.39	34.18	36.97	9.82
$\cup \mathcal{R}(\text{color=red})$	99.09	35.64	38.18	10.18
$\cup \mathcal{R}(\text{color=green})$	99.70	34.91	39.70	10.91
$\cup \mathcal{R}(\text{color=blue})$	99.09	37.09	35.76	10.91
$\cup \mathcal{R}(\text{color=gray})$	99.09	30.18	41.82	11.64
$\cup \mathcal{R}(\text{line-height} = 1.25)$	98.79	38.36	46.97	11.27
$\cup \mathcal{R}(\text{line-height} = 1.5)$	100.00	38.91	65.15	17.82
$\cup \mathcal{R}(\text{line-height} = 2)$	99.70	50.18	80.30	20.00

(a) Retrieval S-NIAH 1k and Reasoning 1k (containsAll $\uparrow$).

Rendering Config	Qwen2.5-VL-7B	InternVL3.5-8B
	Memory	Memory
$\mathcal{R}_{\text{chat}}$	50.54	22.29
$\cup \mathcal{R}(\text{font-size} = 7)$	21.81	19.90
$\cup \mathcal{R}(\text{font-size} = 8)$	29.29	20.15
$\cup \mathcal{R}(\text{font-size} = 9)$	33.81	22.92
$\cup \mathcal{R}(\text{font-size} = 10)$	40.93	21.65
$\cup \mathcal{R}(\text{font-size} = 11)$	45.54	22.74
$\cup \mathcal{R}(\text{font-size} = 12)$	50.54	22.29
$\cup \mathcal{R}(\text{font-size} = 13)$	49.94	25.34
$\cup \mathcal{R}(\text{font-size} = 14)$	53.06	29.96
$\cup \mathcal{R}(\text{font=Arial})$	49.79	23.70
$\cup \mathcal{R}(\text{font=CourierNew})$	53.15	23.63
$\cup \mathcal{R}(\text{font=TimesNewRoman})$	47.05	23.40
$\cup \mathcal{R}(\text{bg-color=white})$	49.81	24.06
$\cup \mathcal{R}_{\text{plain}}$	49.83	22.76
$\cup \mathcal{R}(\text{bg-color=red})$	49.14	28.02
$\cup \mathcal{R}(\text{bg-color=green})$	48.11	28.28
$\cup \mathcal{R}(\text{bg-color=blue})$	46.49	30.21
$\cup \mathcal{R}(\text{color=red})$	43.49	24.76
$\cup \mathcal{R}(\text{color=green})$	43.68	23.11
$\cup \mathcal{R}(\text{color=blue})$	39.93	24.32
$\cup \mathcal{R}(\text{color=gray})$	48.72	23.88

(b) Memory SingleHop (ROUGE-L $\uparrow$).

Table 4: VTCBench Performance (%) with respect to different rendering configurations, and each follow-up experiment alters exactly one configuration from **baseline**.

regime. Together, these results indicate that aggressive VTC weakens not only basic evidence access, but more critically, the semantic integration and inference required for long-context reasoning.

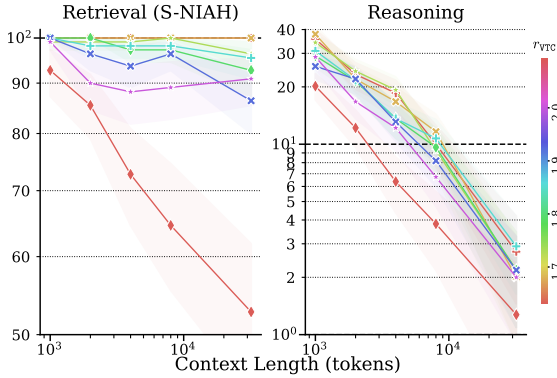


Figure 6: VTCBench Accuracy-Efficiency Pareto trade-off with respect to context length.

3.3.4 Decoupling Perception from Reasoning

In the VTC-Reasoning task, some errors appear to occur even when answer-bearing facts can be recovered from the rendered context. For example, a model may successfully extract the relevant facts (e.g., “Katie is a vegan”) but fail to perform the required logical inference (e.g., concluding that Katie cannot eat fish). To disentangle perception and reasoning, we perform a controlled ablation on VTCBench-Wild Reasoning using Qwen3-VL-8B-

Instruct, a representative model with strong OCR capability. We compare three input modality strategies: 1. *VTC*, directly providing the images, 2. *Text*, directly providing plain text, and 3. *OCR-Text*, a two-hop strategy where the VLM first extracts text from images and then undergoes pure-text reasoning, and we obtain results in Table 5. The Text $\rightarrow$ VTC gap is markedly larger than the Text $\rightarrow$ OCR-Text gap, suggesting that the principal bottleneck arises from reasoning over visual-tokenized representations rather than from OCR errors alone. Accordingly, in this representative case, VTC predominantly undermines reasoning over visual-tokenized text, as opposed to impairing perception or reasoning in isolation.

Input Strategy	Text	VTC	OCR-Text
Qwen3-VL-8B	58.00	11.50 $\downarrow 46.50$	40.63 $\downarrow 17.37$
InternVL3.5-8B	22.62	7.63 $\downarrow 14.99$	14.38 $\downarrow 8.24$

Table 5: VTCBench-Wild Performance (%) on Reasoning with different input modality strategies.

Beyond the perception–reasoning gap, Section F identifies several complementary failure modes: frequent refusal to perform associative reasoning in safety-aligned models, inefficient visual token allocation in thumbnail-based architectures, and incomplete retrieval of distributed evidence across long visual contexts.

4 Conclusion

This study presents VTCBench and VTCBench-Wild, the first systematic benchmarks evaluating VLMs on long-context understanding under the VTC paradigm. While current VLMs achieve adequate simple text retrieval in short-to-moderate contexts, performance often drops more sharply on complex associative reasoning and long-term memory tasks, highlighting a gap between visual-text retrieval and deeper semantic comprehension. However, Gemini-3-Pro’s strong performance on VTCBench-Wild suggests that strong VTC performance is achievable in some current systems. Furthermore, we find that rendering configurations, particularly font size, substantially affect model performance, with high compression ratios severely impairing retrieval. Ultimately, VTC’s efficiency gains currently come at the expense of advanced cognitive capabilities. Developing truly effective VTC-based models may require moving beyond simple scaling toward novel architectures and pre-training objectives that explicitly bridge spatial perception and abstract, long-range semantic reasoning.

Limitation

While this work provides the first systematic investigation into the long-context capabilities of VLMs under the VTC paradigm, we acknowledge several limitations, which offer clear directions for future research.

Our benchmark is currently confined to English general-domain text; VTC effectiveness may differ for non-alphabetic scripts (e.g., Chinese, Arabic) and for specialized content such as source code or legal documents. While VTCBench evaluates a broad set of current models, the VLM landscape evolves rapidly, and our findings represent a snapshot that may not fully generalize to future architectures with novel visual processing or attention mechanisms. Finally, proprietary models accessed through APIs operate as black boxes—potential pre-processing steps such as image resizing or compression, as well as input constraints (e.g., maximum image count), may confound our analyses of rendering configurations and their impact on performance.

Despite these limitations, we believe VTCBench establishes a crucial foundation and a rigorous evaluation framework for advancing the development of more efficient and capable long-context VLMs.

Acknowledgement

This work was supported by New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2025ZD0122902), the Beijing Natural Science Foundation (No. L257004), the National Natural Science Foundation of China (No. 62320106010, No. 62536010, No. 62506356), the Beijing Natural Science Foundation (No. L257015) and the InnoHK program.

References

- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.
- Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024a. Longalign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1376–1395.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024b. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.
- Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2024b. Longlora: Efficient fine-tuning of long-context large language models. In *International Conference on Learning Representations*, volume 2024, pages 8220–8238.
- Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198.
- Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihang Wang, Zhe Su, Xiaotao Gu, Xiao Liu, and 1 others. 2026. Glyph: Scaling context windows via visual-text compression. In

- Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 37145–37158.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.
- Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, and 1 others. 2023. Patch n’pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36:2252–2274.
- Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*.
- Lang Feng, Fuchao Yang, Feng Chen, Xin Cheng, Haiyang Xu, Zhenglin Wan, Ming Yan, and Bo An. 2026. Agentocr: Reimagining agent history via optical self-compression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5067–5086.
- Tianyu Fu, Haofeng Huang, Xuefei Ning, Genghan Zhang, Boju Chen, Tianqi Wu, Hongyi Wang, Zixiao Huang, Shiyao Li, Shengen Yan, and 1 others. 2024. Moa: Mixture of sparse attention for automatic large language model compression. *arXiv preprint arXiv:2406.14909*.
- Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2023. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations*.
- Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. 2024. Late chunking: contextual chunk embeddings using long-context embedding models. *arXiv preprint arXiv:2409.04701*.
- Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, and 1 others. 2025. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv e-prints*, pages arXiv–2507.
- Cheng-Ping Hsieh, Simeng Sun, Samuel Krman, Shantanu Acharya, Dima Rekish, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What’s the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.
- Zhiyuan Hu, Yuliang Liu, Jinman Zhao, Suyuchen Wang, WangYan WangYan, Wei Shen, Qing Gu, Luu Anh Tuan, See Kiong Ng, Zhiwei Jiang, and 1 others. 2025. Longrecipe: Recipe for efficient long context generalization in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11857–11870.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- DeLesley Hutchins, Imanol Schlag, Yuhuai Wu, Ethan Dyer, and Behnam Neyshabur. 2022. Block-recurrent transformers. *Advances in neural information processing systems*, 35:33248–33261.
- Huiqiang Jiang, Qianhui Wu, , Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.
- Gregory Kamradt. 2023. Needle in a haystack - pressure testing llms. https://github.com/gkamradt/LLMTest_NeedleInAHaystack.
- Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. In search of needles in a 11m haystack: Recurrent memory finds what llms miss. *arXiv preprint arXiv:2402.10790*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Yanhong Li, Zixuan Lan, and Jiawei Zhou. 2025. Text or pixels? it takes half: On the token efficiency of visual text inputs in multimodal llms. *arXiv preprint arXiv:2510.18279*.
- Fanfan Liu and Haibo Qiu. 2025. Context cascade compression: Exploring the upper limits of text compression. *arXiv preprint arXiv:2511.15244*.
- Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, and 1 others. 2025. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*.

- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870.
- Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.
- Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A Rossi, Seunghyun Yoon, and Hinrich Schütze. 2025. Nolima: Long-context evaluation beyond literal matching. *arXiv preprint arXiv:2502.05167*.
- Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. 2024. Leave no context behind: Efficient infinite context transformers with infinite attention. *arXiv preprint arXiv:2404.07143*, 101.
- OpenAI. 2025. [Introducing gpt-5 | openai](#).
- Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, and 1 others. 2025. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24838–24848.
- Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.
- Yarui Shi, Shugui Liu, Yu Yang, Wenyu Mao, Yuxin Chen, Qi Gu, Hui Su, Xunliang Cai, Xiang Wang, and An Zhang. 2026a. Memocr: Layout-aware visual memory for efficient long-horizon reasoning. *arXiv preprint arXiv:2601.21468*.
- Yuling Shi, Chaoxiang Xie, Zhensu Sun, Yeheng Chen, Chenxu Zhang, Longfei Yun, Chengcheng Wan, Hongyu Zhang, David Lo, and Xiaodong Gu. 2026b. Codeocr: On the effectiveness of vision language models in code understanding. *arXiv preprint arXiv:2602.01785*.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025a. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.
- Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, and 1 others. 2025b. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*.
- Qwen3-VL Team. 2025. [Qwen3-vl: Sharper vision, deeper thought, broader action](#).
- Michael Tschanen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, and 1 others. 2025. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*.
- Kiran Vodrahalli, Santiago Ontanon, Nilesh Tripuraneni, Kelvin Xu, Sanil Jain, Rakesh Shivanna, Jeffrey Hui, Nishanth Dikkala, Mehran Kazemi, Bahare Fatemi, and 1 others. 2024. Michelangelo: Long context evaluations beyond haystacks via latent structure queries. *arXiv preprint arXiv:2409.12640*.
- Alex Jinpeng Wang, Linjie Li, Yiqi Lin, Min Li, Lijuan Wang, and Mike Zheng Shou. 2024a. Leveraging visual tokens for extended text contexts in multi-modal learning. *Advances in Neural Information Processing Systems*, 37:14325–14348.
- Hengyi Wang, Haizhou Shi, Shiwei Tan, Weiyei Qin, Wenyuan Wang, Tunyu Zhang, Akshay Nambi, Tanuja Ganu, and Hao Wang. 2025a. Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3221–3241.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024b. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025b. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.
- Weiyun Wang, Shuibo Zhang, Yiming Ren, Yuchen Duan, Tiantong Li, Shuo Liu, Mengkang Hu, Zhe Chen, Kaipeng Zhang, Lewei Lu, and 1 others. 2024c. Needle in a multimodal haystack. *Advances in Neural Information Processing Systems*, 37:20540–20565.

- Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36:74530–74543.
- Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, and 1 others. 2024d. Memoryllm: Towards self-updatable large language models. *arXiv preprint arXiv:2402.04624*.
- Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.
- Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Hi-transformer: Hierarchical interactive transformer for efficient and effective long document modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 848–853.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.
- Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. 2025. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. In *International Conference on Learning Representations*, volume 2025, pages 37228–37253.
- Ling Xing, Alex Jinpeng Wang, Rui Yan, Xiangbo Shu, and Jinhui Tang. 2025. Vision-centric token compression in large language model. *Advances in Neural Information Processing Systems*.
- Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, and 1 others. 2024. 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.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. 2024. Compact: Compressing retrieved documents actively for question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21424–21439.
- Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986.
- Qianchi Zhang, Hainan Zhang, Liang Pang, Hongwei Zheng, and Zhiming Zheng. 2024. Adacomp: Extractive context compression with adaptive predictor for retrieval-augmented large language models. *arXiv preprint arXiv:2409.01579*.
- Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.
- Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

Supplementary Material

A Related Work

A.1 Long-Context Modeling

The challenge of efficiently processing extensive textual contexts remains a persistent pursuit for large models, driven by the inherent complexity of the standard Transformer self-attention mechanism (Liu et al., 2025). Recent advancements aim at developing long context language models that can be broadly categorized into architectural modifications and external component integration for workflow-based augmentation. Architectural modifications improve model efficiency internally through sparse and hierarchical attention (Beltagy et al., 2020; Fu et al., 2024; Xiao et al., 2025; Wu et al., 2021), and recurrent transformer (Munkhdalai et al., 2024; Hutchins et al., 2022). Additionally, positional encoding extrapolation methods (Su et al., 2024; Press et al., 2021; Xiong et al., 2024; Ding et al., 2024) adjust position encodings to handle larger sequence lengths. A complementary line of research utilizes external components to intelligently manage and compress context through workflow-based augmentation. Prompt compression (Zhang et al., 2024; Ge et al., 2023; Yoon et al., 2024; Jiang et al., 2023, 2024) reduces the input size through selective token retention or text summarization, thereby optimizing the utility of a fixed context window. Retrieval-augmented generation (Chen et al., 2024a; Günther et al., 2024) shifts the paradigm by replacing long-context ingestion with dynamic information extraction or recall. Memory-based methods (Zhong et al., 2024; Wang et al., 2023, 2024d) utilize external memory modules to store and retrieve long-term conversational or document history, ensuring contextual coherence over long interactions without burdening the active context window. Rather than generating a human-readable memory, C3 (Liu and Qiu, 2025) introduces a pure text-to-latent compression pipeline using two cascaded LLMs to achieve superior text compression ratios. Besides, there are also some training strategies (Chen et al., 2024b; Hu et al., 2025; Bai et al., 2024a) to enable the model to handle long contexts.

A.2 Vision-Text Compression

VTC presents a cross-modal paradigm that addresses the long-context challenge by leveraging visual encoding to increase information density.

Several recent studies have pioneered this framework. For instance, VIST (Xing et al., 2025) compresses distant, low-salience context by rendering it into images; these are subsequently processed by a lightweight vision encoder to produce semantically dense visual tokens, achieving a $2.3\times$ token reduction while maintaining state-of-the-art accuracy on in-context learning benchmarks. Similarly, VisIn-Context (Wang et al., 2024a) and Li et al. (Li et al., 2025) transform long textual sequences into visual representations, significantly lowering computational overhead while enhancing performance in document-understanding tasks. Further advancing this paradigm, DeepSeek-OCR (Wei et al., 2025) demonstrates the feasibility of high-ratio compression via optical 2D mapping, achieving over 96% OCR decoding precision at a $9\text{--}10\times$ compression ratio. Concurrently, Glyph (Cheng et al., 2026) renders long text contexts into compact images and processes them with VLMs. The system successfully demonstrated stable $3\text{--}4\times$ token compression for long text sequences, while preserving performance on OCR and other long-context processing tasks (Bai et al., 2024b; Hsieh et al., 2024). These works undertake a preliminary exploration of utilizing the visual modality as an efficient compression medium to facilitate long textual information processing.

A.3 Long-Context Comprehension Benchmarks

The Needle-In-A-Haystack (NIAH) test is a classic method for evaluating the long-context understanding capability of LLMs. The vanilla NIAH benchmark (Kamradt, 2023) evaluates simple retrieval robustness by embedding a specific fact (needle) within a long, irrelevant context (haystack). While useful for measuring basic context window functionality, it falls short of testing context understanding ability. Later, the NIAH task was reinforced with multiple targets, distractor noise, and fact references to enforce complex, or associative reasoning (Hsieh et al., 2024; Kuratov et al., 2024; Vodrahalli et al., 2024). For example, No-LiMa (Vodrahalli et al., 2024) requires models to perform indirect inference, where the query and the required context do not share direct lexical overlap, forcing the model to rely on deeper semantic understanding and association maintenance across long distances. Beyond text-only benchmarks, NIAH in multimodal tasks has also been explored recently (Wang et al., 2024c, 2025a). Sev-

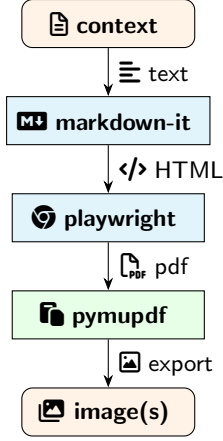


Figure S1: Rendering pipeline of VTCBench. **markdown-it** interprets context as HTML. **playwright** is a browser-based rendering engine that prints HTML to PDF, where styles are injected as CSS. **pymupdf** converts each page in PDF as an image.

eral benchmarks (Maharana et al., 2024; Wu et al., 2025; Chhikara et al., 2025; Zhong et al., 2024) have been designed to evaluate long-term dialogue memory and understanding over very extended conversations.

B Curation of VTCBench

VTCBench assesses models on three tasks (Table S1): information retrieval in VTC-Retrieval, associative reasoning in VTC-Reasoning, and long-term dialogue comprehension in VTC-Memory.

Permuting each task and context length, we randomly fill placeholders to form context-question-answer triplets. The rendering pipeline then converts context to images (examples in Figure S3). Finally, we prompt images and questions to VLMs (prompts in Section H) and evaluate their outputs against the ground truth answer.

B.1 Rendering Pipeline

Figure S1 depicts the rendering pipeline, a real-time browser-based text-to-image engine, which stacks markdown-it¹, playwright², and pymupdf³ for end-to-end conversion and applies text styling by injecting CSS. For example, the rendering preset $\mathfrak{R}_{\text{plain}}$ contains only the basic rule: "p{font-size:12px;}". $\mathfrak{R}$ majorly determines the compression ratio r_{VTC} : VTC-Reasoning with $\mathfrak{R}_{\text{plain}}$ yields an average of 2,000 text tokens per image, whereas its 16pt font size alternative

¹<https://pypi.org/project/markdown-it-py>

²<https://pypi.org/project/playwright>

³<https://pypi.org/project/pymupdf>

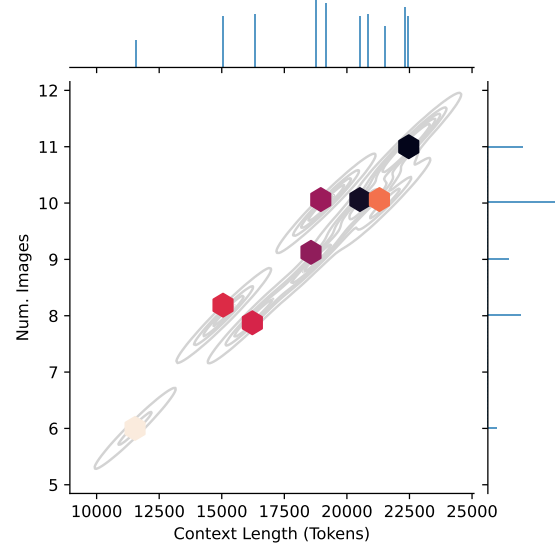


Figure S2: Distribution of context length (in number of text tokens) versus number of images for VTC-Memory with rendering operator $\mathfrak{R}_{\text{chat}}$. A strong correlation of approximately 2,000 tokens per image can be inferred.

$\mathfrak{R}_{\text{plain-16}}$ results in only 1,100 tokens per image, nearly halving r_{VTC} . As for VTC-Memory, which has predefined contexts, we observed a text-token versus number-of-image distribution in Figure S2, using rendering preset $\mathfrak{R}_{\text{chat}}$.

B.2 Task Specifications

B.2.1 VTC-Retrieval

VTC-Retrieval has 4 sub-tasks:

1. *Single NIAH (S-NIAH)*: the vanilla NIAH test, and its objective is to retrieve a single value (the “needle value”) according to a given key (the “needle key”) from an unrelated context with needle injected (the “haystack”).
2. *Multi-keys NIAH (MK-NIAH)*: multiple distinct key-value pairs are inserted, but its objective is to retrieve exactly one key’s value, with the rest serving as distractors.
3. *Multi-values NIAH (MV-NIAH)*: multiple values with an identical key are inserted, and its objective is to retrieve all values for the single key. This sub-task evaluates the model’s capacity for high-recall retrieval without omitting relevant information for a given key.
4. *Multi-queries NIAH (MQ-NIAH)*: multiple distinct key-value pairs are inserted, and its objective is to retrieve all of the values given all

Table S1: Illustrations of VTC-Retrieval, VTC-Reasoning, and VTC-Memory templates. We highlight **queries**, **keys**, **values**, and **distractors** accordingly. For VTC-Retrieval and VTC-Reasoning, **queries**, **keys**, **values**, and **distractors** are randomly generated; for VTC-Memory, **queries**, **keys**, **values**, and **distractors** are provided as-is.

[illegible]

- Figure S3: Examples of Rendered Context for VTC-Retrieval, VTC-Reasoning, VTC-Memory using $\mathfrak{R}_{\text{plain}}$, $\mathfrak{R}_{\text{plain}}$, and $\mathfrak{R}_{\text{chat}}$, respectively.

keys. This task evaluates the model’s proficiency in multi-query associative recall, testing its ability to return a complete set of requested items.

B.2.2 VTC-Memory

VTC-Memory contains four sub-tasks:

1. *Single-hop (SH)*: the answer can be derived from a single utterance or fact.
2. *Multi-hop (MH)*: the answer requires aggregating information from multiple turns.
3. *Temporal (TP)*: the answer depends on the ordering or timing of events within the dialogue.
4. *Open-domain (OD)*: the answer draws on world knowledge beyond the immediate conversation.

C Curation of VTCBench-Wild

To reflect context diversity and styling variety in the real world, we consider rendering and problems as independent variables, together producing 2,831,400 unique combinations. We randomly sample 800/800/600 image-question-answer triplets for Retrieval/Reasoning/Memory tasks to form VTCBench-Wild.

For the rendering pool, we permute 3 key CSS properties and obtain 99 ($11 \times 3 \times 3$) rendering configurations, including *font size*: 11 values ranging from 10 pixels to 20 pixels, *font family*: 3 fonts including Times New Roman, Helvetica, and Courier New, and *line height*: 3 values among 1.0, 1.2, and 1.5. Samples are drawn uniformly across the 99 rendering configurations, ensuring balanced coverage of the typographic parameter space.

For the problem pool, we uniformly sample all QA pairs (and permute context lengths for Retrieval and Reasoning) in three individual subtasks of VTCBench. VTC-Retrieval and VTC-Reasoning permute 6 context lengths (1k, 2k, 4k, 8k, 16k, 32k tokens) and place needles at 10% granularity (0%–100%, 11 positions), resulting in 7,920 and 19,140 unique QA pairs, respectively. VTC-Memory is fully static and provides 1,540 QA pairs.

D Calculation of Compression Ratio

Recall that the VTC ratio $r_{\text{VTC}} = \frac{N_T}{N_I}$ is the ratio between a context of N_T text tokens and its image counterpart requiring N_I image tokens. This

ratio is affected by both the rendering operator $\mathfrak{R} : \text{text} \mapsto \text{image}$ and VLM’s vision processing pipeline, so we provide some calculation examples of r_{VTC} for VLMs given $\mathfrak{R} = \mathfrak{R}_{\text{plain}}$. For simplicity, the following formulas ignore model-specific resizings and paddings, assuming an input image of size (H, W) pixels. By image-processing strategies, we can roughly categorize open-weight VLMs into three groups: (1) Models with dynamic resolution like the Qwen-VL series (Bai et al., 2025; Wang et al., 2024b; Team, 2025), Kimi-VL-A3B (Team et al., 2025b), and GLM-4.1V (Cheng et al., 2026; Hong et al., 2025); (2) Models using a global thumbnail, represented by the InternVL family (Wang et al., 2025b); (3) Other architectures with different visual pipelines like DeepSeek-OCR (Wei et al., 2025) and Gemma-3 (Team et al., 2025a) family.

D.1 Dynamic Resolution Models

D.1.1 Qwen2.5-VL Series

Qwen2.5-VL is a multimodal vision-language model series developed by the Qwen Team at Alibaba Group that understands and processes both images, videos, and text. It uses a NaViT (Dehghani et al., 2023) structure, and the visual token can be obtained by $\frac{H \times W}{14 \times 14 \times 4}$. Specifically, an 896×896 image is 1024 tokens.

D.1.2 Qwen3-VL Series

Qwen3-VL is a series of open-source vision-language models developed by the Qwen team with various versions, including dense and MoE (Mixture of Experts) architectures and Instruct and Thinking editions. Qwen3-VL uses a Siglip2 (Tschannen et al., 2025) vision encoder with dynamic resolution, and the number of visual tokens injected into the LLM is $\frac{H \times W}{16 \times 16 \times 4}$. Specifically, an 896×896 image is 784 tokens.

D.1.3 GLM-4.1V & Glyph

GLM-4.1V and Glyph use a similar structure as Qwen2.5-VL, and the calculation formula is $\frac{H \times W}{14 \times 14 \times 4}$.

D.1.4 Kimi-VL-A3B-Instruct

An efficient open-source VLM by MoonshotAI that uses a MoE architecture to handle multimodal tasks like image, video, and text understanding. Kimi-VL employs MoonViT as its native-resolution vision encoder; MoonViT is initialized from and continually pre-trained on SigLIP-SO-400M (Zhai

et al., 2023). Its image patch size is 14×14 ; it performs a 2×2 spatial downsampling while correspondingly expanding the channel dimension. Consequently, the number of visual tokens is $\frac{H \times W}{14 \times 14 \times 4}$. An 896×896 image input takes 1024 visual tokens.

D.2 Thumbnail-based Models

D.2.1 InternVL3.5 Series

InternVL3.5 uses InternViT (Chen et al., 2024c) to split an image into 448×448 tiles and a 448×448 thumbnail to capture the global context. Each tile refers to 256 tokens. For an 896×896 image input, we have $5 \times 256 = 1280$ tokens.

D.2.2 GPT-5

GPT-5 is a VLM developed by OpenAI, representing the fifth iteration in its series of generative pre-trained transformer (GPT) foundation models. We use GPT-5 with detail=high for all our experiments, and the pipeline⁴ is:

1. Scale to fit in a 2048×2048 px square.
2. Scale so that the image’s shortest side is 768px, and in our case 768×768 px.
3. Each 512px square (tile) in the image costs 140 tokens. 4 tiles cost 560 tokens.
4. Add the base 70 tokens to the total, yielding a final $560 + 70 = 630$ tokens per an 896×896 image.

D.3 Others

D.3.1 Gemini-2.5-Pro

Gemini-2.5-Pro is Google’s advanced reasoning model, excelling at complex problem-solving, coding, and analyzing large datasets, documents, and multimedia content. Gemini-2.5-Pro⁵ tiles images into 768×768 pixel tiles, each costing 258 tokens. This means an 896×896 image would cost $258 \times 4 = 1032$ tokens.

D.3.2 DeepSeek-OCR

DeepSeek-OCR is an open-source AI system that uses an innovative “Contexts Optical Compression” approach to read and compress documents, making it more efficient for LLMs. We use the Gundam model for DeepSeek-OCR, and the visual token is $4 \times 100 + 256 = 656$.

⁴<https://platform.openai.com/docs/guides/images-vision>

⁵<https://ai.google.dev/gemini-api/docs/image-understanding>

Model	Parameters
DeepSeek-OCR	temperature=0, ngram_size=30, window_size=90, whitelist_token_ids= {128821, 128822}, skip_special_tokens=False
Gemma-3 Series	top_k=64, top_p=0.95
Gemini-2.5-Pro	reasoning_effort=minimal
GLM-4.1V-9B-Thinking	top_k=2, top_p=0.6, temperature=0.8
Glyph	temperature=1.0
GPT-5	detail=high, reasoning_effort=minimal, verbosity=low
Kimi-VL-A3B-Instruct	temperature=0.2
Qwen2.5-VL Series	repetition_penalty=1.05, temperature=0.000001
Qwen3-VL Series	top_k=20, top_p=0.8, repetition_penalty=1.0, temperature=0.7
Qwen3-8B (LLM)	top_k=20, top_p=0.95, temperature=0.6

Table S2: Inference parameters of VLMs and LLM baseline.

D.3.3 Gemma-3 Series

Gemma-3 vision processor⁶ uses SigLip (Zhai et al., 2023), which resizes images to 896×896 pixels and applies 4×4 average pooling to output 256 tokens per image. This high-factor pooling strategy ensures a substantial compression ratio, condensing the high-resolution visual data into compact embeddings.

E Experimental Details

Table S2 presents inference parameters employed in our experiments. We follow the default generation parameters provided on HuggingFace and use the minimal possible thinking configuration.

F More Error Analysis

F.1 Refusal to Conduct Associative Reasoning

Beyond the perception–integration bottleneck identified in Section 3.3.4, we observe a distinct failure mode tied to model alignment. A notable failure

⁶<https://developers.googleblog.com/gemma-explained-whats-new-in-gemma-3>

VTC-Reasoning (Refusal)					
1k	2k	4k	8k	16k	32k
66.18	62.00	62.91	60.55	64.55	63.45

Table S3: Refusal ratio (%) of Qwen3-VL-235B-A22B-Instruct.

mode among more recent models like Qwen3-VL and InternVL3.5 is their frequent refusal to answer reasoning queries, exceeding over 60% as shown in Table S3. These models over-rely on exact lexical matching rather than employing the multi-step associative logic required for VTC-Reasoning. Consequently, when query phrasing lacks an exact literal match, the models would assume the information is absent. Despite strong retrieval capabilities, this over-conservative behavior seriously damages reasoning performance relative to earlier iterations (e.g., Qwen2.5-VL), a limitation we hypothesize stems from overly tuned safety alignments that erroneously flag associative queries as unanswerable.

F.2 Thumbnail Type Models

A specific failure mode arises from certain VLM architectural designs, such as InternVL3.5 (Wang et al., 2025b) and GPT-5⁷ (OpenAI, 2025). They typically involve encoding both a low-resolution “thumbnail” of the entire image to capture a global view, and high-resolution tiles for fine-grained detail. While this strategy is highly effective for natural images, where global composition and local details are distinct, it proves inefficient in the VTC paradigm. Images in VTCBench consist of uniformly dense text, and downscaled thumbnails generally appear as noisy mosaic textures that are challenging for VLMs to interpret clearly. Consequently, thumbnail tokens contribute little relative to the high-resolution tile tokens, and for InternVL3.5, this would mean a loss of 256 tokens, a fifth of the total 1280 tokens for an 896×896 image. This misspending of visual processing tokens represents a mismatch between VLM architectures and the nature of VTC, which likely contributes to the notable performance degradation observed in models like the InternVL3.5 series (Table 1).

⁷GPT-5’s image processing pipeline is opaque, but we hypothesize it employs image thumbnails, as it requires 70 “base tokens” per image.

Retrieval S-NIAH 1k (containsAll $\uparrow$)		
word-word	word-number	uuid-number
91.82	80.91	85.46

Table S4: Performance of Qwen2.5-VL-7B-Instruct on VTC-Retrieval with respect to needle key-value types.

$\mathfrak{R}$	Memory (ROUGE-L $\uparrow$)			
	SingleHop	MultiHop	Temporal	OpenDomain
$\mathfrak{R}_{\text{chat}}$	4.10	2.43	1.69	2.16
$\mathfrak{R}_{\text{plain}}$	30.92	6.49	7.48	13.54

Table S5: Performance of Glyph against different rendering—green text backgrounds $\mathfrak{R}_{\text{chat}}$, or white $\mathfrak{R}_{\text{plain}}$.

F.3 Inaccurate Information Aggregation

In the VTC-Retrieval MV-NIAH subtask, two needles share the same key, but are assigned different values and placed at least 20% of the total context length apart. The model is required to synthesize these cues and output a composite answer that reflects both values. As shown in Figure S4, the performance gap between containsAny (finding at least one value) and containsAll (finding both) widens slightly with increasing context length. This result suggests that the primary bottleneck is not the aggregation logic itself, but rather the failure to retrieve all relevant pieces of information from extensive visual contexts. The model is generally capable of synthesizing facts it can find, but its ability to find all facts diminishes over longer sequences.

F.4 Sensitivity to Needle Types

In the VTC-Retrieval S-NIAH sub-task, we isolate three sub-conditions that each pair a specific needle key type with a specific needle value type. By keeping all other factors constant, we can attribute performance gaps directly to the nature of the needle. The results (Table S4) reveal a clear hierarchy: The results show a systematic decline in retrieval accuracy as the semantic “richness” of the needle diminishes. This finding aligns with the observations reported by Hsieh et al. (Hsieh et al., 2024) on the brittleness of retrieval-oriented LLMs to identifier-like queries.

F.5 Sensitivity to Rendering Configurations

OCR-specific VLMs are known to be sensitive to text formatting, e.g., image resolution, typographic

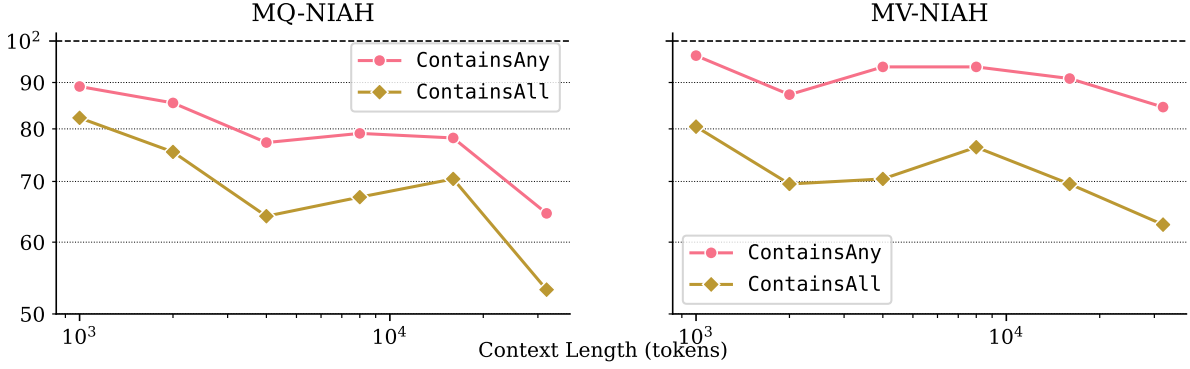


Figure S4: Comparison of metrics for aggregation subtasks in VTC-Retrieval: MV-NIAH (1 key, 2 values) and MQ-NIAH (2 keys, 1 value each), evaluated with Qwen2.5-VL-7B-Instruct.

Subtask	Prompt	VTC-Retrieval (containsAll ↑)				
		1k	2k	4k	8k	16k
S	CP	51.82	50.91	50.00	16.36	1.82
	QA	30.91	32.73	25.45	0.00	0.00
MK	CP	40.00	50.91	51.82	12.73	0.91
	QA	23.64	26.36	17.27	0.00	0.00
MQ	CP	21.82	18.18	19.09	6.82	0.00
	QA	7.73	11.36	6.82	0.00	0.00
MV	CP	20.91	24.09	17.73	14.55	1.36
	QA	10.45	16.82	9.09	1.36	0.00

Table S6: Performance of DeepSeek-OCR against different prompts—CP (completion, e.g., *one of the magic numbers of long-context is ...*), or QA (question-answering, e.g., *What is one of the magic numbers of long-context?*).

style, and background color. In particular, Glyph reports its lack of generalization to rendering styles. To assess this, we applied a plain-text rendering style $\mathfrak{R}_{\text{plain}} = \mathfrak{R}_{\text{chat}} \cup \mathfrak{R}(\text{bg-color} = \text{white})$ to VTC-Memory and observed a notable performance gap shown in Table S5, thereby corroborating the Glyph’s claim that it is highly sensitive to rendering configurations.

F.6 Sensitivity to Prompt

OCR-specific VLMs may also be sensitive to the evaluation prompt. DeepSeek-OCR’s training pipeline does not include a supervised-finetuning stage (Wei et al., 2025), resulting in a lack of instruction-following abilities. We used a completion prompt instead of question-answering, which showed substantial improvements across all subtasks in VTC-Retrieval (Table S6).

G Gradient-based Learning for VTC-Reasoning Enhancement

Model	Retrieval	Reasoning	Memory	Overall
Qwen3-VL-8B	89.00	11.50	33.67	45.73
+ VTC-GSPO	91.38 ↑2.38	75.13 ↑63.63	33.50 ↓0.17	69.68 ↑23.95

Table S7: Performance (%) on VTCBench-Wild.

To mitigate associative reasoning refusals in VTC, we utilized Gemini-3.1-Pro to synthesize 200 associative reasoning data pairs, which were subsequently subjected to manual verification to ensure factual accuracy. These associative pairs span ten diverse domains: Geography & Bizarre Landmarks, Dietary Restrictions & Specific Foods, Historical/Cultural Figures & Masterpieces, Professions & Exclusive Tools, Diseases & Specific Symptoms, Hardcore Physics & Chemistry Phenomena, Historical Events & Exclusive Symbols, Niche Biosphere & Unique Ecology, Finance/Economics & Extreme Markets and Niche Sports & Exclusive Terminology. The detailed generation prompts are provided in the Prompt section. Using these data, we construct approximately 1.5k training examples for RL from the 200 associated pairs. We then apply Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025) to fine-tune the Qwen3-VL-8B model. The comparative results are presented in Table S7. The refusal rate on VTCBench-Wild decreases dramatically from 37% to 0.2%, and associative reasoning improves substantially. The post-training evaluation shows a marked improvement in VTC performance, further highlighting the promise of the VTC paradigm.

H Prompts

VTC-Retrieval

{haystack} {question}

VTC-Reasoning

{haystack} Answer a question based on the above book snippet. Your answer should be short and based on either explicitly stated facts or strong, logical inferences. Return only the final answer with no additional explanation or reasoning. {question}

VTC-Memory

{haystack} Based on the above context, write an answer in the form of a short phrase for the following question. Answer with exact words from the context whenever possible. {question}

VTC-Wild-Retrieval

{haystack} Answer a question based on the above book snippet. Your answer should be short and based on either explicitly stated facts or strong, logical inferences. Return only the final answer with no additional explanation or reasoning. Question: {question}

VTC-Wild-Reasoning

{haystack} Answer a question based on the above book snippet. Some special magic numbers are hidden within the following text. Make sure to memorize it. I will quiz you about the numbers afterwards. Question: {question}

VTC-Wild-Memory

{haystack} Based on the above context, write an answer in the form of a short phrase for the following question. Answer with exact words from the context whenever possible. Question: {question}

LLM Judge

Please as a grading expert, judge whether the final answers given by the candidates below are consistent with the standard answers, that is, whether the candidates answered correctly. Here are some evaluation criteria: 1. Please refer to the given standard answer. You don't need to re-generate the answer to the question because the standard answer has been given. You only need to judge whether the candidate's answer is consistent with the standard answer according to the form of the question. **THE STANDARD ANSWER IS ALWAYS CORRECT AND THE QUESTION IS PERFECTLY VALID. NEVER QUESTION THEM.** 2. **ONLY** compare the **FINAL ANSWER** - **COMPLETELY IGNORE** any potential errors in the **REASONING PROCESSES**. 3. Some answers may be expressed in different ways, such as some answers may be a mathematical expression, some answers may be a textual description, as long as the meaning expressed is the same. Before making a judgment, please understand the question and the standard answer first, and then judge whether the candidate's answer is correct. If the standard answer does not specify a unit, but the candidate's answer includes a unit that is correct for the value given, consider it correct. 4. Some answers may consist of multiple items, such as multiple-choice questions, multiple-select questions, fill-in-the-blank questions, etc. Regardless of the question type, the final answer will be considered correct as long as it matches the standard answer, regardless of whether the reasoning process is correct. For multiple-select questions and multi-blank fill-in-the-blank questions, all corresponding options or blanks must be answered correctly and match the standard answer exactly to be deemed correct. 5. If the prediction is given with `\boxed{ {} }`, please ignore the `\boxed{ {} }` and only judge whether the candidate's answer is consistent with the standard answer. 6. If the candidate's answer is invalid (e.g., incomplete (cut off mid-response), lots of unnormal repetitive content, or irrelevant to the question, saying it can't answer the question because some irresistible factors, like ethical issues, no enough information, etc.), select option C (INVALID). Please judge whether the following answers are consistent with the standard answer based on the above criteria. Grade the predicted answer of this new question as one of: A: CORRECT B: INCORRECT C: INVALID Just return the letters "A", "B", or "C", with no text around it. Here is your task. Don't apologize or correct yourself if there was a mistake; we are just trying to grade the answer.\n <Original Question Begin>: {question}\n <Original Question End>\n <Standard Answer Begin>: {gold_answer}\n <Standard Answer End>\n <Candidate's Answer Begin>: {prediction}\n <Candidate's Answer End>\n Judging the correctness of the candidate's answer:

Prompt for Generating Latent Associative Reasoning Data Pairs

You are a rigorous AI algorithm researcher. I need you to help me construct a batch of high-quality datasets designed to evaluate the "Latent Associative Reasoning" capabilities of Large Language Models (LLMs) in long-context scenarios.

[Core Task]

Please generate 200 pairs of "Question and Hidden Fact (Needle)" data points. You need to leverage real-world common sense, interdisciplinary expertise, or historical allusions to establish highly profound implicit mappings between the "Query Keyword (W_q)" and the "Hidden Fact Keyword (W_n)".

[Strict Construction Rules Part]

[Domain Diversity Requirements Part]

[Reference Examples Part]

Take a deep breath, strictly follow the rules and the 10 domain distributions above, and output the 200 high-quality data pairs (Numbered 1 to 200) for me.

Strict Construction Rules for Generating Associative Reasoning Data Pairs

[Strict Construction Rules]

1. **Zero Lexical Overlap (Absolute Red Line):** The "Query Keyword (W_q)" in the question and the "Fact Keyword (W_n)" in the hidden fact must be completely different. There must be absolutely no literal or morphological word overlap (e.g., no shared root words)!
2. **Logical Uniqueness (Exclusivity):** W_n must be able to uniquely reverse-infer W_q with a very high probability. It cannot be a broad or generic trait. For example: W_n "cutting meat with a knife" cannot uniquely deduce W_q "butcher" (a chef does this too); but W_n "throwing a red and white Poké Ball" can uniquely deduce W_q "Pokémon Trainer".
3. **Character Placeholder:** The question must follow a unified format asking about a character (e.g., "Which character..."). The hidden fact sentences must exclusively use the exact placeholder '[CHAR]' to refer to this person.
4. **Two Sentence Structures:** For each hidden fact, you must provide two syntactic variations:
 - **Default Order:** In the logical flow of the sentence, the placeholder '[CHAR]' must appear *before* the fact keyword W_n .
 - **Inverted Order:** In the logical flow of the sentence, the fact keyword W_n must appear *before* the placeholder '[CHAR]' (this can be achieved via passive voice, sentence inversion, or fronted modifiers).
5. **Natural Context:** The sentences must be fluent and natural, fitting everyday dialogue or fictional narrative styles. Avoid stiff, expository writing. Do not add artificial transitional words like "In fact," or "In summary," at the beginning or end.

Domain Diversity Requirements for Generating Associative Reasoning Data Pairs

[Domain Diversity Requirements]

Please distribute the generation evenly across the following 10 interdisciplinary/vertical domains (generate 2 pairs per domain, totaling 20 pairs):

1. **Geography & Bizarre Landmarks** (e.g., France -> The Louvre; Egypt -> The Sphinx)
2. **Dietary Restrictions & Specific Foods** (e.g., Lactose Intolerance -> Severe diarrhea after drinking milk; Gout -> Absolutely avoiding seafood and crabs)
3. **Historical/Cultural Figures & Masterpieces** (e.g., Da Vinci -> The Mona Lisa; Christopher Nolan -> The spinning top in Inception)
4. **Professions & Exclusive Tools** (e.g., Programmer -> Cherry MX Red mechanical keyboard; Doctor -> Stethoscope/Scalpel)
5. **Diseases & Specific Symptoms** (e.g., Insomnia -> Staring at the ceiling at 4:30 AM; Parkinson's -> Severe resting tremors in the hands)
6. **Hardcore Physics & Chemistry Phenomena** (e.g., Pure Sodium Metal -> Violent explosion when thrown into water; Relativity -> Time dilation and length contraction)
7. **Historical Events & Exclusive Symbols** (e.g., French Revolution -> The Guillotine; Trojan War -> Giant wooden horse)
8. **Niche Biosphere & Unique Ecology** (e.g., Chameleon -> Changing color to camouflage; Anglerfish -> Glowing lure on its head)
9. **Finance/Economics & Extreme Markets** (e.g., Hyperinflation -> Pushing wheelbarrows of cash to buy a loaf of bread; Bear Market -> Stock screens flashing red/circuit breakers)
10. **Niche Sports & Exclusive Terminology** (e.g., Curling -> Frantically sweeping the ice with brooms; Chess -> Castling)

Reference Examples for Generating Associative Reasoning Data Pairs

[Reference Examples (Please strictly align with this output format)]

Data Pair 1

- **Domain:** Hardcore Physics & Chemistry Phenomena
- **Question (Contains W_q):** Which character stole a large chunk of high-purity **sodium metal** from the laboratory?
- **Default Order Fact (W_n follows '[CHAR]')**: '[CHAR]' had just tossed that substance into a glass basin filled with water when it triggered a deafening, violent explosion.
- **Inverted Order Fact (W_n precedes '[CHAR]')**: The deafening, violent explosion in the water basin was triggered right after that substance was carelessly tossed in by '[CHAR]'.
- **Associative Logic:** Exploding violently upon contact with water (W_n) is an exclusive, highly specific chemical reaction unique to the alkali metal sodium (W_q).

Data Pair 2

- **Domain:** Diseases & Specific Symptoms
- **Question (Contains W_q):** Which character has suffered from severe **asthma** since childhood?
- **Default Order Fact (W_n follows '[CHAR]')**: '[CHAR]' must immediately inhale from an albuterol inhaler to catch a breath after every intense workout.
- **Inverted Order Fact (W_n precedes '[CHAR]')**: The albuterol inhaler is a life-saving medication that must be immediately used by '[CHAR]' to relieve suffocation after intense workouts.
- **Associative Logic:** The albuterol inhaler (W_n) is the absolute exclusive medication clinically used for emergency relief of asthma (W_q) attacks.

Model	r_{VTC}	1k	2k	4k	8k	16k	32k
Qwen3-8B	NA	98.18	100.00	96.36	97.73	95.45	94.55
Qwen3-VL-8B (LLM)	NA	100.00	100.00	100.00	100.00	100.00	100.00
GPT-5 (LLM)	NA	100.00	100.00	100.00	95.45	97.27	99.09
InternVL3.5-8B	1.56	32.73	23.64	36.36	21.82	20.91	12.73
InternVL3.5-38B	1.56	22.73	20.00	16.36	22.73	15.45	24.55
Gemini-2.5-Pro	1.98	100.00	50.00	55.45	54.55	40.91	41.82
Gemma-3-27B-it	2.00	89.09	84.55	85.45	87.27	OOM	OOM
GLM-4.1V-9B-Thinking	2.00	26.36	16.36	11.82	14.55	3.64	2.73
Glyph	2.00	92.73	83.64	90.00	83.64	94.55	85.45
GPT-5	2.00	76.36	76.36	71.82	61.82	62.73	60.00
InternVL3.5-8B	2.00	19.09	22.73	10.91	13.64	12.73	10.00
InternVL3.5-38B	2.00	17.27	16.36	12.73	13.64	9.09	11.82
Kimi-VL-A3B-Instruct	2.00	98.18	67.27	59.09	55.45	61.82	66.36
Qwen2.5-VL-7B-Instruct	2.00	84.55	77.27	82.73	84.55	78.18	84.55
Qwen2.5-VL-72B-Instruct	2.00	92.73	94.55	95.45	97.27	91.82	98.18
Qwen3-VL-8B-Instruct	2.00	99.09	89.09	93.64	90.91	90.00	89.09
Qwen3-VL-235B-A22B-Instruct	2.00	97.27	82.73	92.73	88.18	89.09	74.55
Qwen3-VL-8B-Instruct	2.55	97.27	72.73	75.45	74.55	56.36	59.09
Qwen3-VL-235B-A22B-Instruct	2.55	98.18	89.09	89.09	84.55	59.09	74.55
DeepSeek-OCR (Completion)	3.12	51.82	50.91	50.00	16.36	1.82	OOM
DeepSeek-OCR (QA)	3.12	30.91	32.73	25.45	0.00	0.00	OOM
GPT-5	3.17	50.00	48.18	32.73	36.36	33.64	26.36
Gemma-3-27B-it	8.00	20.00	19.09	20.91	19.09	20.00	19.09

Table S8: VTC-Retrieval S-NIAH accuracy.

Model	r_{VTC}	1k	2k	4k	8k	16k	32k
Qwen3-8B	NA	98.18	97.27	90.91	92.73	91.82	95.45
Qwen3-VL-8B (LLM)	NA	100.00	100.00	100.00	100.00	100.00	100.00
GPT-5 (LLM)	NA	100.00	100.00	99.09	98.18	97.27	98.18
InternVL3.5-8B	1.56	28.18	29.09	22.73	19.09	10.91	10.00
InternVL3.5-38B	1.56	26.36	32.73	29.09	24.55	20.00	20.00
Gemini-2.5-Pro	1.98	100.00	53.64	49.09	49.09	30.91	35.45
Gemma-3-27B-it	2.00	88.18	91.82	88.18	81.82	OOM	OOM
GLM-4.1V-9B-Thinking	2.00	10.00	13.64	6.36	10.91	8.18	3.64
Glyph	2.00	92.73	81.82	83.64	81.82	82.73	78.18
GPT-5	2.00	85.45	71.82	68.18	61.36	59.09	49.09
InternVL3.5-8B	2.00	12.73	19.09	7.27	11.82	6.36	8.18
InternVL3.5-38B	2.00	27.27	16.36	12.73	15.45	14.55	12.73
Kimi-VL-A3B-Instruct	2.00	84.55	68.18	48.18	44.55	47.27	44.55
Qwen2.5-VL-7B-Instruct	2.00	93.64	80.00	81.82	79.09	88.18	76.36
Qwen2.5-VL-72B-Instruct	2.00	89.09	80.00	87.27	90.91	86.36	80.91
Qwen3-VL-8B-Instruct	2.00	93.64	84.55	85.45	90.00	90.91	85.86
Qwen3-VL-235B-A22B-Instruct	2.00	100.00	92.73	99.09	96.36	92.12	85.18
Qwen3-VL-8B-Instruct	2.55	90.91	78.18	69.09	69.09	55.45	66.36
Qwen3-VL-235B-A22B-Instruct	2.55	100.00	84.55	89.09	83.64	62.73	52.73
DeepSeek-OCR (Completion)	3.12	40.00	50.91	51.82	12.73	0.91	OOM
DeepSeek-OCR (QA)	3.12	23.64	26.36	17.27	0.00	0.00	OOM
GPT-5	3.17	44.55	42.73	34.55	32.73	28.18	21.82
Gemma-3-27B-it	8.00	28.18	26.36	32.73	22.73	10.91	15.45

Table S9: VTC-Retrieval MK-NIAH accuracy.

Model	r_{VTC}	1k	2k	4k	8k	16k	32k
Qwen3-8B	NA	99.09	100.00	100.00	100.00	99.09	96.82
Qwen3-VL-8B (LLM)	NA	99.09	96.36	96.82	98.64	98.18	98.64
GPT (LLM)	NA	100.00	98.18	99.09	90.91	80.91	96.36
InternVL3.5-8B	1.56	11.82	15.45	16.82	21.82	15.45	15.45
InternVL3.5-38B	1.56	17.27	15.91	19.55	20.45	18.64	18.64
Gemini-2.5-Pro	1.98	100.00	50.00	55.45	54.55	40.00	41.82
Gemma-3-27B-it	2.00	90.91	83.64	83.64	67.27	OOM	OOM
GLM-4.1V-9B-Thinking	2.00	17.73	9.09	5.00	4.09	0.91	2.27
Glyph	2.00	96.36	85.45	86.36	76.82	76.82	75.45
GPT-5	2.00	84.09	75.00	72.27	64.55	62.73	61.82
InternVL3.5-8B	2.00	12.27	11.36	10.45	8.18	8.18	8.64
InternVL3.5-38B	2.00	20.45	14.09	11.82	14.55	7.73	8.18
Kimi-VL-A3B-Instruct	2.00	77.73	41.82	40.00	33.64	32.27	28.64
Qwen2.5-VL-7B-Instruct	2.00	80.45	69.55	70.45	76.36	69.55	62.73
Qwen2.5-VL-72B-Instruct	2.00	83.64	80.45	83.64	73.64	75.45	59.55
Qwen3-VL-8B-Instruct	2.00	93.18	90.00	88.18	85.45	82.27	69.55
Qwen3-VL-235B-A22B-Instruct	2.00	96.36	86.36	92.27	93.18	86.82	76.68
Qwen3-VL-8B-Instruct	2.55	93.64	62.73	55.91	46.36	38.64	38.64
Qwen3-VL-235B-A22B-Instruct	2.55	96.82	81.82	74.55	70.91	56.82	44.55
DeepSeek-OCR (Completion)	3.12	20.91	24.09	17.73	14.55	1.36	OOM
DeepSeek-OCR (QA)	3.12	10.45	16.82	9.09	1.36	0.00	OOM
GPT-5	3.17	41.36	32.27	32.73	34.55	30.45	29.09
Gemma-3-27B-it	8.00	23.18	14.55	20.45	17.27	13.18	9.55

Table S10: VTC-Retrieval MV-NIAH accuracy.

Model	r_{VTC}	1k	2k	4k	8k	16k	32k
Qwen3-8B	NA	100.00	98.18	100.00	98.18	100.00	95.45
Qwen3-VL-8B (LLM)	NA	98.64	98.64	98.18	99.55	98.64	99.09
GPT-5 (LLM)	NA	99.09	96.36	95.45	91.82	88.18	93.64
InternVL3.5-8B	1.56	21.82	19.09	18.64	20.45	16.82	11.82
InternVL3.5-38B	1.56	20.00	16.36	22.27	17.27	16.82	12.73
Gemini-2.5-Pro	1.98	100.00	60.91	58.18	60.00	42.73	43.18
Gemma-3-27B-it	2.00	87.27	79.55	71.82	61.36	OOM	OOM
GLM-4.1V-9B-Thinking	2.00	14.55	6.82	7.73	5.00	2.73	1.36
Glyph	2.00	84.09	74.55	79.55	71.82	67.73	63.64
GPT-5	2.00	81.82	75.00	72.27	63.18	58.18	57.73
InternVL3.5-8B	2.00	15.91	13.18	10.91	13.18	6.82	5.00
InternVL3.5-38B	2.00	15.45	11.82	10.00	8.18	7.73	6.82
Kimi-VL-A3B-Instruct	2.00	87.73	53.18	48.64	43.18	35.91	32.27
Qwen2.5-VL-7B-Instruct	2.00	82.27	75.45	64.09	67.27	70.45	53.18
Qwen2.5-VL-72B-Instruct	2.00	90.45	83.18	84.55	84.55	75.45	67.27
Qwen3-VL-8B-Instruct	2.00	95.00	90.00	82.27	89.55	84.55	73.86
Qwen3-VL-235B-A22B-Instruct	2.00	95.00	92.27	95.00	90.91	96.82	88.96
Qwen3-VL-8B-Instruct	2.55	95.00	75.91	65.45	55.91	47.27	49.09
Qwen3-VL-235B-A22B-Instruct	2.55	95.91	92.27	89.55	80.45	75.91	62.34
DeepSeek-OCR (Completion)	3.12	21.82	18.18	19.09	6.82	0.00	OOM
DeepSeek-OCR (QA)	3.12	7.73	11.36	6.82	0.00	0.00	OOM
GPT-5	3.17	46.36	31.36	28.18	28.18	24.09	24.09
Gemma-3-27B-it	8.00	16.82	11.82	15.91	15.00	13.64	11.36

Table S11: VTC-Retrieval MQ-NIAH accuracy.

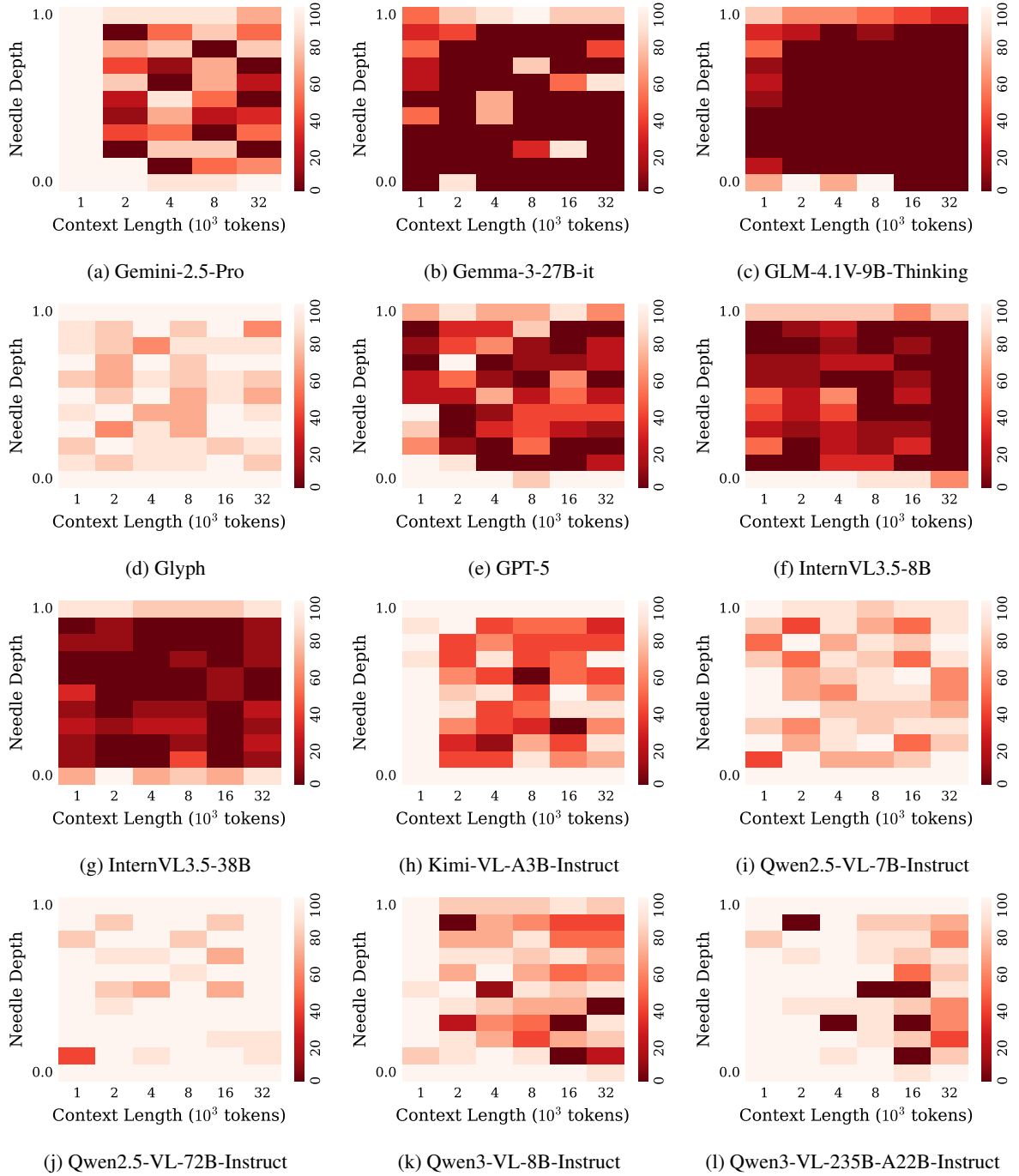


Figure S5: VTC-Retrieval S-NIAH (rendering $\mathfrak{R}_{\text{plain}}$) accuracy with respect to needle placement depth and context length.

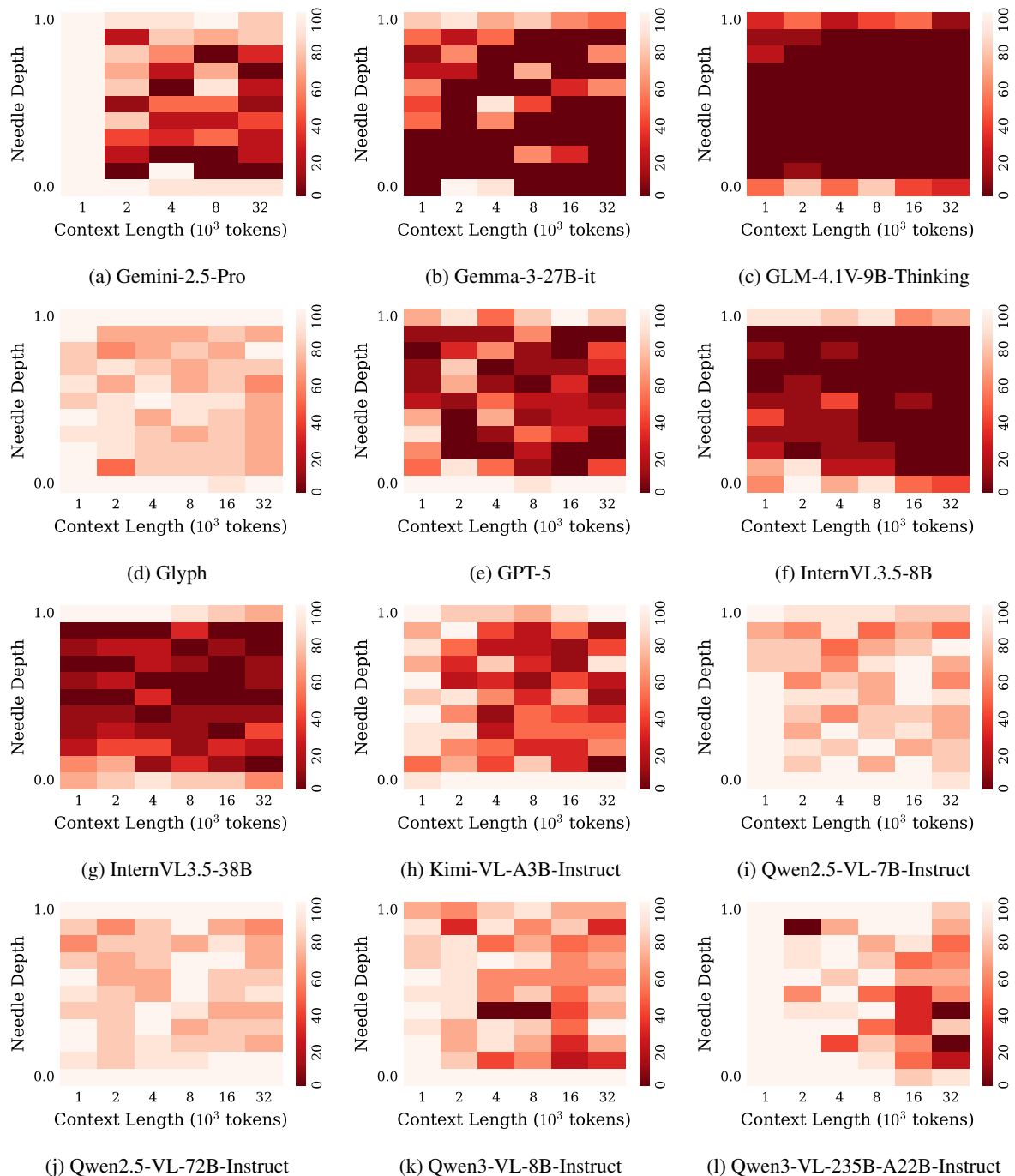


Figure S6: VTC-Retrieval MK-NIAH (rendering $\mathfrak{R}_{\text{plain}}$) accuracy with respect to needle placement depth and context length.

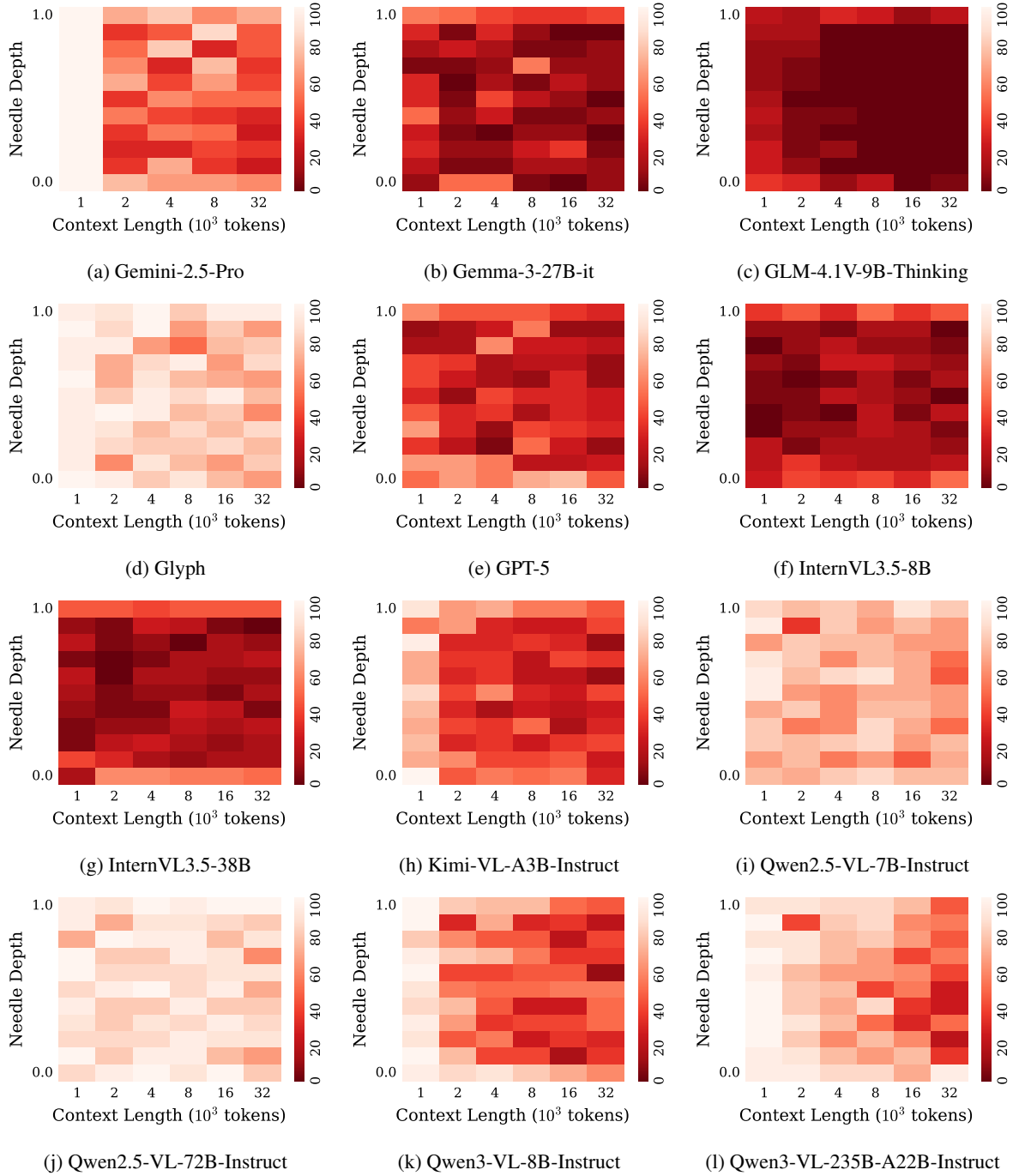


Figure S7: VTC-Retrieval MV-NIAH (rendering $\mathfrak{R}_{\text{plain}}$) accuracy with respect to needle placement depth and context length.

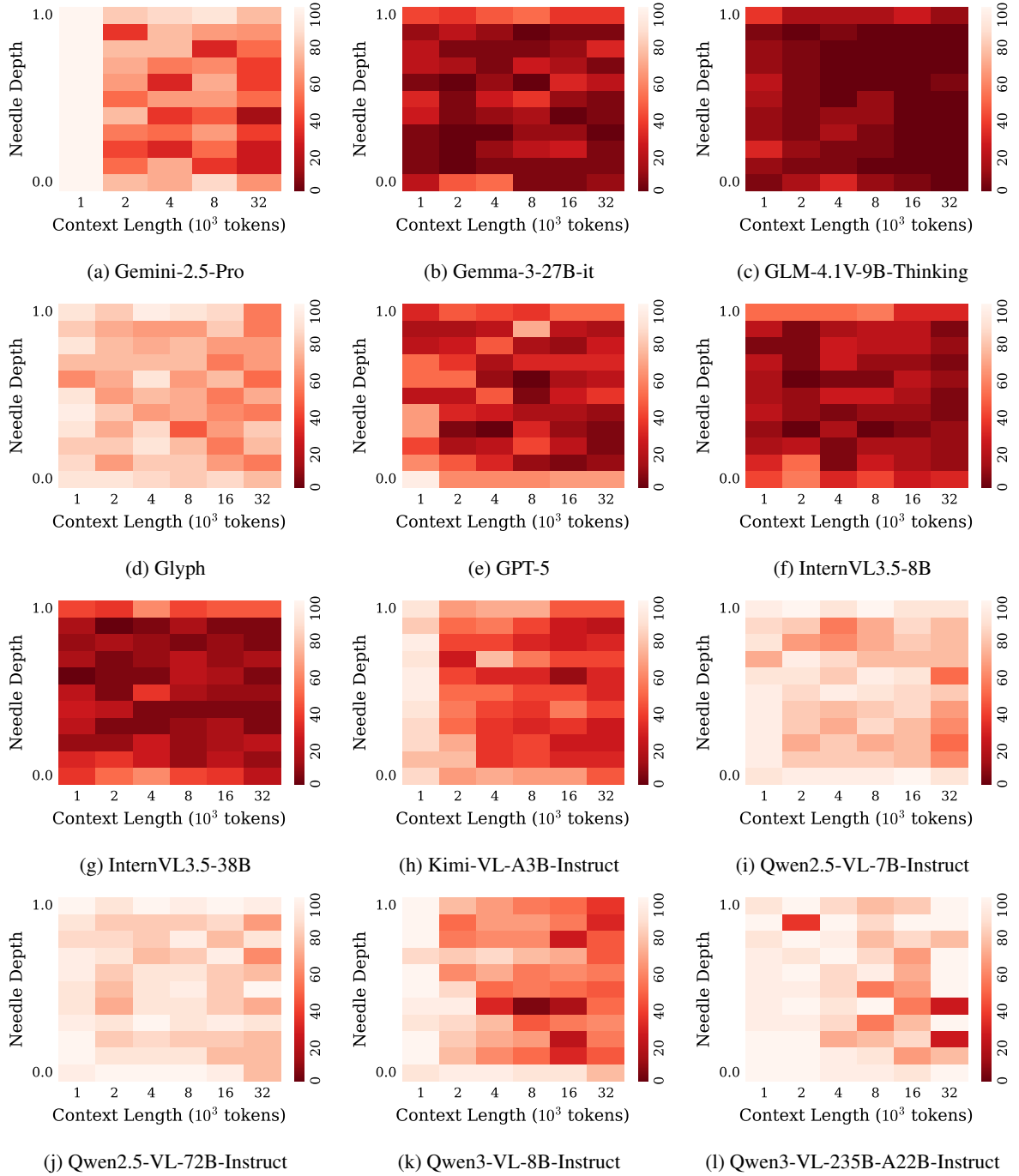


Figure S8: VTC-Retrieval MQ-NIAH (rendering $\mathfrak{R}_{\text{plain}}$) accuracy with respect to needle placement depth and context length.

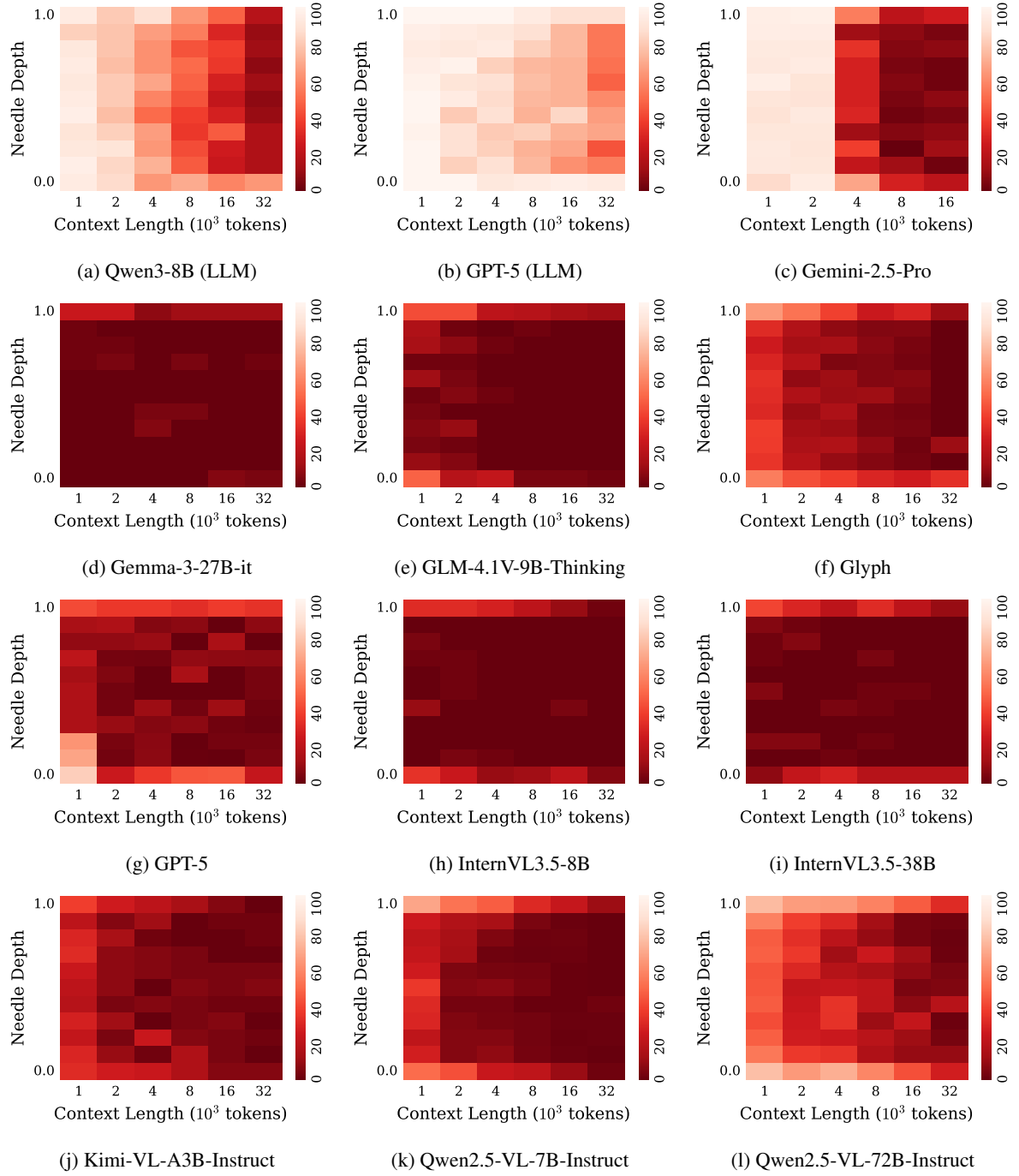


Figure S9: VTC-Reasoning (rendering $\mathcal{R}_{\text{plain}}$) accuracy with respect to needle placement depth and context length.