KIMI-VL TECHNICAL REPORT

Kimi Team

ABSTRACT

We present **Kimi-VL**, an efficient open-source Mixture-of-Experts (MoE) vision-language model (VLM) that offers **advanced multimodal reasoning, long-context understanding, and strong agent capabilities**—all while activating only **2.8B** parameters in its language decoder (Kimi-VL-A3B).

Kimi-VL demonstrates strong performance across challenging domains: as a general-purpose VLM, Kimi-VL excels in multi-turn agent tasks (*e.g.,* OSWorld), matching flagship models. Furthermore, it exhibits remarkable capabilities across diverse challenging vision language tasks, including college-level image and video comprehension, OCR, mathematical reasoning, multi-image understanding. In comparative evaluations, it effectively competes with cutting-edge efficient VLMs such as GPT-4o-mini, Qwen2.5-VL-7B, and Gemma-3-12B-IT, while surpassing GPT-4o in several key domains.

Kimi-VL also advances in processing long contexts and perceiving clearly. With a 128K extended context window, Kimi-VL can process diverse long inputs, achieving impressive scores of 64.5 on LongVideoBench and 35.1 on MMLongBench-Doc. Its native-resolution vision encoder, MoonViT, further allows it to see and understand ultra-high-resolution visual inputs, achieving 83.2 on InfoVQA and 34.5 on ScreenSpot-Pro, while maintaining lower computational cost for common tasks.

Building upon Kimi-VL, we introduce an advanced long-thinking variant: **Kimi-VL-Thinking-2506**. Developed through long chain-of-thought (CoT) supervised fine-tuning (SFT) and reinforcement learning (RL), the latest model exhibits strong long-horizon reasoning capabilities (64.0 on MMMU, 46.3 on MMMU-Pro, 56.9 on MathVision, 80.1 on MathVista, 65.2 on VideoMMMU) while obtaining robust general abilities (84.4 on MMBench, 83.2 on V* and 52.8 on ScreenSpot-Pro). With only around 3B activated parameters, it sets a new standard for efficient yet capable multimodal *thinking* models. Code and models are publicly accessible at `https://github.com/MoonshotAI/Kimi-VL`.

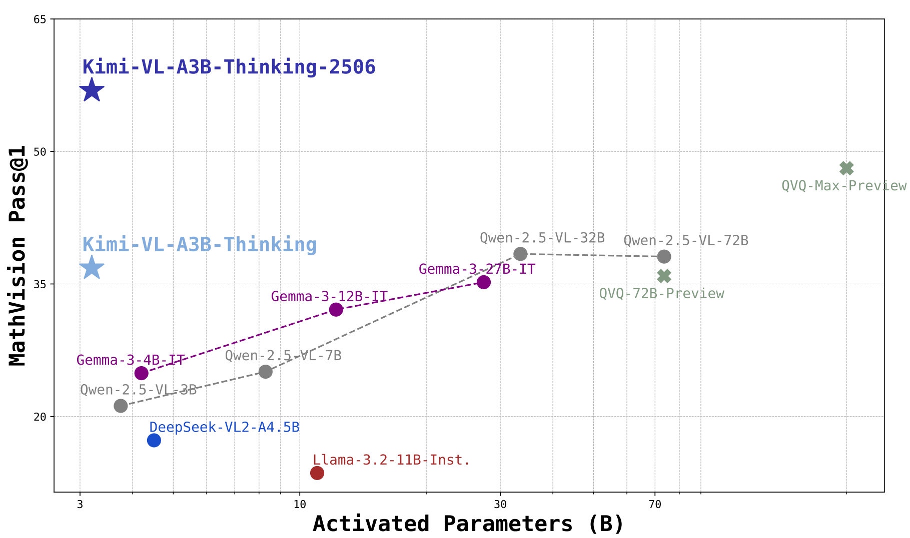


Figure 1: Comparison between **Kimi-VL-Thinking-2506** and frontier open-source VLMs, including short-thinking VLMs (*e.g.* Gemma-3 series, Qwen2.5-VL series) and long-thinking VLMs (QVQ-72B/Max-Preview), on MathVision benchmark. Our model achieves strong multimodal reasoning with just 2.8B LLM activated parameters.

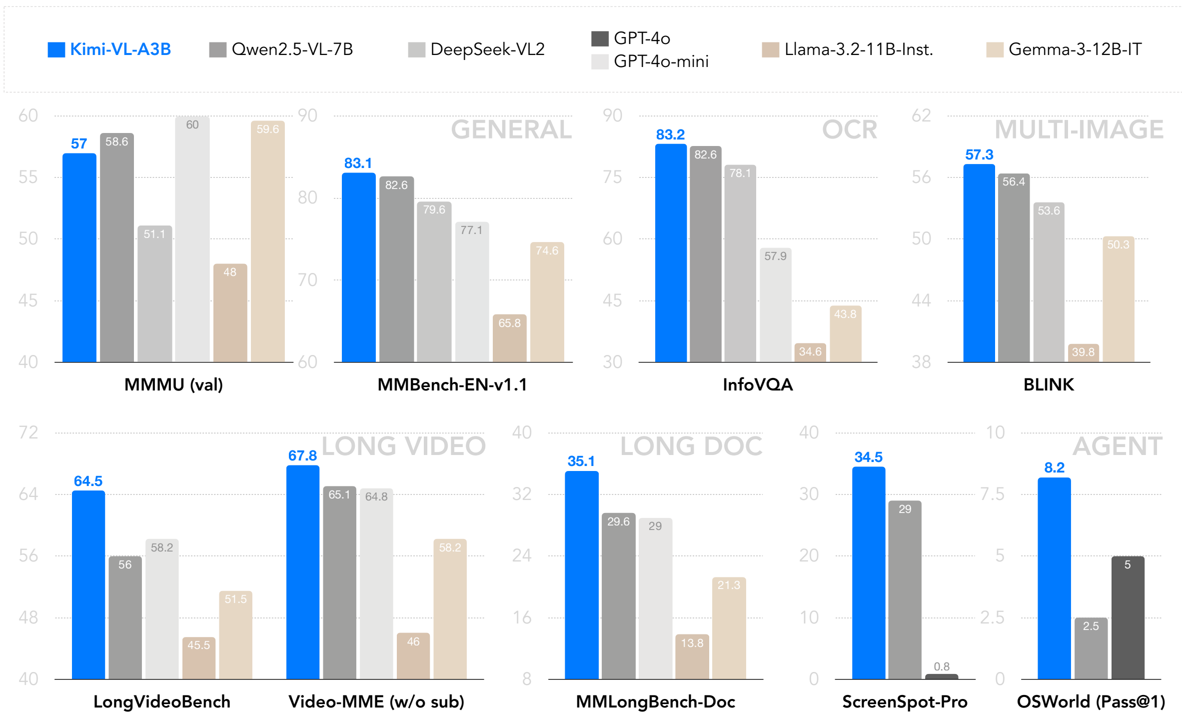


Figure 2: Highlights of **Kimi-VL** performance for a wide range of benchmarks like, general benchmarks (MMMU, MMBench), OCR (InfoVQA), multi-image (BLINK), long video (LongVideoBench, Video-MME), long document (MMLongBench-Doc), and agent (ScreenSpot-Pro and OSWorld). Detailed results are presented in Table 3.

1 Introduction

With the rapid advancement of artificial intelligence, human expectations for AI assistants have transcended traditional language-only interactions, increasingly aligning with the inherently multimodal nature of our world. To better understand and interact with these expectations, new generations of natively multimodal models, such as GPT-4o (OpenAI et al. 2024) and Google Gemini (Gemini Team et al. 2024), have emerged with the capability to seamlessly perceive and interpret visual inputs alongside language processing. Most recently, advanced multimodal models, pioneered by OpenAI o1 series (OpenAI 2024) and Kimi k1.5 (K. Team et al. 2025), have further pushed these boundaries by incorporating deeper and longer reasoning on multimodal inputs, thereby tackling more complex problems in the multimodal domain.

Nevertheless, development in large VLMs in the open-source community has significantly lagged behind their language-only counterparts, particularly in aspects of scalability, computational efficiency, and advanced reasoning capabilities. While language-only model DeepSeek R1 (DeepSeek-AI, D. Guo, et al. 2025) has already leveraged the efficient and more scalable mixture-of-experts (MoE) architecture and facilitated sophisticated long chain-of-thought (CoT) reasoning, most recent open-source VLMs, *e.g.* Qwen2.5-VL (Bai et al. 2025) and Gemma-3 (Gemma Team et al. 2025), continue to rely on dense architectures and do not support long-CoT reasoning. Early explorations into MoE-based vision-language models, such as DeepSeek-VL2 (Zhiyu Wu et al. 2024) and Aria (D. Li et al. 2024), exhibit limitations in other crucial dimensions. Architecturally, both models still adopt relatively traditional fixed-size vision encoders, hindering their adaptability to diverse visual inputs. From a capability perspective, DeepSeek-VL2 supports only a limited context length (4K), while Aria falls short in fine-grained visual tasks. Additionally, neither of them supports long-thinking abilities. Consequently, there remains a pressing need for an open-source VLM that effectively integrates structural innovation, stable capabilities, and enhanced reasoning through long-thinking.

In light of this, we present **Kimi-VL**, a vision-language model for the open-source community. Structurally, Kimi-VL consists of our Moonlight (J. Liu et al. 2025a) MoE language model with only **2.8B** activated (16B total) parameters, paired with a 400M native-resolution MoonViT vision encoder. In terms of capability, as illustrated in Figure 2, Kimi-VL can robustly handle diverse tasks (fine-grained perception, math, college-level problems, OCR, agent, *etc.*) across a broad spectrum of input forms (single-image, multi-image, video, long-document, *etc.*). Specifically, it features the following exciting abilities:

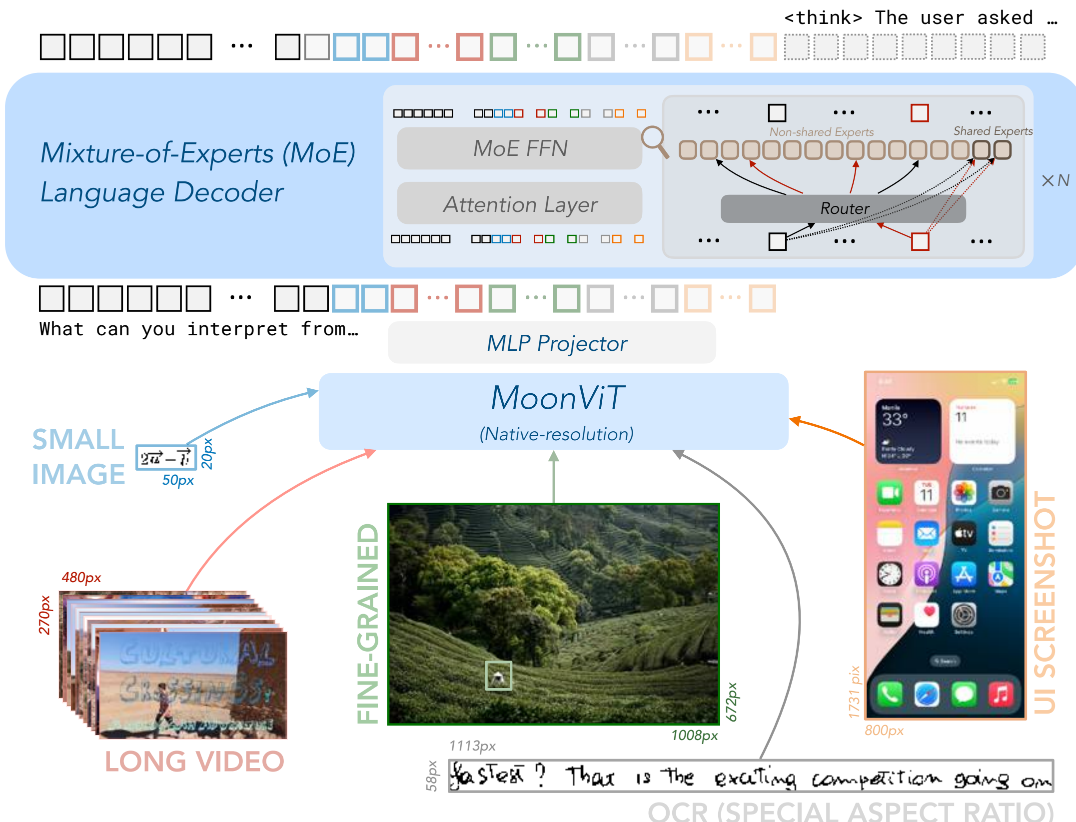


Figure 3: The model architecture of Kimi-VL and Kimi-VL-Thinking, consisting of a MoonViT that allows native-resolution images, an MLP projector, and a Mixture-of-Experts (MoE) language decoder.

1) Kimi-VL is smart: it has comparable text ability against efficient pure-text LLMs; without long thinking, Kimi-VL is already competitive in multimodal reasoning and multi-turn agent benchmarks, *e.g.,* MMMU, MathVista, OSWorld.

2) Kimi-VL processes long: it effectively tackles long-context understanding on various multimodal inputs within its 128K context window, far ahead of similar-scale competitors on long video benchmarks and MMLongBench-Doc.

3) Kimi-VL perceives clear: it shows all-round competitive ability over existing efficient dense and MoE VLMs in various vision-language scenarios: visual perception, visual world knowledge, OCR, high-resolution OS screenshot, *etc.*

Furthermore, with long-CoT activation and reinforcement learning (RL), we introduce the long-thinking version of Kimi-VL, **Kimi-VL-Thinking**, which further substantially improves performance on more complex multimodal reasoning scenarios. Despite its small scale, Kimi-VL-Thinking offers compelling performance on hard reasoning benchmarks (*e.g.,* MMMU, MathVision, MathVista), outperforming many state-of-the-art VLMs with even larger sizes. We further release and improved version of the thinking model, **Kimi-VL-Thinking-2506**. The improved version has even better performance on these reasoning benchmarks while retaining or improving on common visual perception and understanding scenarios, *e.g.* high-resolution perception (V*), OS grounding, video and long document understanding.

2 Approach

2.1 Model Architecture

The architecture of Kimi-VL consists of three parts: a native-resolution vision encoder (MoonViT), an MLP projector, and an MoE language model, as depicted in Figure 3. We introduce each part in this section.

MoonViT: A Native-resolution Vision Encoder We design MoonViT, the vision encoder of Kimi-VL, to natively process images at their varying resolutions, eliminating the need for complex sub-image splitting and splicing operations, as employed in LLaVA-OneVision (B. Li et al. 2024). We incorporate the packing method from NaViT (Dehghani et al. 2023), where images are divided into patches, flattened, and sequentially concatenated into 1D sequences. These

Figure 4: The pre-training stages of Kimi-VL consume a total of 4.4T tokens after text-only pre-training of its language model. To preserve text abilities, all stages that update the language model are joint training stages.

preprocessing operations enable MoonViT to share the same core computation operators and optimization as a language model, such as the variable-length sequence attention mechanism supported by FlashAttention (Dao et al. 2022), ensuring non-compromised training throughput for images of varying resolutions.

MoonViT is initialized from and continually pre-trained on SigLIP-SO-400M (Zhai et al. 2023), which originally employs learnable fixed-size absolute positional embeddings to encode spatial information. While we interpolate these original position embeddings to better preserve SigLIP's capabilities, these interpolated embeddings become increasingly inadequate as image resolution increases. To address this limitation, we incorporate 2D rotary positional embedding (RoPE) (J. Su et al. 2023) across the height and width dimensions, which improves the representation of fine-grained positional information, especially in high-resolution images. These two positional embedding approaches work together to encode spatial information for our model and seamlessly integrate with the flattening and packing procedures. This integration enables MoonViT to efficiently process images of varying resolutions within the same batch. The resulting continuous image features are then forwarded to the MLP projector and, ultimately, to the MoE language model for subsequent training stages. In Kimi-VL-A3B-Thinking-2506, we further continually train this MoonViT to authentically encode up to 3.2 million pixels from a single image, 4 times compared to the original limit.

MLP Projector We employ a two-layer MLP to bridge the vision encoder (MoonViT) and the LLM. Specifically, we first use a pixel shuffle operation to compress the spatial dimension of the image features extracted by MoonViT, performing 2×2 downsampling in the spatial domain and correspondingly expanding the channel dimension. We then feed the pixel-shuffled features into a two-layer MLP to project them into the dimension of LLM embeddings.

Mixture-of-Experts (MoE) Language Model The language model of Kimi-VL utilizes our Moonlight model (J. Liu et al. 2025a), an MoE language model with 2.8B activated parameters, 16B total parameters, and an architecture similar to DeepSeek-V3 (DeepSeek-AI, A. Liu, et al. 2025). For our implementation, we initialize from an intermediate checkpoint in Moonlight's pre-training stage—one that has processed 5.2T tokens of pure text data and activated an 8192-token (8K) context length. We then continue pre-training it using a joint recipe of multimodal and text-only data totaling 2.3T tokens, as detailed in Sec. 2.3.

2.2 Muon Optimizer

We use an enhanced Muon optimizer (J. Liu et al. 2025b) for model optimization. Compared to the original Muon optimizer (Jordan et al. 2024), we add weight decay and carefully adjust the per-parameter update scale. Additionally, we develop a distributed implementation of Muon following the ZeRO-1 (Rajbhandari et al. 2020) optimization strategy, which achieves optimal memory efficiency and reduced communication overhead while preserving the algorithm's mathematical properties. This enhanced Muon optimizer is used throughout the entire training process to optimize all model parameters, including the vision encoder, the projector, and the language model.

2.3 Pre-Training Stages

As illustrated in Figure 4 and Table 1, after loading the intermediate language model discussed above, Kimi-VL's pre-training comprises a total of 4 stages consuming 4.4T tokens overall: first, standalone ViT training to establish a robust native-resolution visual encoder, followed by three joint training stages (pre-training, cooldown, and long-context activation) that simultaneously enhance the model's language and multimodal capabilities. The details are as follows.

ViT Training Stages The MoonViT is trained on image-text pairs, where the text components consist of a variety of targets: image alt texts, synthetic captions, grounding bboxes, and OCR texts. The training incorporates two objectives: a SigLIP (Zhai et al. 2023) loss $\mathcal{L}_{siglip}$ (a variant of contrastive loss) and a cross-entropy loss $\mathcal{L}_{caption}$ for caption generation conditioned on input images. Following CoCa's approach (J. Yu et al. 2022), the final loss function is

Table 1: Overview of training stages: data composition, token volumes, sequence lengths, and trainable components.

Stages	ViT Training	Joint Pre-training	Joint Cooldown	Joint Long-context
Data	Alt text Synthesis Caption Grounding OCR	+ Text, Knowledge Interleaving Video, Agent	+ High-quality Text High-quality Multimodal Academic Sources	+ Long Text Long Video Long Document
Tokens	2T + 0.1T	1.4T	0.6T	0.3T
Sequence length	8192	8192	8192	32768->131072
Training	ViT	ViT & LLM	ViT & LLM	ViT & LLM

Table 2: **Needle-in-a-Haystack (NIAH)** test on text/video haystacks, where needles are uniformly distributed at various positions within the haystack. We report recall accuracy across different haystack lengths up to 131,072 tokens (128K).

Haystack Length	$(0, 2048]$	$(2048, 4096]$	$(4096, 8192]$	$(8192, 16384]$	$(16384, 32768]$	$(32768, 65536]$	$(65536, 131072]$
- *text haystack*	100.0	100.0	100.0	100.0	100.0	100.0	87.0
- *video haystack*	100.0	100.0	100.0	100.0	100.0	100.0	91.7

formulated as $\mathcal{L} = \mathcal{L}_{siglip} + \lambda\mathcal{L}_{caption}$, where $\lambda = 2$. Specifically, the image and text encoders compute the contrastive loss, while the text decoder performs next-token prediction (NTP) conditioned on features from the image encoder. To accelerate training, we initialized both encoders with SigLIP SO-400M (Zhai et al. 2023) weights and implemented a progressive resolution sampling strategy to gradually allow larger size; the text decoder is initialized from a tiny decoder-only language model. During training, we observed an emergence in the caption loss while scaling up OCR data, indicating that the text decoder had developed some OCR capabilities. After training the ViT in the CoCa-alike stage with 2T tokens, we align the MoonViT to the MoE language model using another 0.1T tokens, where only MoonViT and MLP projector are updated. This alignment stage significantly reduces the initial perplexity of MoonViT embeddings in the language model, allowing a smoother joint pre-training stage as follows.

Joint Pre-training Stage In the joint pre-training stage, we train the model with a combination of pure text data (sampled from the same distribution as the initial language model) and a variety of multimodal data (as discussed in Sec. 3.1). We continue training from the loaded LLM checkpoint using the same learning rate scheduler, consuming an additional 1.4T tokens. The initial steps utilize solely language data, after which the proportion of multimodal data gradually increases. Through this progressive approach and the previous alignment stage, we observe that joint pre-training preserves the model's language capabilities while successfully integrating visual comprehension abilities.

Joint Cooldown Stage The stage following the pre-training stage is a multimodal cooldown phase, where the model is continue trained with high-quality language and multimodal datasets to ensure superior performance. For the language part, through empirical investigation, we observe that the incorporation of synthetic data during the cooling phase yields significant performance improvements, particularly in mathematical reasoning, knowledge-based tasks, and code generation. The general text components of the cooldown dataset are curated from high-fidelity subsets of the pre-training corpus. For math, knowledge, and code domains, we employ a hybrid approach: utilizing selected pre-training subsets while augmenting them with synthetically generated content. Specifically, we leverage existing mathematical knowledge and code corpora as source material to generate question-answer (QA) pairs through a proprietary language model, implementing rejection sampling techniques to maintain quality standards (Yue, Qu, et al. 2023; D. Su et al. 2024). These synthesized QA pairs undergo comprehensive validation before being integrated into the cooldown dataset. For the multimodal part, in addition to the two strategies as employed in text cooldown data preparation, *i.e.* question-answer synthesis and high-quality subset replay, to allow more comprehensive visual-centric perception and understanding (B. Li et al. 2024; Tong et al. 2024; J. Guo et al. 2024), we filter and rewrite a variety of academic visual or vision-language data sources to QA pairs. Unlike post-training stages, these language and multimodal QA pairs in the cooldown stage are only included for activating specific abilities and henceforth facilitating learning high-quality data, thus, we keep their ratio at a low portion to avoid overfitting these QA patterns. The joint cooldown stage significantly improves both language and multimodal abilities of the model.

Joint Long-context Activation Stage In the final pre-training stage, we extend the context length of the model from 8192 (8K) to 131072 (128K), with the inverse frequency of its RoPE (J. Su et al. 2023) embeddings reset from 50,000

Figure 5: The post-training stages of Kimi-VL and Kimi-VL-Thinking, including two stages of joint SFT in 32K and 128K context, and further long-CoT SFT and RL stages to activate and enhance long thinking abilities.

to 800,000. The joint long-context stage is conducted in two sub-stages, where each one extends the model's context length by four times. For data composition, we filter and upsample the ratio of long data to 25% in each sub-stage, while using the remaining 75% tokens to replay shorter data in its previous stage; our exploration confirms that this composition allows the model to effectively learn long-context understanding while maintaining short-context ability.

To allow the model to activate long-context abilities on both pure-text and multimodal inputs, the long data used in Kimi-VL's long-context activation consists of not only long text, but also long multimodal data, including long interleaved data, long videos, and long documents. Similar as cooldown data, we also synthesize a small portion of QA pairs to augment the learning efficiency of long-context activation. After the long-context activations, the model can pass needle-in-a-haystack (NIAH) evaluations with either long pure-text or long video haystack, proving its versatile long-context ability. We provide the NIAH recall accuracy on various range of context length up to 128K in Table 2.

2.4 Post-Training Stages

Joint Supervised Fine-tuning (SFT) In this phase, we fine-tune the base model of Kimi-VL with instruction-based fine-tuning to enhance its ability to follow instructions and engage in dialogue, culminating in the creation of the interactive Kimi-VL model. This is achieved by employing the ChatML format (Openai, 2024), which allows for a targeted instruction optimization while maintaining architectural consistency with Kimi-VL. We optimize the language model, MLP projector, and vision encoder using a mixture of pure-text and vision-language SFT data, which will be described in Sec 3.2. Supervision is applied only to answers and special tokens, with system and user prompts being masked. The model is exposed to a curated set of multimodal instruction-response pairs, where explicit dialogue role tagging, structured injection of visual embeddings, and preservation of cross-modal positional relationships are ensured through the format-aware packing. Additionally, to guarantee the model's comprehensive proficiency in dialogue, we incorporate a mix of multimodal data and pure text dialogue data used in Moonlight, ensuring its versatility across various dialogue scenarios.

We first train the model at the sequence length of 32k tokens for 1 epoch, followed by another epoch at the sequence length of 128k tokens. In the first stage (32K), the learning rate decays from 2×10^{-5} to 2×10^{-6}, before it re-warmups to 1×10^{-5} in the second stage (128K) and finally decays to 1×10^{-6}. To improve training efficiency, we pack multiple training examples into each single training sequence.

Long-CoT Supervised Fine-Tuning With the refined RL prompt set, we employ prompt engineering to construct a small yet high-quality long-CoT warmup dataset, containing accurately verified reasoning paths for both text and image inputs. This approach resembles rejection sampling (RS) but focuses on generating long-CoT reasoning paths through prompt engineering. The resulting warmup dataset is designed to encapsulate key cognitive processes that are fundamental to human-like reasoning, such as **planning**, where the model systematically outlines steps before execution; **evaluation**, involving critical assessment of intermediate steps; **reflection**, enabling the model to reconsider and refine its approach; and **exploration**, encouraging consideration of alternative solutions. By performing a lightweight SFT on this warm-up dataset, we effectively prime the model to internalize these multimodal reasoning strategies. As a result, the fine-tuned long-CoT model demonstrates improved capability in generating more detailed and logically coherent responses, which enhances its performance across diverse reasoning tasks.

Reinforcement Learning To further advance the model's reasoning abilities, we then train the model with reinforcement learning (RL), enabling the model to autonomously generate structured CoT rationales. Specifically, similar as Kimi k1.5 (K. Team et al. 2025), we adopt a variant of online policy mirror descent as our RL algorithm, which iteratively refines the policy model π_θ to improve its problem-solving accuracy. During the i-th training iteration, we treat the current model as a reference policy model and optimize the following objective, regularized by relative entropy to stabilize policy updates:

$$\max_{\theta} \mathbb{E}_{(x,y^*)\sim\mathcal{D}} \left[\mathbb{E}_{(y,z)\sim\pi_\theta} \left[r(x,y,y^*)\right] - \tau \mathrm{KL}(\pi_\theta(x)||\pi_{\theta_i}(x))\right] , \tag{1}$$

where r is a reward model that justifies the correctness of the proposed answer y for the given problem x, by assigning a value $r(x, y, y^*) \in \{0, 1\}$ based on the ground truth y^*, and $\tau > 0$ is a parameter controlling the degree of regularization.

Each training iteration begins by sampling a problem batch from the dataset $\mathcal{D}$, and the model parameters are updated to θ_{i+1} using the policy gradient derived from (1), with the optimized policy model subsequently assuming the role of reference policy for the subsequent iteration. To enhance RL training efficiency, we implement a length-based reward to penalize excessively long responses, mitigating the overthinking problem where the model generates redundant reasoning chains. Besides, we employ two sampling strategies including curriculum sampling and prioritized sampling, which leverage difficulty labels and per-instance success rates to focus training effort on the most pedagogically valuable examples, thereby optimizing the learning trajectory and improving training efficiency.

Through large-scale reinforcement learning training, we can derive a model that harnesses the strengths of both basic prompt-based CoT reasoning and sophisticated planning-enhanced CoT approaches. During inference, the model maintains standard autoregressive sequence generation, eliminating the deployment complexities associated with specialized planning algorithms that require parallel computation. Simultaneously, the model develops essential meta-reasoning abilities including error detection, backtracking, and iterative solution refinement by effectively utilizing the complete history of explored reasoning paths as contextual information. With endogenous learning from its complete reasoning trace history, the model can effectively encode planned search procedures into its parametric knowledge.

2.5 Infrastructure

Storage We utilize S3 (Amazon Web Services 2023) compatible object storage from cloud service vendors to store our visual-text data. To minimize the time between data preparation and model training, we store visual data in its original format and have developed an efficient and flexible data loading system. This system provides several key benefits:

- Supports on-the-fly data shuffling, mixing, tokenization, loss masking and packing during training, allowing us to adjust data proportions as needed;
- Enables random augmentation of both visual and text data, while preserving the correctness of 2D coordinate and orientation information during transformations;
- Ensures reproducibility by strictly controlling random states and other states across different data loader workers, guaranteeing that any interrupted training can be resumed seamlessly—the data sequence after resumption remains identical to an uninterrupted run;
- Delivers high-performance data loading: through multiple caching strategies, our system reliably supports training on large scale clusters while maintaining controlled request rates and throughput to the object storage.

Additionally, to ensure consistent dataset quality control, we developed a centralized platform for data registration, visualization, compiling statistics, synchronizing data across cloud storage systems, and managing dataset lifecycles.

Parallelism We adopt a 4D parallelism strategy—Data Parallelism (S. Li et al. 2020), Expert Parallelism (Fedus et al. 2022), Pipeline Parallelism (Y. Huang et al. 2019; Narayanan et al. 2021), and Context Parallelism (Jacobs et al. 2023; H. Liu et al. 2023)—to accelerate the speed of Kimi-VL . After optimizing parallel strategies, the resulting training throughput of our model is around 60% higher than a 7B dense VLM (*e.g.* VLMs based on Qwen2.5-7B).

- **Data Parallelism (DP).** DP replicates the model across multiple devices, each processing different micro-batches. This setup allows larger effective batch sizes by simply increasing the number of devices.
- **Expert Parallelism (EP).** EP distributes expert modules in the MoE layer across multiple devices. When combined with DP, experts on a given device can handle tokens from different DP groups, enhancing computational efficiency.
- **Pipeline Parallelism (PP).** PP splits the model into multiple layer-based stages. To minimize pipeline bubbles, we allocate the Vision Tower (VT) and several decoder layers to the first stage, place the output layer and additional decoder layers in the last stage, and distribute the remaining decoder layers evenly across intermediate stages based on their time overhead.
- **Context Parallelism (CP).** CP addresses long-sequence training by splitting sequences across different CP ranks in conjunction with flash attention (Dao et al. 2022). This substantially reduces peak memory usage and relieves the memory pressure from attention computations.

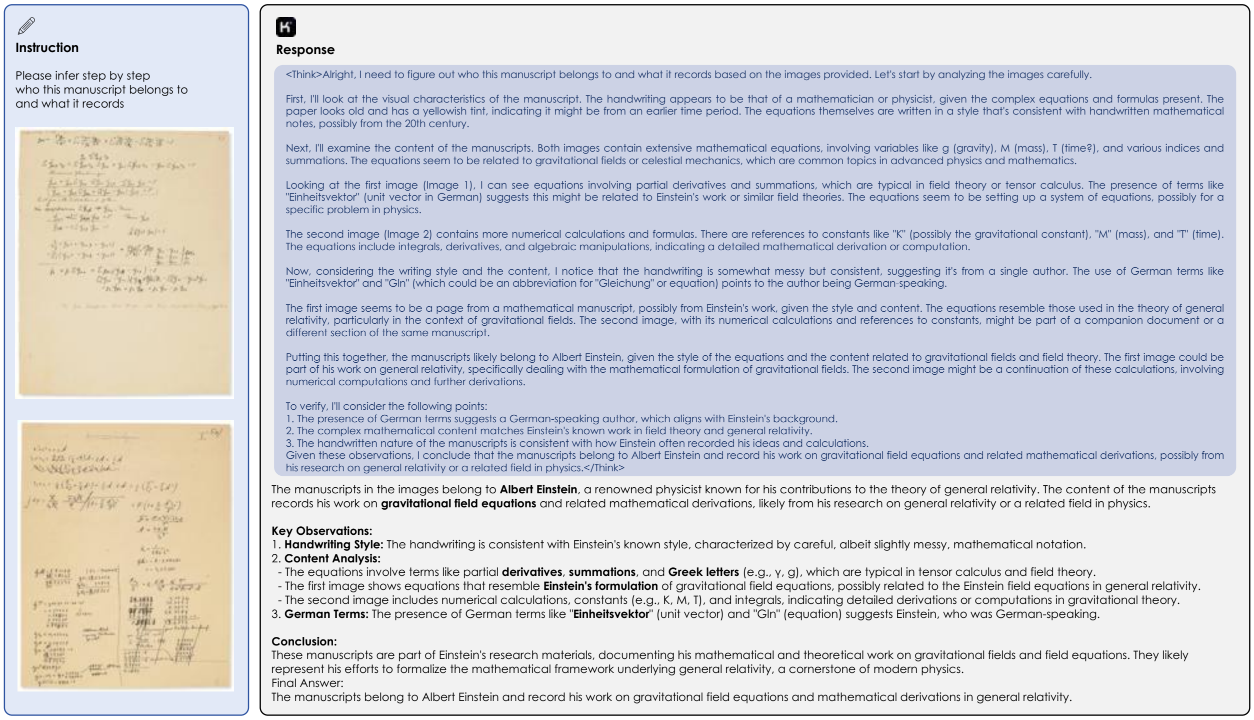


Figure 6: Manuscript reasoning visualization. Kimi-VL-Thinking demonstrates the ability to perform historical and scientific inference by analyzing handwritten manuscripts step by step. In this example, our model identifies the author as Albert Einstein based on handwriting style, content analysis, and language cues. It reasons that the manuscripts relate to gravitational field equations, consistent with Einstein's contributions to general relativity.

Beyond these four parallel strategies, we incorporate ZeRO1 (Rajbhandari et al. 2020) and Selective Checkpointing Activation (T. Chen et al. 2016; Korthikanti et al. 2022) to further optimize memory usage. ZeRO1 reduces optimizer state overhead by using a distributed optimizer while avoiding extra communication costs. Selective Checkpointing Activation trades time for space by recomputing only those layers that have low time overhead but high memory consumption, striking a balance between computation efficiency and memory demands. For extremely long sequences, we expand recomputation to a broader set of layers to prevent out-of-memory errors.

3 Data Construction

3.1 Pre-Training Data

Our multimodal pre-training corpus is designed to provide high-quality data that enables models to process and understand information from multiple modalities, including text, images, and videos. To this end, we have also curated high-quality data from six categories – caption, interleaving, OCR, knowledge, video, and agent – to form the corpus.

When constructing our training corpus, we developed several multimodal data processing pipelines to ensure data quality, encompassing filtering, synthesis, and deduplication. Establishing an effective multimodal data strategy is crucial during the joint training of vision and language, as it both preserves the capabilities of the language model and facilitates alignment of knowledge across diverse modalities.

We provide a detailed description of these sources in this section, which is organized into the following categories:

Caption Data Our caption data provides the model with fundamental modality alignment and a broad range of world knowledge. By incorporating caption data, the multimodal LLM gains wider world knowledge with high learning efficiency. We have integrated various open-source Chinese and English caption datasets like (Schuhmann et al. 2022; Gadre et al. 2024) and also collected substantial in-house caption data from multiple sources. However, throughout the training process, we strictly limit the proportion of synthetic caption data to mitigate the risk of hallucination stemming from insufficient real-world knowledge.

For general caption data, we follow a rigorous quality control pipeline that avoids duplication and maintain high image-text correlation. We also vary image resolution during pre-training to ensure that the vision tower remains effective when processing images of both high- and low-resolution.

Image-text Interleaving Data During the pre-training phase, the model benefits from interleaving data for many aspects. For example, multi-image comprehension ability can be boosted by interleaving data; interleaving data always provides detailed knowledge for the given image; a longer multimodal context learning ability can also be gained by interleaving data. What's more, we also find that interleaving data can contribute positively to maintaining the model's language abilities. Thus, image-text interleaving data is an important part in our training corpus. Our multimodal corpus considered open-sourced interleave datasets like (Zhu et al. 2024; Laurençon et al. 2024) and also constructed large-scale in-house data using resources like textbooks, webpages, and tutorials. Further, we also find that synthesizing the interleaving data benefits the performance of multimodal LLM for keeping the text knowledge. To ensure each image's knowledge is sufficiently studied, for all the interleaving data, despite standard filtering, deduping, and other quality control pipeline, we also integrate a data reordering procedure to keep all the image and text in the correct order.

OCR Data Optical Character Recognition (OCR) is a widely adopted technique that converts text from images into an editable format. In our model, a robust OCR capability is deemed essential for better aligning the model with human values. Accordingly, our OCR data sources are diverse, ranging from open-source to in-house datasets, encompassing both clean and augmented images, and spanning over single-page and multi-page inputs.

In addition to the publicly available data, we have developed a substantial volume of in-house OCR datasets, covering multilingual text, dense text layouts, web-based content, and handwritten samples. Furthermore, following the principles outlined in OCR 2.0 (Wei et al. 2024), our model is also equipped to handle a variety of optical image types, including figures, tables, geometry diagrams, mermaid plots, and natural scene text. We apply extensive data augmentation techniques—such as rotation, distortion, color adjustments, and noise addition—to enhance the model's robustness. As a result, our model achieves a high level of proficiency in OCR tasks.

In addition to single-page OCR data, we collect and convert a large volume of in-house multi-page OCR data to activate the model's understanding of long documents in the real world. With the help of these data, our model is capable of performing accurate OCR on a single image but can also comprehend an entire academic paper or a scanned book.

Knowledge Data The concept of multimodal knowledge data is analogous to the previously mentioned text pre-training data, except here we focus on assembling a comprehensive repository of human knowledge from diverse sources to further enhance the model's capabilities. For example, carefully curated geometry data in our dataset is vital for developing visual reasoning skills, ensuring the model can interpret the abstract diagrams created by humans.

Our knowledge corpus adheres to a standardized taxonomy to balance content across various categories, ensuring diversity in data sources. Similar to text-only corpora, which gather knowledge from textbooks, research papers, and other academic materials, multimodal knowledge data employs both a layout parser and an OCR model to process content from these sources. While we also include filtered data from internet-based and other external resources.

Because a significant portion of our knowledge corpus is sourced from internet-based materials, infographics can cause the model to focus solely on OCR-based information. In such cases, relying exclusively on a basic OCR pipeline may limit training effectiveness. To address this, we have developed an additional pipeline that better captures the purely textual information embedded within images.

Agent Data For agent tasks, the model's grounding and planning capabilities have been significantly enhanced. In addition to utilizing publicly available data, a platform has been established to efficiently manage and execute virtual machine environments in bulk. Within these virtual environments, heuristic methods were employed to collect screenshots and corresponding action data. This data was then processed into dense grounding formats and continuous trajectory formats. The design of the Action Space was categorized according to Desktop, Mobile, and Web environments. Furthermore, icon data was collected to strengthen the model's understanding of the meanings of icons within software graphical user interfaces (GUIs). To enhance the model's planning ability for solving multi-step desktop tasks, a set of computer-use trajectories was collected from human annotators, each accompanied by synthesized Chain-of-Thought (Aguvis (Yiheng Xu et al. 2024)). These multi-step agent demonstrations equip Kimi-VL with the capability to complete real-world desktop tasks (on both Ubuntu and Windows).

Video Data In addition to image-only and image-text interleaved data, we also incorporate large-scale video data during pre-training, cooldown, and long-context activation stages to enable two directions of essential abilities of our model: first, to understand a long-context sequence dominated by images (*e.g.* hour-long videos) in addition to long text; second, to perceive fine-grained spatio-temporal correspondence in short video clips.

Our video data are sourced from diverse resources, including open-source datasets as well as in-house web-scale video data, and span videos of varying durations. Similarly, to ensure sufficient generalization ability, our video data cover a wide range of scenes and diverse tasks. We cover tasks such as video description and video grounding, among others. For long videos, we carefully design a pipeline to produce dense captions. Similar to processing the caption data, we strictly limit the proportion of the synthetic dense video description data to reduce the risk of hallucinations.

Text Data Our text pretrain corpus directly utilizes the data in Moonlight J. Liu et al. 2025a, which is designed to provide comprehensive and high-quality data for training large language models (LLMs). It encompasses five domains: English, Chinese, Code, Mathematics & Reasoning, and Knowledge. We employ sophisticated filtering and quality control mechanisms for each domain to ensure the highest quality training data. For all pretrain data, we conducted rigorous individual validation for each data source to assess its specific contribution to the overall training recipe. This systematic evaluation ensures the quality and effectiveness of our diverse data composition. To optimize the overall composition of our training corpus, the sampling strategy for different document types is empirically determined through extensive experimentation. We conduct isolated evaluations to identify document subsets that contribute most significantly to the model's knowledge acquisition capabilities. These high-value subsets are upsampled in the final training corpus. However, to maintain data diversity and ensure model generalization, we carefully preserve a balanced representation of other document types at appropriate ratios. This data-driven approach helps us optimize the trade-off between focused knowledge acquisition and broad generalization capabilities.

3.2 Instruction Data

At this stage, the data is primarily aimed at enhancing the model's conversational abilities and instruction-following capabilities. To cover as many scenarios as possible, we enrich the data across different domains. For non-reasoning tasks, including chart interpretation, agent grounding, OCR, image-grounded conversations, question-answering, writing, and text processing, we initially construct a seed dataset through human annotation. This seed dataset is used to train a seed model. Subsequently, we collect a diverse set of prompts and employ the seed model to generate multiple responses to each prompt. Annotators then rank these responses and refine the top-ranked response to produce the final version. For reasoning tasks like visual coding, visual reasoning, and math/science problems, where rule-based and model-based verifications are more accurate and efficient than human judgment, we utilize rejection sampling to expand the SFT dataset. The complete vanilla SFT dataset comprises approximately a 1:1 ratio of text tokens to image tokens.

3.3 Reasoning Data

Our reasoning data is meticulously constructed for activation and enhancement of the model's multimodal reasoning capabilities during both the long-CoT supervised fine-tuning and reinforcement learning stages. Through developing a generation pipeline that resembles rejection sampling (RS) and prompt engineering, we collect and synthesize an amount of high-quality long-CoT data. Specifically, we first assemble a collection of QA data with ground truth annotations that require multi-step reasoning, such as mathematical problem-solving and domain-specific VQA. Subsequently, we sample multiple detailed reasoning trajectories for each question by leveraging a powerful long-CoT model - Kimi k1.5 (K. Team et al. 2025) with curated reasoning prompts. In rejection sampling, we feed the true labels and model predictions into an off-the-shelf reward model for judgment. Wrong chain-of-thought responses are filtered out according to the model evaluation as well as some rule-based rewards, thus improving the reasoning data quality.

4 Evaluation

We begin by presenting our comprehensive model and conducting a comparative analysis with leading state-of-the-art (SoTA) solutions. Following this introduction, we proceed to assess various sub-capabilities of the model through detailed performance evaluations. This part examines how effectively the model handles different tasks and scenarios, providing insights into its strengths and limitations across diverse functional domains.

4.1 Comparison to the State-of-the-Art Models

Table 3 presents a comprehensive evaluation of Kimi-VL against state-of-the-art vision-language models across multiple benchmarks. Although having a more parameter-efficient architecture (2.8B+0.4B activated parameters) compared to larger models such as GPT-4o, Llama-3.2-11B-Inst. and Gemma3-12B-IT, Kimi-VL demonstrates competitive or superior performance in several key areas. Our model employs a Mixture-of-Experts (MoE) architecture similar to DeepSeek-VL2, but outperforms it on most benchmarks with significantly fewer parameters (activated: 2.8B *vs* 4.5B; total: 16B *vs* 28B); it also outperforms Qwen2.5-VL-7B (*actually 8.3B*) on 19 out of 24 benchmarks, though the latter

[*] GPT-4o and GPT-4o-mini results use Omniparser without UIA, according to Bonatti et al. 2024.

	Benchmark (Metric)	GPT-4o	GPT-4o-mini	Qwen2.5-VL-7B	Llama3.2-11B-Inst.	Gemma3-12B-IT	DeepSeek-VL2	Kimi-VL-A3B
	Architecture	-	-	Dense	Dense	Dense	MoE	MoE
	# Act. Params (LLM+VT)	-	-	7.6B+0.7B	8B+2.6B	12B+0.4B	4.1B+0.4B	2.8B+0.4B
	# Total Params	-	-	8B	11B	12B	28B	16B
College-level	MMMU_{val} (Pass@1)	69.1	**60.0**	58.6	48	59.6	51.1	57.0
	VideoMMMU (Pass@1)	61.2	-	47.4	41.8	**57.2**	44.4	52.6
	MMVU_{val} (Pass@1)	67.4	**61.6**	50.1	44.4	57.0	52.1	52.2
General	MMBench-EN-v1.1 (Acc)	83.1	77.1	82.6	65.8	74.6	79.6	**83.1**
	MMStar (Acc)	64.7	54.8	**63.9**	49.8	56.1	55.5	61.3
	MMVet (Pass@1)	69.1	66.9	**67.1**	57.6	64.9	60.0	66.7
	RealWorldQA (Acc)	75.4	67.1	**68.5**	63.3	59.1	68.4	68.1
	AI2D (Acc)	84.6	77.8	83.9	77.3	78.1	81.4	**84.9**
Multi-image	BLINK (Acc)	68.0	53.6	56.4	39.8	50.3	-	**57.3**
Math	MathVista (Pass@1)	63.8	52.5	68.2	47.7	56.1	62.8	**68.7**
	MathVision (Pass@1)	30.4	-	25.1	13.6	**32.1**	17.3	21.4
OCR	InfoVQA (Acc)	80.7	57.9	82.6	34.6	43.8	78.1	**83.2**
	OCRBench (Acc)	815	785	864	753	702	811	**867**
OS Agent	ScreenSpot-V2 (Acc)	18.1	-	86.8	-	-	-	**92.8**
	ScreenSpot-Pro (Acc)	0.8	-	29.0	-	-	-	**34.5**
	OSWorld (Pass@1)	5.03	-	2.5	-	-	-	**8.22**
	WindowsAgentArena (Pass@1)*	9.4	2.7	3.4	-	-	-	**10.4**
Long Document	MMLongBench-Doc (Acc)	42.8	29.0	29.6	13.8	21.3	-	**35.1**
Long Video	Video-MME (w/o sub. / w/ sub.)	71.9/77.2	64.8/68.9	65.1/71.6	46.0/49.5	58.2/62.1	-	**67.8/72.6**
	MLVU_{MCQ} (Acc)	64.6	48.1	70.2	44.4	52.3	-	**74.2**
	$\text{LongVideoBench}_{\text{val}}$	66.7	58.2	56.0	45.5	51.5	-	**64.5**
Video Perception	$\text{EgoSchema}_{\text{full}}$	72.2	-	65.0	54.3	56.9	38.5	**78.5**
	VSI-Bench	34.0	-	34.2	20.6	32.4	21.7	**37.4**
	TOMATO	37.7	28.8	27.6	21.5	28.6	27.2	**31.7**

Table 3: Performance of Kimi-VL against proprietary and open-source efficient VLMs; performance of GPT-4o is also listed in gray for reference. Top and second-best models are in **boldface** and underline respectively. Some results of competing models are unavailable due to limitation of model ability on specific tasks or model context length.

has 2.59× more activated parameters. The following sections analyze performance across specific domains, which reveals Kimi-VL 's strengths in OCR, math, agent, long-form content understanding, multi-image and video perception.

4.1.1 College-level Academic Problems

Our Kimi-VL model demonstrates competitive performance on college-level academic benchmarks. On MMMU validation set, it achieves a score of 57.0%, which outperforms DeepSeek-VL2 (51.1%) and is comparable to Qwen2.5-VL-7B (58.6%) and even Gemma-3-12B-IT (59.6%), despite having significantly fewer activated parameters. On video college-level problems, it significantly outperforms Qwen2.5-VL-7B and DeepSeek-VL2, only behind >10B Gemma-3-12B-IT, demonstrating reasonable university-level understanding capabilities compared to larger models. These results indicate that Kimi-VL effectively balances parameter efficiency with academic reasoning abilities.

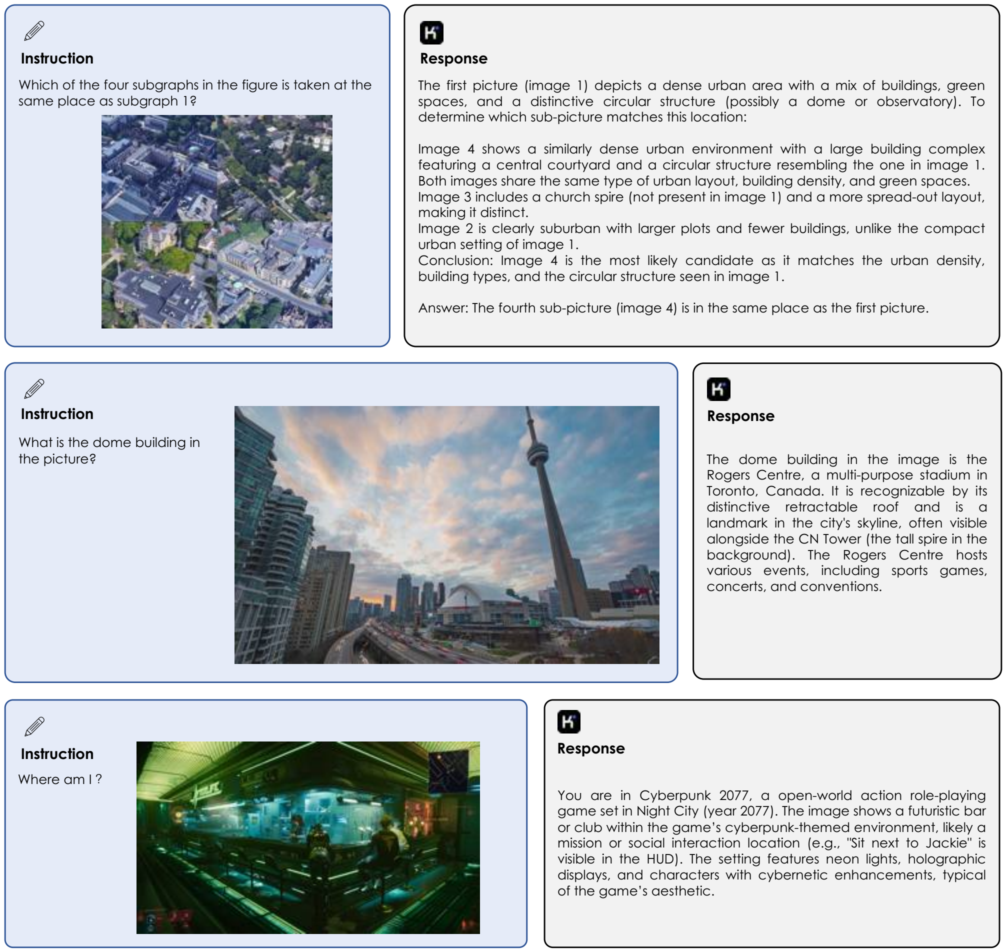


Figure 7: Kimi-VL exhibits strong visual reasoning capabilities by grounding visual content in spatial, contextual, and cultural knowledge. It accurately identifies matching urban locations based on structural and layout features, interprets scenes from video games like Cyberpunk 2077 using stylistic cues, and recognizes real-world landmarks such as the Rogers Centre in Toronto.

4.1.2 General Visual Ability

Kimi-VL exhibits strong general visual understanding capabilities across multiple benchmarks. On MMBench-EN-v1.1, it achieves 83.1% accuracy, outperforming all efficient VLMs in comparison, and performing on par with GPT-4o. For AI2D, our model achieves 84.9% and surpasses all compared models including GPT-4o (84.6%). On MMVet, Kimi-VL scores 66.7% and ties closely with Qwen2.5-VL-7B (67.1%) and GPT-4o-mini (66.9%). For RealWorldQA, it achieves 68.1%, outperforming Gemma3-12B (59.1%) and approaching Qwen2.5-VL-7B (68.5%). These results demonstrate that our model maintains robust general visual understanding despite its compact architecture.

In multi-image reasoning tasks, Kimi-VL shows promising capabilities with a score of 57.3% on the BLINK benchmark. This performance surpasses Qwen2.5-VL-7B (56.4%), GPT-4o-mini (53.6%), Gemma3-12B-IT (50.3%), and Llama3.2-11B-Inst. (39.8%). The ability to reason across multiple images requires understanding spatial and temporal relationships between visual elements, which our model handles effectively with fewer parameters than most competitors.

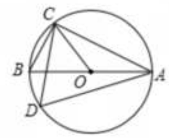
4.1.3 Mathematical Reasoning

With its relatively small scale, Kimi-VL also demonstrates strong mathematical reasoning capabilities, particularly on the MathVista benchmark where it achieves 68.7%, outperforming all compared models including GPT-4o (63.8%) and Qwen2.5-VL-7B (68.2%). It indicates our model's exceptional ability to understand and solve mathematical problems

[†] Video source: `https://vimeo.com/channels/top/54348266`

Instruction

回答问题



如图,AB为⊙O的直径,点D、C在⊙O上,∠D=62°,则∠ACO的度数为() A. 26° B. 28° C. 30° D. 32°

Response

根据题目描述，AB是圆O的直径，点D和C在圆O上，且∠D = 62°。我们需要求∠ACO的度数。

1. 分析已知条件：
- AB是直径，所以∠ACB = 90°（圆周角定理）。
- ∠D = 62°。

2. 利用圆周角定理：
- ∠AOC是圆心角，等于2倍的圆周角∠D。
- ∠AOC = 2 × 62° = 124°。

3. 计算∠ACO:
- ∠ACO是三角形AOC的内角。
- ∠ACO + ∠CAO + ∠AOC = 180°。
- 因为OA = OC（半径相等），所以∠CAO = ∠ACO。
- 设∠ACO = x，则：

$$
\begin{aligned}
x + x + 124° &= 180° \\
2x + 124° &= 180° \\
2x &= 56° \\
x &= 28°
\end{aligned}
$$

因此，∠ACO的度数为28°。答案是B。

Figure 8: Kimi-VL demonstrates its capability to perform symbolic reasoning and geometric inference by solving a circle geometry problem step by step. The model analyzes given conditions, applies geometric theorems such as the inscribed angle theorem and properties of triangle angles, and accurately derives the target angle.

presented in visual contexts. On the more challenging MathVision benchmark, due to limited activated parameters, Kimi-VL outperforms DeepSeek-VL2 and Llama-3.2-11B-Inst., but lags behind Qwen2.5-VL-7B and Gemma-12B-IT. Nevertheless, through RL and test-time scaling, Kimi-VL-Thinking has significantly improved and already on par with 30B-level VLMs (see Table 4). These results highlight our model's effectiveness in combining visual perception with mathematical problem-solving, an essential capability for real-world applications.

4.1.4 Document Understanding and OCR

Kimi-VL excels in document understanding and OCR tasks across all benchmarks in this category. On InfoVQA, it achieves 83.2% accuracy, outperforming GPT-4o (80.7%) and DeepSeek-VL2 (78.1%). For OCRBench, our model scores 86.7%, surpassing all other models including GPT-4o-mini (78.5%) and DeepSeek-VL2 (81.1%). These results demonstrate that our model has exceptional text recognition and document understanding capabilities, making it especially suitable for applications involving document processing and information extraction.

4.1.5 Agent Grounding and Multi-turn Agent Interaction

In agent-based tasks, Kimi-VL demonstrates remarkable performance. On single-step grounding, our model shows strong accuracy, with 92.0% on ScreenSpot-V2 and 34.5% on extremely difficult ScreenSpot-Pro (on 4K screens), proving its strong agent grounding abilities. More importantly, it also shows strong multi-step turn agent interaction abilities: For OSWorld, Kimi-VL reaches 8.22%, outperforming GPT-4o (5.03%) and other capable open-source models; On WindowsAgentArena, our model achieves 10.4%, also surpassing GPT-4o (9.4%) and others. These results highlight Kimi-VL's exceptional ability to understand and interact with operating system interfaces, suggesting strong potential for applications in automated UI navigation and task execution.

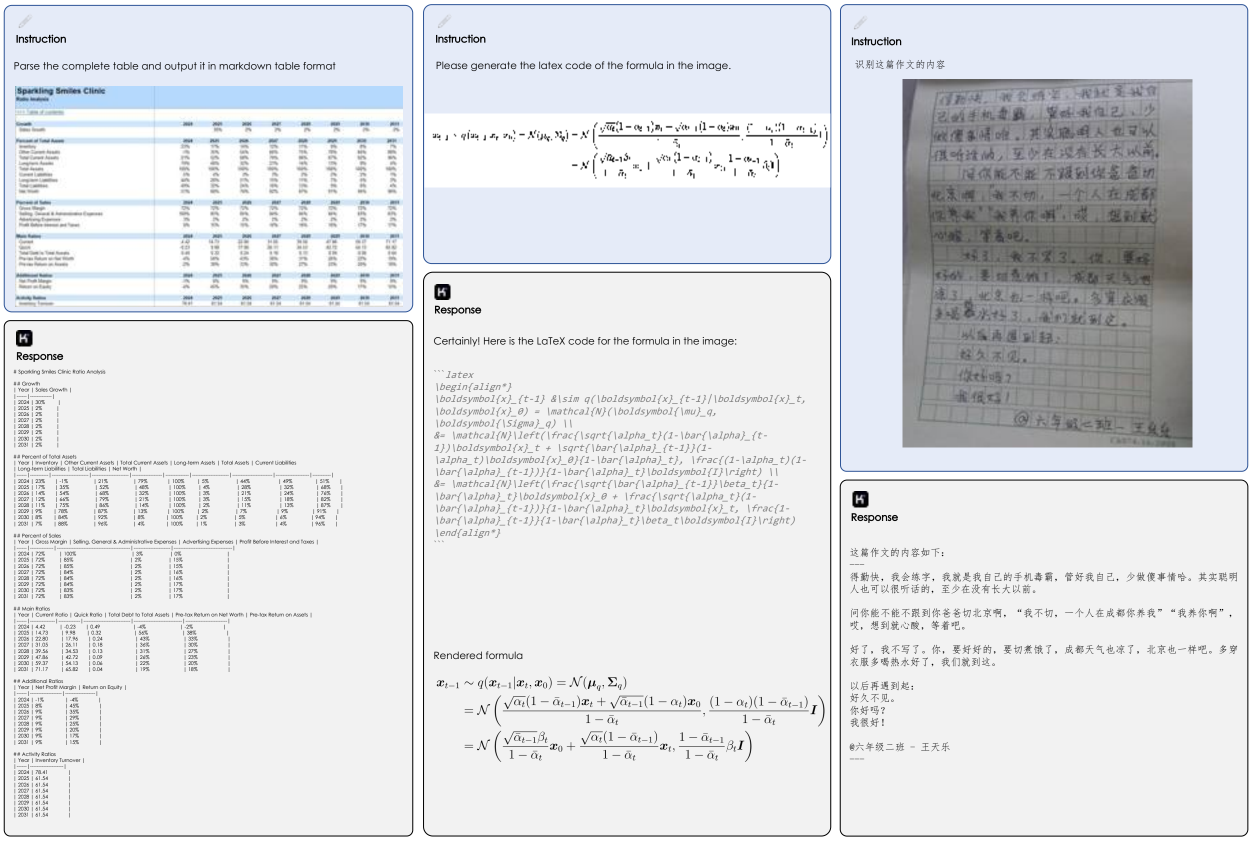


Figure 9: Diverse OCR visualization. Kimi-VL demonstrates strong OCR capabilities across varied content types, including structured financial tables, complex mathematical formulas, and handwritten Chinese text. The model accurately parses tabular data into markdown, converts formulas to LaTeX, and transcribes handwritten paragraphs with contextual understanding, showcasing its versatility in multimodal text extraction and interpretation.

4.1.6 Long Document and Long Video Understanding

Kimi-VL demonstrates competitive performance in long-form content understanding. On MMLongBench-Doc, a challenging benchmark with question-answering on up to 100+ pages, it achieves 35.1%, outperforming GPT-4o-mini (29.0%) and Qwen2.5-VL-7B (29.6%), only behind GPT-4o (42.8%). For long video understanding, on Video-MME, our model outperforms all efficient VLMs and especially leads on the fairer *w/o subtitle* setting, where models have to find answers from video frames instead of hacking from input subtitles; on *w/ subtitle*setting, it also reaches extraordinary 72.6% accuracy. On the MCQ subset of MLVU, Kimi-VL achieves an impressive 74.2% score, achieving state-of-the-art and surpassing both GPT-4o (64.6%) and Qwen2.5-VL-7B (70.2%). For LongVideoBench, it scores 64.5%, outperforming all compared models except GPT-4o (66.7%). These results demonstrate Kimi-VL 's strong capability to understand long-form PDFs and videos.

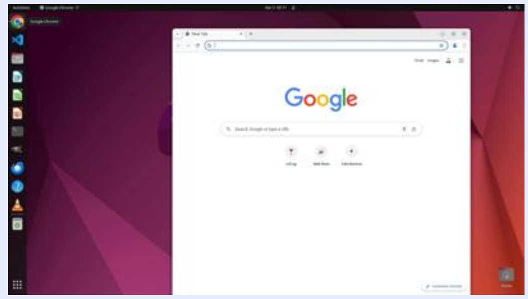
4.1.7 Egocentric and Fine-grained Video Perception

Kimi-VL also shows strong performance in more nuanced video perception tasks. On EgoSchema full set (*hidden test set*), it achieves 78.5%, significantly outperforming GPT-4o (72.2%), Qwen2.5-VL-7B (65.0%). For VSI-Bench, a very challenging benchmark that requires to understand spatial relationships and correspondences of multiple objects in a video, our model scores 37.4%, surpassing GPT-4o (34.0%) and Qwen2.5-VL-7B (34.2%). In TOMATO that examines fine-grained temporal perception of VLMs, Kimi-VL reaches 31.7%, outperforming Qwen2.5-VL-7B (27.6%) and GPT-4o-Mini (28.8%). These results demonstrate our model's strong capability to understand dynamic visual content, track objects over time, and interpret complex actions in video sequences, making it well-suited for applications requiring temporal visual understanding.

‡Video source: `https://www.youtube.com/watch?v=kYWUEV_e2ss`

Instruction:

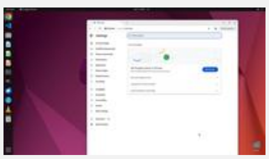
Can you enable the 'Do Not Track' feature in Chrome to enhance my online privacy?



Step 1

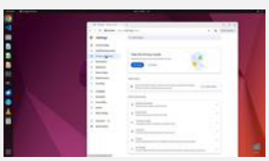
Thought: ... To enable the 'Do Not Track' feature in Chrome, I need to access Chrome's settings. Looking at the interface, I can see the vertical ellipsis menu button in the top-right corner of the browser window. This menu button is the standard way to access Chrome's settings and other features. ...
Action: Click on the vertical ellipsis menu button in the top-right corner of the Chrome window to open the browser menu.
Toolcall: click(x=0.884, y=0.144)

Step 2



Thought: ... The next logical step would be to access Chrome's settings, where privacy options like "Do Not Track" are typically located. Looking at the menu that's currently open, I can see the "Settings" option at the bottom of the menu. This would be the most direct path to access privacy settings. ...
Action: Click on the "Settings" option in the Chrome menu to access the browser's settings page.
Toolcall: click(x=0.744, y=0.851)

Step 3

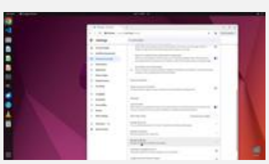


Thought: ... Now, I need to find where the "Do Not Track" setting is located. Based on my knowledge of Chrome's settings, this feature is typically found in the Privacy and security section. I can see that "Privacy and security" is one of the options in the left sidebar menu. ...
Action: Click on the "Privacy and security" option in the left sidebar menu.
Toolcall: click(x=0.386, y=0.311)

Step 4

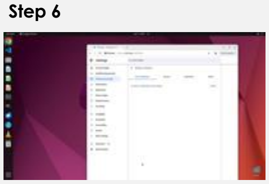
Thought: ... Now we need to find the "Do Not Track" setting. Looking at the current screen, I can see that "Do Not Track" is not immediately visible in the list of options. However, I notice that there's a "Security" option with a lock icon that might contain privacy-related settings including "Do Not Track". ...
Action: Click on the "Security" option in the Privacy and security section to access more detailed security settings.
Toolcall: click(x=0.522, y=0.884)

Step 5



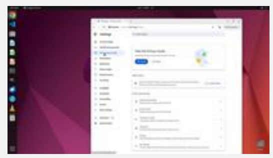
Thought: ... To find the "Do Not Track" setting, I need to scroll down in the Security section. The current view doesn't show all available options, and the "Do Not Track" setting is likely further down the page. Scrolling down will reveal more settings options that aren't currently visible. ...
Action: Scroll down in the Security settings page to reveal more options.
Toolcall: scroll(-5)

Step 6



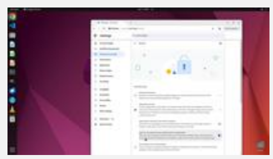
Thought: ... The "Manage HTTPS/SSL certificates and settings" option is visible in the list. This is likely where the "Do Not Track" setting would be located, as certificate management is related to privacy and security features. ...
Action: Click on the "Manage HTTPS/SSL certificates and settings" option to access certificate management settings.
Toolcall: click(x=0.527, y=0.883)

Step 7



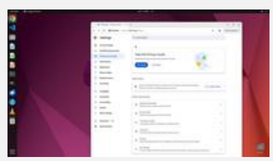
Thought: ... I notice that we're currently on the "Manage certificates" page, which is not where the "Do Not Track" setting would be located. ... To enable the "Do Not Track" feature, we need to navigate to the appropriate settings section. ...
Action: Click on the "Privacy and security" option in the left sidebar menu to expand its submenu.
Toolcall: click(x=0.365, y=0.317)

Step 8



Thought: ... I can see that "Security" is one of the options in the Privacy and security section. This makes sense as "Do Not Track" is a security feature that helps protect user privacy by preventing websites from tracking browsing activity. ...
Action: Click on the "Security" option in the Privacy and security section to access more detailed security settings.
Toolcall: click(x=0.522, y=0.884)

Step 9

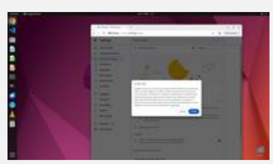


Thought: ... Since the current page doesn't show the "Do Not Track" setting, the logical next action would be to continue navigating through the settings. The back button in the browser would allow the user to return to the previous page and continue their search for the "Do Not Track" setting. ...
Action: Click the back button in the browser to return to the previous page.
Toolcall: click(x=0.486, y=0.242)

Step 10

Thought: ... Looking at the available options, "Third-party cookies" is visible in the list. This option controls whether third-party cookies are blocked in Incognito mode. While this isn't exactly "Do Not Track," it is related to privacy controls. ...
Action: Click on the "Third-party cookies" option in the Privacy and security section to access its settings.
Toolcall: click(x=0.542, y=0.772)

Step 11



Thought: ... I notice there's an "Advanced" section in the current view that contains a "Send a 'Do Not Track' request with your browsing traffic" option. This is exactly what we need to enable the "Do Not Track" feature. The toggle switch for this option is currently off. ...
Action: Click the toggle switch next to "Send a 'Do Not Track' request with your browsing traffic" to enable the feature.
Toolcall: click(x=0.808, y=0.871)

Step 12

Thought: ... The dialog explains what "Do Not Track" does and has two buttons: "Cancel" and "Confirm". To complete the task of enabling this feature, we need to confirm the action by clicking the "Confirm" button. ...
Action: Click the "Confirm" button to enable the "Do Not Track" feature.
Toolcall: click(x=0.708, y=0.669)

Figure 10: Kimi-VL is capable of following multi-step reasoning processes to complete complex GUI tasks. In this example, it successfully enables the “Do Not Track” feature in the Chrome browser to enhance online privacy. The agent interprets each screen, identifies relevant UI elements, and performs the appropriate actions sequentially with clear thoughts, actions, and API calls.

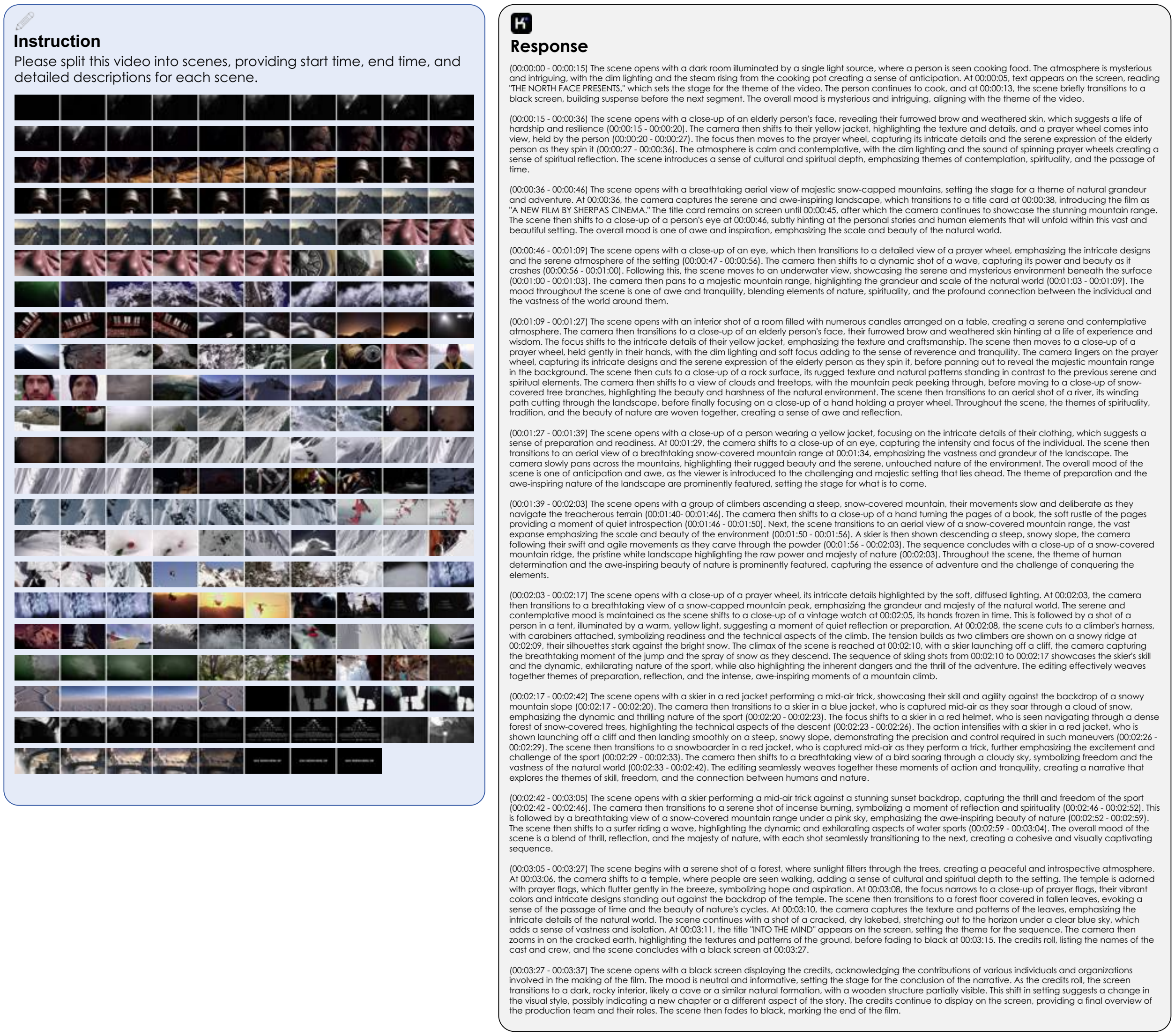


Figure 11: Video scene splitting. Kimi-VL processes a long-form video by segmenting it into coherent scenes and providing detailed start/end timestamps along with fine-grained natural language descriptions for each scene.†

4.2 `Kimi-VL-A3B-Thinking`: A Reasoning Extension of Kimi-VL

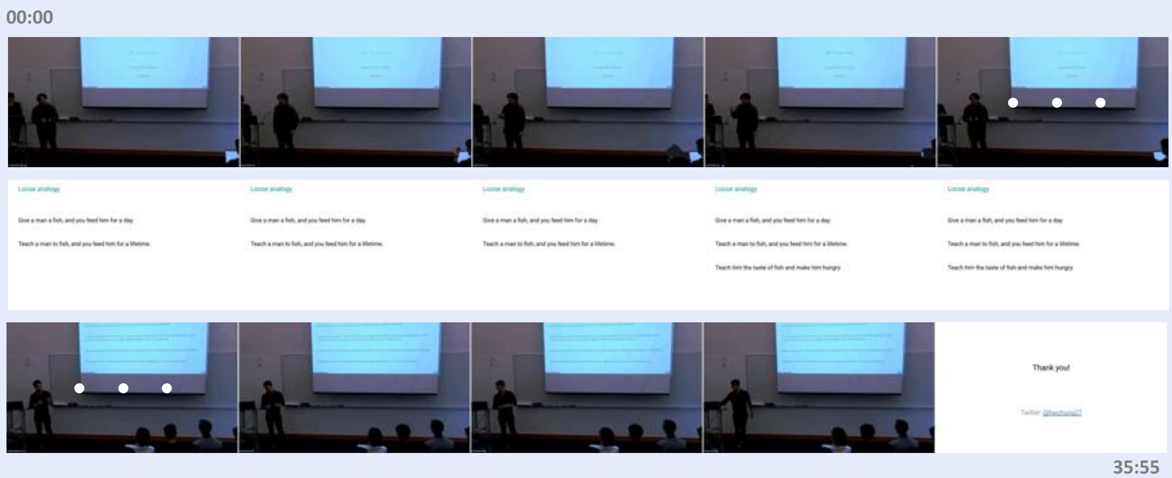
Furthermore, we conduct a reasoning extension to empower Kimi-VL to reason with CoT and present a long-thinking version of the model, **Kimi-VL-Thinking**, through long-CoT activation and reinforcement learning. We validate its superior performance on several image benchmarks, as shown in Table 4.

Kimi-VL-Thinking significantly improves over the base Kimi-VL model, with gains of 2.6% on MathVista, 4.7% on MMMU, and 15.4% on MathVision, demonstrating its capability to leverage test-time computation for deeper reasoning and better handling of complex multimodal queries. In Table 4, Kimi-VL-Thinking further outperforms or rivals state-of-the-art thinking and non-thinking models: achieving 71.3% on MathVista, outperforming GPT-4o (63.8%) and GPT-4o-mini (56.7%); scoring 61.7% on MMMU, surpassing GPT-4o-mini (60.0%) and Qwen2.5-VL-7B (58.6%); and reaching 36.8% on MathVision, exceeding GPT-4o (30.4%) and Gemma-3-27B-IT (35.5%), even QVQ-72B (35.9%). While marginally behind some larger-scale models on select benchmarks, Kimi-VL-Thinking accomplishes these results with only 3B activated parameters—orders of magnitude fewer than its counterparts—underscoring its strong efficiency and effectiveness in multimodal reasoning.

Our Kimi-VL-Thinking model also exhibits strong test-time scaling properties, as shown in Figure 13. Specifically, increasing the max thinking token length at inference time consistently improves test-time accuracy across all three

Instruction

There is an old Chinese saying: Give a man a fish, and you feed him for a day; teach a man to fish, and you feed him for a lifetime. In this representation, the author puts forward a further requirement. Find it then explain it in detail.



Response

The requirement is to "Teach him the taste of fish and make him hungry." This implies that in addition to teaching someone how to perform a task or acquire a skill, it is also important to inspire and motivate them to continue learning and improving on their own. By making them hungry for more knowledge or experience, they will be more likely to seek out new challenges and opportunities for growth, which can lead to greater success and fulfillment in the long run.

Figure 12: Catching and understanding key details from an hour-long video course. Kimi-VL demonstrates its ability to comprehend and interpret instructional video content by analyzing frame sequences and extracting conceptual progression over time. In this case, the model identifies a deepening of the traditional saying "Teach a man to fish, and you feed him for a lifetime" into a more nuanced idea: "Teach him the taste of fish and make him hungry."‡

Benchmark (Metric)	**Non-Thinking Model**						**Thinking Model**				
	GPT-4o	**GPT-4o-mini**	**Qwen2.5-VL-72B**	**Qwen2.5-VL-7B**	**Gemma-3-27B**	**Gemma-3-12B**	**o1-1217**	**QVQ-72B-Preview**	**Kimi-k1.5**	**Kimi-VL-A3B-Thinking**	**Kimi-VL-A3B-Thinking-2506**
MathVision (full) (Pass@1)	30.4	-	38.1	25.1	35.5	32.1	-	35.9	38.6	36.8	**56.9**
MathVista (mini) (Pass@1)	63.8	56.7	74.8	68.2	62.3	56.4	71.0	71.4	74.9	71.3	**80.1**
MMMU (val) (Pass@1)	69.1	60.0	74.8	58.6	64.8	59.6	**77.3**	70.3	70.0	61.7	64.0
MMMU-Pro (avg) (Pass@1)	**51.7**	37.6	51.1	38.1	-	32.1	-	-	-	43.0	46.3
VideoMMMU (Pass@1)	61.1	-	60.2	47.0	61.8	57.2	-	-	-	55.5	**65.2**

Table 4: Performance of Kimi-VL-Thinking and Kimi-VL-Thinking-2506 on multimodal reasoning benchmarks. The metrics evaluated include MathVista (mini), MMMU (val), MMMU-Pro (average), MathVision (full) and VideoMMMU, with results expressed in Pass@1. The Kimi-VL-Thinking-2506 performs well in most cases, showcasing the enhanced reasoning and processing capabilities of the *"thinking"* variant across different domains and scales.

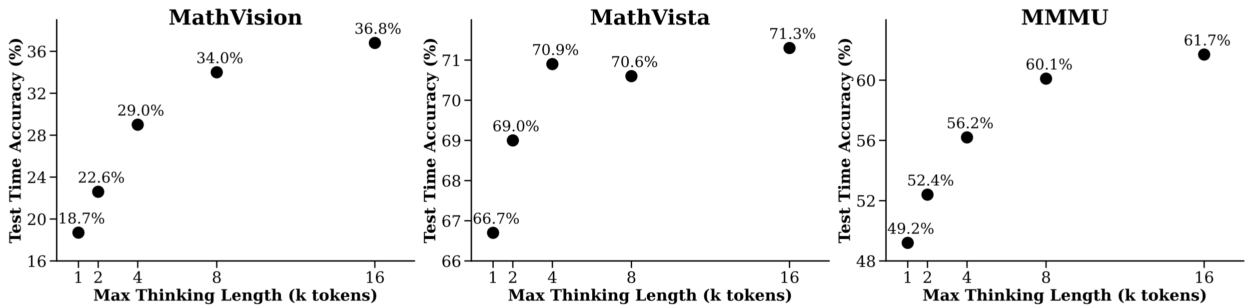


Figure 13: Test-time accuracy when scaling the max thinking token length of our **Kimi-VL-Thinking** model.

benchmarks. For example, on **MathVision**, the accuracy rises steadily from 18.7% at 1k tokens to 36.8% at 16k tokens, and similar upward trend is also observed on **MMMU**, indicating that the model is able to utilize longer reasoning chains for better performance. However, not all benchmarks benefit equally from longer thinking lengths. On **MathVista**, performance saturates early, with accuracy reaching 70.9% at 4k tokens and no further significant gains observed as the token length increases to 16k. It suggests that for this task, the necessary reasoning depth is already captured within a relatively short context, and additional computation does not yield further improvements.

4.3 `Kimi-VL-A3B-Thinking-2506`: From Reasoning Extension to Integrated Thinking Model

Table 5: Performance of Kimi-VL-A3B-Thinking-2506 on multimodal benchmarks that do not require extensive reasoning.

Benchmark (Metric)	GPT-4o	Qwen2.5-VL-7B	Gemma3-12B-IT	Kimi-VL-A3B-Instruct	Kimi-VL-A3B-Thinking	Kimi-VL-A3B-Thinking-2506
General Multimodal						
MMBench-EN-v1.1 (Acc)	83.1	83.2	74.6	82.9	76.0	**84.4**
RealWorldQA (Acc)	75.4	68.5	59.1	68.1	64.0	**70.0**
OCRBench (Acc)	815	864	702	864	864	**869**
MMStar (Acc)	64.0	63.0	56.1	61.7	64.2	**70.4**
MMVet (Acc)	69.1	67.1	64.9	66.7	69.5	**78.1**
Video						
MMVU_{val} (Pass@1)	67.4	50.1	57.0	52.7	53.0	**57.5**
Video-MME (w/ sub.) (Acc)	77.2	71.6	62.1	**72.7**	66.0	71.9
OS-Agent Grounding						
ScreenSpot-Pro (Acc)	0.8	29.0	—	35.4	—	**52.8**
ScreenSpot-V2 (Acc)	18.1	84.2	—	**92.8**	—	91.4
OSWorld-G (Acc)	-	31.5	—	41.6	—	**52.5**
Long Document						
MMLongBench-Doc (Acc)	42.8	29.6	21.3	35.1	32.5	**42.1**

While Kimi-VL-A3B-Thinking shows excellent thinking abilities on hard reasoning tasks, we further provide the updated **Kimi-VL-A3B-Thinking-2506**[§], a new reasoning variant that is not only smarter, but integrates key abilities of Kimi-VL-A3B-Instruct (perception, video, long-document, and OS-agent abilities) into this thinking model.

Kimi-VL-Thinking-2506 significantly improves reasoning efficiency while reducing token consumption. As shown in Table 4, Kimi-VL-Thinking-2506 achieves 56.9% on MathVision (+20.1% improvement on original Kimi-VL-Thinking), 80.1% on MathVista (+8.4%), 46.3% on MMMU-Pro (+3.2%), and 64.0% on MMMU (+2.1%), demonstrating non-trivial gains across multiple reasoning benchmarks. Meanwhile, while solving these hard reasoning problems, the 2506 version reduces the average output token length by around 20% (*e.g.*, 2.9K → 2.4K on MMMU-val and 5.8K → 4.4K on MathVision), facilitating it to be more efficient and user-friendly for practical deployments.

[§]Tech Blog: `https://huggingface.co/blog/moonshotai/kimi-vl-a3b-thinking-2506`

Beyond extensive reasoning tasks, Kimi-VL-Thinking demonstrates stronger visual perception capabilities (Table 5). Compared to the previous non-thinking variant (Kimi-VL-A3B-Instruct), Kimi-VL-A3B-Thinking-2506 achieves competitive or superior results on general multimodal understanding benchmarks: 84.4% on MMBench-EN-v1.1, 70.4% on MMStar, 70.0% on RealWorldQA, and 78.4% on MMVet, underscoring its broader competence in vision-language tasks. In terms of token efficiency, the 2506 version only requires in average 180 tokens per answer when solving MMBench, 1/3 compared to the previous thinking model while improving 8.4% accuracy.

Kimi-VL-A3B-Thinking-2506 also extends its reasoning ability to video and long-context domains. It establishes new state-of-the-art results among open-source models on VideoMMMU (65.2%, 4% better than GPT-4o), a challenging video reasoning benchmark; it also maintains robust general video understanding performance with 71.9% on Video-MME, matching the long video understanding ability of Kimi-VL-A3B-Instruct. It also scores 42.1% (first open-source model matching GPT-4o) on MMLongBench-Doc (Table 5), a 10% improvement over the previous thinking model and 7% over the previous instruct model, demonstrating its robust ability on broader long-form visual inputs.

As mentioned in the method part, the continual training on MoonViT (3.2 million max input pixels) of Kimi-VL-A3B-Thinking-2506 leads to substantial improvements on **high-resolution** perception and OS grounding benchmarks, achieving 83.2% on V* Benchmark (without external tools), 52.8% on ScreenSpot-Pro, and 52.5% on OSWorld-G (full set with refusal samples), showing huge improvements over both previous variants. We hope that this high-resolution multimodal reasoning model brings about interesting new capabilities in the real world.

5 Conclusion, Limitation, and Future Work

We introduce Kimi-VL, a VLM designed with a balanced approach to cover both multimodal and text-only pre-training/post-training, underpinned by an MoE-based architecture for scalable efficiency. Its 128K extended context window enables precise retrieval in lengthy texts and videos, while the native-resolution encoder MoonViT helps maintain high accuracy with low computational overhead in ultra-high-resolution visual tasks. Additionally, Kimi-VL-Thinking facilitates effective long-chain reasoning in complex image and video inference. Overall, Kimi-VL demonstrates robust adaptability and efficiency across multimodal, long-context, and high-resolution tasks, indicating substantial potential for future research and industrial applications.

However, Kimi-VL still faces several challenges:

1. Although the current model size performs effectively for many standard tasks, it remains too limited to address highly specialized or domain-specific problems, or problems that are strongly dependent on language abilities, restricting Kimi-VL's ability to handle extremely complex scenarios.
2. While the reasoning capability is already strong for typical use cases, it has yet to reach its theoretical upper bound, particularly for intricate tasks requiring multi-step inference or deeper contextual understanding.
3. Despite providing a 128K extended context window, due to limited parameters in its attention layers (which is only comparable to a 3B model), its long-context abilities is still insufficient for certain advanced applications that involve extremely long sequences or high-volume contextual information.

In the future, we will tackle these challenges by scaling up the model size, expanding pre-training data, and enhancing post-training algorithms. Our next steps include optimizing Kimi-VL and releasing larger versions, as well as refining post-training and test-time scaling mechanisms for a better thinking model. These efforts will pave the way for more advanced applications in both research and industry.

References

Amazon Web Services. *Amazon Simple Storage Service (Amazon S3)*. Web. Available at: `https://aws.amazon.com/s3/`. 2023. URL: `https://aws.amazon.com/s3/` (visited on 12/15/2023).

Bai, Shuai et al. *Qwen2.5-VL Technical Report*. 2025. arXiv: `2502.13923 [cs.CV]`. URL: `https://arxiv.org/abs/2502.13923`.

Bonatti, Rogerio et al. *Windows Agent Arena: Evaluating Multi-Modal OS Agents at Scale*. 2024. arXiv: `2409.08264 [cs.AI]`. URL: `https://arxiv.org/abs/2409.08264`.

Chen, Lin et al. "Are We on the Right Way for Evaluating Large Vision-Language Models?" In: *arXiv preprint arXiv:2403.20330* (2024).

Chen, Tianqi et al. *Training Deep Nets with Sublinear Memory Cost*. 2016. arXiv: `1604.06174 [cs.LG]`. URL: `https://arxiv.org/abs/1604.06174`.