DeepLatent: Think with Images via Parallel Latent Visual Reasoning

Anonymous ACL submission

Abstract

The emerging paradigm of "thinking with images" embeds visual states into intermediate reasoning steps, defining a new frontier for Vision-Language Models. Existing approaches diverge along two lines. Tool-assisted methods apply explicit visual operations but suffer from high latency and restricted manipulation types. Latent reasoning methods autoregressively produce implicit visual states, but underperform tool-assisted methods, and their latent tokens fail to capture effective visual information. In this work, we propose **DeepLatent**, a parallel framework for latent visual reasoning. First, we introduce LatentFormer. It uses learnable 2D tokens to generate context-conditioned latent states in parallel, anchoring every visual update directly in the original image features. Second, we design a continuous-space reinforcement learning algorithm. It optimizes latent modulation parameters directly in the embedding space, significantly improving latent representation quality. The framework is trained via knowledge distillation followed by this continuous-space RL algorithm. Furthermore, we contribute DeepLatent-180K, a large-scale dataset tailored for latent visual reasoning. Extensive evaluations across multiple benchmarks demonstrate that DeepLatent achieves state-of-the-art performance. Code, model and data are available in the appendix.

1 Introduction

Recent advances in Vision-Language Models (VLMs) have enabled increasingly sophisticated visual reasoning, particularly through Chain-of-Thought (CoT) approaches (Xu et al., 2025a; Huang et al., 2025). However, in mainstream VLMs (Li et al., 2024; Zhu et al., 2025), the reasoning trajectory is typically expressed purely in language. The visual representation remains static after a single initial encoding step. Consequently, the model cannot dynamically revisit or adapt its visual focus as reasoning unfolds. In contrast, human cognition naturally interleaves vision and reasoning, iteratively updating visual states based on an evolving understanding of the task. Bridging this gap requires embedding active visual interaction directly into the reasoning process, enabling models to genuinely "think with images."

Existing efforts mainly explore this direction in two ways. Tool-based methods manipulate images through external operations or executable code (OpenAI, 2025; Zheng et al., 2025; Wu et al., 2025; Zhang et al., 2025c). These methods are effective and provide interpretable visual evidence, but they are limited to predefined manipulation types and often incur high inference latency. Latent visual reasoning methods take a more integrated path. They generate latent visual tokens in an autoregressive (AR) paradigm, internalizing intermediate visual reasoning as a model capability. However, latent approaches still underperform tool-based methods on challenging benchmarks (e.g., a gap of up to 5% on HRBench), and they usually require complex latent alignment.

Beyond these practical limitations, latent reasoning faces a deeper challenge. Empirical analysis (Li et al., 2026) reveals that latent tokens in previous work exhibit high inter-token homogeneity and contain limited visual information. We relate this phenomenon to two systematic biases in the AR paradigm. (1) *Structural mismatch.* Images are continuous 2D fields whose semantics arise from spatial neighborhoods, not from a unidirectional causal chain. Forcing AR generation over flattened 1D sequences conflicts with this structure, making it difficult for latents to capture spatial semantics. (2) *Information source deviation.* Latent visual reasoning aims to derive new visual representations from the input image. However, in AR generation, later latents are conditioned primarily on earlier latent hidden states rather than image features. This shifts the information flow from image-to-latent to latent-to-latent. Moreover,

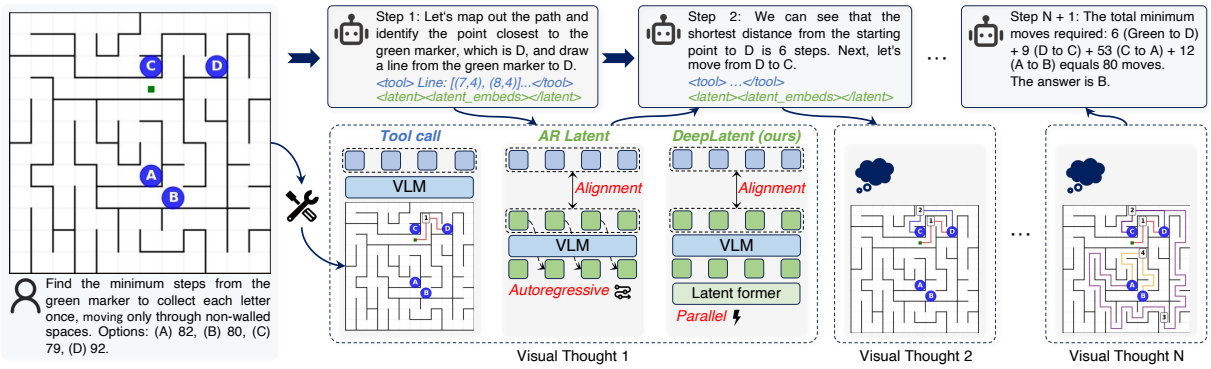


Figure 1: Interleaved visual-textual reasoning and architectural comparison. Unlike prior approaches that rely on discrete tool calls or serial autoregressive latent generation, DeepLatent employs a parallel LatentFormer to enable efficient and synchronized visual-state updates during complex reasoning.

these latent hidden states replace the visual features typically generated by the visual encoder. Their distributions can differ substantially from those of the encoder outputs, which may weaken the pretrained vision-language alignment.

To address these biases, we propose **DeepLatent**, a parallel, context-conditioned latent reasoning framework built on two key principles. First, it generates all latents in parallel, removing the structural mismatch of AR generation. Second, it grounds every latent directly in image features, avoiding information source deviation. At the core of DeepLatent is LatentFormer. It initializes learnable 2D latent queries and refines them through multi-layer cross-attention over image features. We further derive three modulation parameters to adapt the latents to the reasoning context. They control the latent scale, latent distribution, and image feature distribution. The generated latents are fed into the LLM in a single forward pass, enabling efficient and synchronized visual state updates. On top of this framework, we design a continuous-space reinforcement learning (RL) algorithm to optimize the modulation parameters directly in the embedding space. This further improves latent quality beyond what supervised distillation alone can achieve.

Our contributions are summarized as follows:

(1) We propose DeepLatent, a parallel and context-conditioned latent visual reasoning framework. Its LatentFormer jointly generates latent states while grounding each of them directly in visual features, improving both reasoning performance and inference efficiency.

(2) We develop a continuous-space sampling and RL algorithm for latent variable optimization. Combined with knowledge distillation, it substantially improves latent quality and yields strong performance across diverse benchmarks.

(3) We introduce DeepLatent-180K, a large-scale dataset for latent visual reasoning, which we release to facilitate further research in this domain.

2 Related Work

2.1 Thinking with Images via Tools

A line of work augments VLMs with external visual tools for dynamic image manipulation during reasoning. **Programmatic visual interfaces.** Early methods prompt LLMs to generate executable code that invokes visual APIs (Gupta and Kembhavi, 2023; Surís et al., 2023; Yang et al., 2023). Thyme (Zhang et al., 2025c) and related work further extend such interfaces to support richer image manipulation and distilled program execution (Hu et al., 2024b,a; Fu et al., 2025; Zhao et al., 2025). Reward-driven approaches instead learn when and how to invoke tools through RL (Su et al., 2025; Wu et al., 2025). **Visual grounding.** DyFo (Li et al., 2025d) and related methods focus on task-relevant regions through bounding-box prediction or spatial search (Wu and Xie, 2024). Some models explicit grounding steps as intermediate supervision (Shao et al., 2024a; Qi et al., 2025). DeepEyes (Zheng et al., 2025), Pixel-Reasoner (Wang et al., 2025b), and related work (Fan et al., 2025; Lai et al., 2025) further unify grounding and reasoning via RL. Additional efforts revisit visual evidence through explicit re-insertion or attention redistribution (Chu et al., 2025; Yang et al., 2026).

2.2 Latent Visual Reasoning

Motivated by latent reasoning in LLMs (Hao et al., 2024; Cheng and Van Durme, 2024), recent work embeds visual information into continuous latents during VLM reasoning. Monet, LVR, and Mirage (Wang et al., 2025c; Li et al., 2025b; Yang et al., 2025) supervise latent tokens via visual dis-

tillation in multi-stage pipelines, then apply RL to refine reasoning. CoVT (Qin et al., 2025) introduces expert-specific visual tokens, aligned during training by multiple auxiliary expert heads. Others align latents to visual encoder outputs at each reasoning step (Wu et al., 2026b; Jeon et al., 2026), or distill visual features through teacher-guided regression (Dong et al., 2025; Zhang et al., 2025a). In contrast, some methods (Pham and Ngo, 2025; Ray et al., 2025) learn modality-agnostic tokens in a unified continuous space without explicit visual distillation. Further work explores iterative visual refocusing (Ma et al., 2025), memory-augmented grounding (Yu et al., 2025b), single-token compression (Lv et al., 2026), and predictive latent trajectory learning (Adhikari and Lapata, 2026). Despite their varied designs, these methods typically produce latent states autoregressively.

Tool-assisted methods incur high latency and limited flexibility. Latent methods avoid explicit tool overhead, but remain less competitive and retain limited visual information (Li et al., 2026). DeepLatent bridges this gap via parallel latent generation anchored to the original image features.

3 DeepLatent

As illustrated in Fig. 2, DeepLatent employs a dedicated LatentFormer for parallel latent generation, bypassing autoregressive latent prediction inside the LLM. When the VLM emits a `<latent>` token, a set of learnable 2D latent embeddings interacts with the question image features via cross-attention, forming a spatial latent grid grounded in the image. These latents are inserted jointly into the reasoning sequence as internal visual states, so the LLM can process the full grid in a single forward pass rather than one token at a time.

Built on Qwen2.5-VL (Bai et al., 2025b), DeepLatent is trained with a three-stage pipeline, following prior latent visual reasoning methods. In SFT Stage 1, the model is trained on image-text interleaved data with assistant images, and the corresponding LLM hidden states are cached as teacher visual states. In SFT Stage 2, the assistant images are removed, and the LatentFormer is trained to reconstruct the cached teacher states with a cosine similarity loss, together with next-token prediction and latent scale losses. In the RL stage, we apply GRPO (Shao et al., 2024b) to further refine latent invocation and optimize latent modulation parameters in continuous space.

3.1 SFT Stage 1: Interleaved Reasoning

Image-text interleaved reasoning trajectories provide rich supervision, as they include assistant images that serve as intermediate visual context for subsequent textual reasoning. In SFT Stage 1, we train the model with standard next-token prediction over complete trajectories:

$$\mathcal{L}_{\mathrm{S1}} = -\sum_{t \in \mathcal{T}} \log p_\theta(y_t \mid y_{<t}, q, I_0, I_{1:N}), \quad (1)$$

where y_t is the ground-truth response token at position $t \in \mathcal{T}$, q and I_0 are the input question text and image, and $I_{1:N}$ are the assistant images. After training, we perform a forward pass over the same trajectories and cache the LLM hidden states corresponding to each assistant image. These cached "teacher visual states" will supervise the latent distillation in SFT Stage 2.

3.2 SFT Stage 2: Latent Distillation

In SFT Stage 2, explicit assistant images in the training trajectories are replaced by latent blocks. DeepLatent learns to recover the visual states previously provided by these images, conditioned on both the reasoning context and the question image.

Latent Construction. When the model emits a `<latent>` token, we interrupt the autoregressive decoding and inject three learnable embeddings: a scale-control embedding $\mathbf{e}_s$, a latent-modulation embedding $\mathbf{e}_l$, and an image-modulation embedding $\mathbf{e}_i$. The LLM then produces corresponding hidden states $\mathbf{h}_s$, $\mathbf{h}_l$, and $\mathbf{h}_i$ conditioned on the current reasoning context. Their functional roles are strictly decoupled: $\mathbf{h}_s$ dictates the latent spatial scale; $\mathbf{h}_l$ modulates the queries to drive visual search (e.g., localizing relevant regions); and $\mathbf{h}_i$ modulates the question image features to perform implicit visual editing (e.g., synthesizing auxiliary structures). The LatentFormer constructs the latent embeddings $\mathbf{Z}$ from these states, after which a `</latent>` token closes the block and text generation resumes. The complete latent sequence takes the form: `<latent>` $\mathbf{e}_s$ $\mathbf{e}_l$ $\mathbf{e}_i$ $\mathbf{Z}$ `</latent>`.

Spatial latent queries. We maintain a learnable latent grid $\mathbf{L} \in \mathbb{R}^{M_H \times M_W \times d}$ as the query template for LatentFormer, where (M_H, M_W) denotes the maximum grid size. A latent scale head takes $\mathbf{h}_s$ as input and predicts the grid height and width as two independent discrete distributions:

$$\begin{aligned} \hat{\mathbf{p}}_H &= \mathrm{MLP}_h(\mathbf{h}_s), \quad \hat{H}_z = \arg\max(\hat{\mathbf{p}}_H), \\ \hat{\mathbf{p}}_W &= \mathrm{MLP}_w(\mathbf{h}_s), \quad \hat{W}_z = \arg\max(\hat{\mathbf{p}}_W). \end{aligned} \quad (2)$$

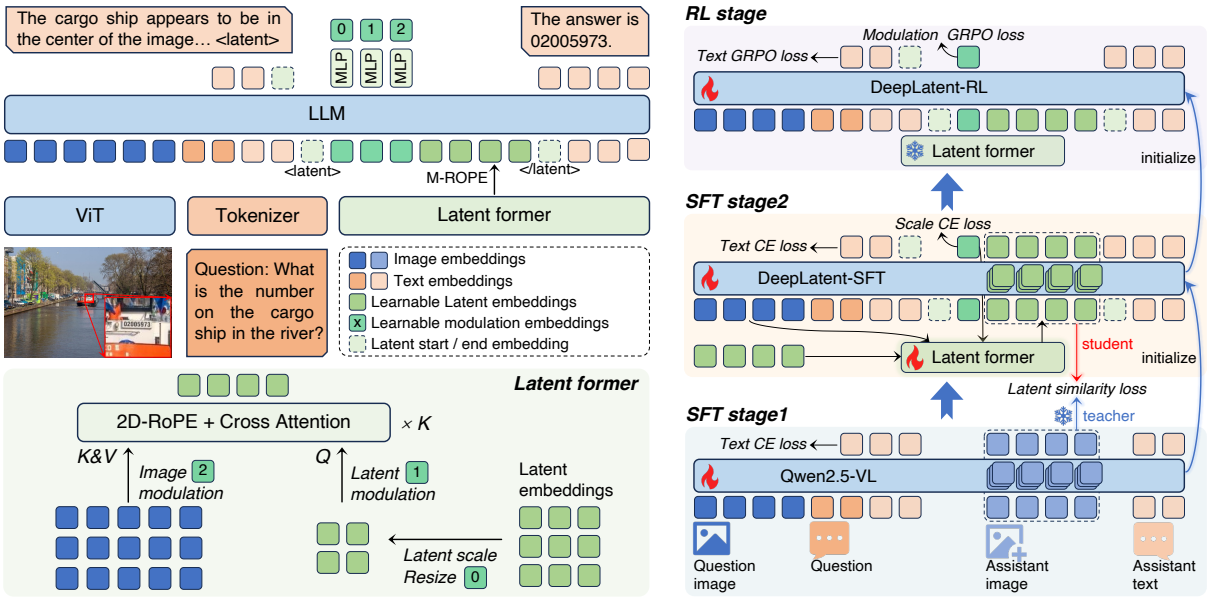


Figure 2: DeepLatent architecture and training pipeline. Left: At each `<latent>` position, the LLM modulates the LatentFormer to generate latents and insert them into the sequence. Right: SFT Stage 1 trains Qwen2.5-VL on interleaved image-text data and caches assistant image hidden states. SFT Stage 2 trains the LatentFormer to reconstruct cached visual states. RL stage jointly optimizes text generation and latent modulation via GRPO.

To decouple scale learning from latent construction, we use the ground-truth (GT) grid size (H_z^*, W_z^*), derived from the assistant image feature dimensions, for latent generation during training, and use the predicted size $(\hat{H}_z, \hat{W}_z)$ at inference. The query grid $\mathbf{L}$ is then spatially resized to $H_z \times W_z \times d$.

Stochastic affine modulation. For each modulation state $\mathbf{h}_x$ ($x \in \{l, i\}$), an independent MLP predicts channel-wise modulation parameters $[\boldsymbol{\gamma}_x, \boldsymbol{\beta}_x, \log \boldsymbol{\sigma}_x] \in \mathbb{R}^{1\times1\times3d}$, which are broadcast across spatial dimensions. We then modulate the resized latent queries $\mathbf{L}$ and the question image features $\mathbf{I}_0 \in \mathbb{R}^{H_I \times W_I \times d}$ as:

$$\begin{aligned} \tilde{\mathbf{L}} &= \boldsymbol{\gamma}_l \odot \mathbf{L} + \boldsymbol{\beta}_l + \boldsymbol{\sigma}_l \odot \boldsymbol{\epsilon}_l, \\ \tilde{\mathbf{I}} &= \boldsymbol{\gamma}_i \odot \mathbf{I}_0 + \boldsymbol{\beta}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}_i, \end{aligned} \tag{3}$$

where $\boldsymbol{\epsilon}_l$ and $\boldsymbol{\epsilon}_i$ are i.i.d. $\mathcal{N}(0, 1)$ noise. This operation couples a deterministic affine transformation with a reparameterized stochastic perturbation. The injected noise acts as an implicit regularizer during training, promoting a smoother latent space and better generalization. At inference, the stochastic term is omitted for deterministic execution.

LatentFormer. LatentFormer is a lightweight latent transformer with K stacked blocks. Starting from the initial latent grid $\mathbf{Z}^{(0)} = \tilde{\mathbf{L}}$, each block applies self-attention to refine the latent features, followed by cross-attention with the modulated image features $\tilde{\mathbf{I}}$. The k-th block computes:

$$\begin{aligned} \mathbf{Z}' &= \mathbf{Z}^{(k-1)} + \mathrm{SelfAttn}\big(R_{2\mathrm{D}}(\mathbf{Z}^{(k-1)})\big), \\ \mathbf{Z}^{(k)} &= \mathbf{Z}' + \mathrm{CrossAttn}\big(R_{2\mathrm{D}}(\mathbf{Z}'), R_{2\mathrm{D}}(\tilde{\mathbf{I}}), \tilde{\mathbf{I}}\big), \end{aligned} \tag{4}$$

where $R_{2\mathrm{D}}(\cdot)$ denotes 2D RoPE (Wang et al., 2024). The final latent representation $\mathbf{Z} = \mathbf{Z}^{(K)} \in \mathbb{R}^{H_z \times W_z \times d}$ is flattened into $H_z W_z$ latent tokens and inserted into the designated latent positions of the LLM input sequence.

Loss Function. SFT Stage 2 objective consists of three losses. First, we employ a next-token prediction loss to supervise discrete generation:

$$\mathcal{L}_{\mathrm{CE}} = -\textstyle\sum_{t \in \mathcal{T}} \log p_\theta(y_t \mid y_{<t}, q, I_0, \mathbf{Z}_{1:N}), \tag{5}$$

where $\mathcal{T}$ denotes the positions of discrete text and `<latent>` tokens, and the continuous latent embeddings $\mathbf{Z}_{1:N}$ serve only as conditioning context.

Second, we align the student's latent-block hidden states with the cached teacher visual states from Stage 1 via a cosine similarity loss:

$$\mathcal{L}_{\mathrm{Align}} = \frac{1}{|\mathcal{D}||\mathcal{P}|} \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}} \big(1 - \cos(\mathbf{h}_{d,p}^{\mathrm{T}}, \mathbf{h}_{d,p}^{\mathrm{S}})\big) \tag{6}$$

where $\mathbf{h}_{d,p}^{\mathrm{T}}$ and $\mathbf{h}_{d,p}^{\mathrm{S}}$ denote the teacher and student hidden states at layer d and position p, respectively. Here, $\mathcal{D}$ denotes the selected LLM layers used for alignment, and $\mathcal{P}$ denotes the alignment positions.

Third, we supervise the latent spatial scale:

$$\mathcal{L}_{\mathrm{Scale}} = \mathrm{CE}(\hat{\mathbf{p}}_H, H_z^*) + \mathrm{CE}(\hat{\mathbf{p}}_W, W_z^*), \tag{7}$$

where $\hat{\mathbf{p}}_H \in \mathbb{R}^{M_H}, \hat{\mathbf{p}}_W \in \mathbb{R}^{M_W}$ are the predicted scale logits, and (H_z^*, W_z^*) are the GT grid sizes.

The final loss is formulated as:

$$\mathcal{L}_{\mathrm{S2}} = \mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{Align}} \mathcal{L}_{\mathrm{Align}} + \lambda_{\mathrm{Scale}} \mathcal{L}_{\mathrm{Scale}}, \tag{8}$$

in which we set $\lambda_{\mathrm{Align}} = 4.0$ and $\lambda_{\mathrm{Scale}} = 0.5$.

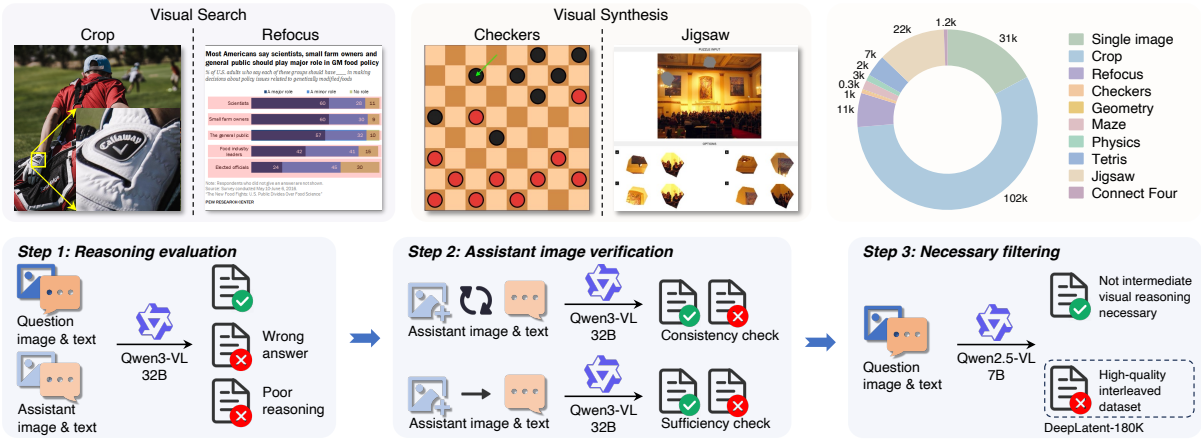


Figure 3: Overview of the DeepLatent-180K dataset composition and the automated filtering pipeline.

3.3 Reinforcement Learning

To further refine the model's reasoning capabilities, we apply reinforcement learning from verifiable rewards via GRPO. Our objective is to jointly optimize the discrete decision of *when* to invoke a latent block and the continuous decision of *how* to modulate the latent embeddings. To preserve the latent construction pipeline learned during SFT, we freeze the visual encoder, LatentFormer, and MLP_s. Only the LLM and the modulation MLPs (MLP_l, MLP_i) are updated in this stage.

Continuous-space GRPO. Standard GRPO is defined for discrete text tokens and cannot directly optimize stochastic modulation. We therefore extend the action space to include both discrete actions (text and `<latent>` tokens) and continuous modulation actions produced by MLP_l and MLP_i. The stochastic modulation in Eq. (3) naturally defines the corresponding action distributions. Under policy π_θ, the modulation MLPs predict Gaussian parameters for latent query and image feature modulation. Given the sampled modulation outputs $\tilde{\mathbf{L}}^{\mathrm{old}}$ and $\tilde{\mathbf{I}}^{\mathrm{old}}$ from rollout, their log-probabilities under the current policy are computed as:

$$\begin{aligned} \log \pi_\theta(\tilde{\mathbf{L}}^{\mathrm{old}}) &= \log \mathcal{N}\big(\tilde{\mathbf{L}}^{\mathrm{old}};\ \boldsymbol{\mu}_\theta^l,\ (\boldsymbol{\sigma}_\theta^l)^2\big), \\ \log \pi_\theta(\tilde{\mathbf{I}}^{\mathrm{old}}) &= \log \mathcal{N}\big(\tilde{\mathbf{I}}^{\mathrm{old}};\ \boldsymbol{\mu}_\theta^i,\ (\boldsymbol{\sigma}_\theta^i)^2\big), \end{aligned} \tag{9}$$

where $\boldsymbol{\mu}_\theta^l = \boldsymbol{\gamma}_{\theta,l} \odot \mathbf{L} + \boldsymbol{\beta}_{\theta,l}$, and $\boldsymbol{\mu}_\theta^i = \boldsymbol{\gamma}_{\theta,i} \odot \mathbf{I}_0 + \boldsymbol{\beta}_{\theta,i}$. We sum the log-densities over all dimensions of the sampled modulation tensor. The sampled modulations are then deterministically mapped to the latent block $\mathbf{Z}$ by the frozen latent module. Hence, $\mathbf{Z}$ itself is not treated as an RL action.

GRPO objective. For each image-question pair (I_0, q), the policy samples G trajectories $\{\mathbf{o}_g\}_{g=1}^G$ and assigns each trajectory an outcome reward. A normalized advantage is computed as:

$$A^{(g)} = \frac{R^{(g)} - \mathrm{mean}(\{R^{(j)}\}_{j=1}^G)}{\mathrm{std}(\{R^{(j)}\}_{j=1}^G)}. \tag{10}$$

We update the policy π_θ by maximizing:

$$\begin{aligned} \mathcal{J}(\theta) = \mathbb{E}_{I_0,q,\mathbf{o}\sim\pi_{\mathrm{old}}} \frac{1}{G}\sum_{g=1}^{G} \frac{1}{|\mathcal{M}^{(g)}|} \sum_{t\in\mathcal{M}^{(g)}} \Big[\\ \min\big(r_{g,t}A^{(g)}, \mathrm{clip}(r_{g,t}, 1-\epsilon, 1+\epsilon)A^{(g)}\big) \\ - \beta_{\mathrm{KL}}\mathbb{D}_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}})\Big]. \end{aligned} \tag{11}$$

Here, $\mathcal{M}^{(g)}$ indexes all policy actions in trajectory g, including text tokens, `<latent>` tokens, and continuous modulation actions, and $|\mathcal{M}^{(g)}|$ is the number of such actions. At step t, the importance sampling ratio is $r_{g,t} = \frac{\pi_\theta(\mathbf{o}_{g,t}|I_0,q,\mathbf{o}_{g,<t})}{\pi_{\theta_{\mathrm{old}}}(\mathbf{o}_{g,t}|I_0,q,\mathbf{o}_{g,<t})}$. Both $r_{g,t}$ and $\mathbb{D}_{\mathrm{KL}}$ take the corresponding form for the action type at step t: categorical probabilities and standard KL divergence for discrete tokens, and Gaussian densities and closed-form KL for continuous modulations. The reference policy π_{ref} is frozen from SFT Stage 2.

3.4 DeepLatent-180K

High-quality interleaved image-text data is essential for DeepLatent-SFT, yet existing datasets exhibit two limitations: many interleaved samples can be answered without intermediate images (providing no valid supervision for intermediate visual reasoning), and current Visual Search data lacks sufficient resolution for fine-grained details. To address this, we introduce DeepLatent-180K, augmenting existing datasets with high-resolution images and rigorous filtering to ensure all samples strictly require intermediate visual reasoning.

As summarized in Fig. 3, the dataset comprises *single-image* and *interleaved reasoning* samples. The latter is categorized into two main types: **Vi-**

sual Search (e.g., cropping or refocusing on relevant regions) and **Visual Synthesis** (e.g., constructing intermediate states for mazes, jigsaw).

We construct the training data from two primary sources. First, we curate samples from existing VLM datasets (Zebra-CoT (Li et al., 2025a), ReFocus, CogCoM, Thyme-SFT). Second, to enhance visual perception capacity, we utilize high-resolution images from SAM-1B (Kirillov et al., 2023), FineRS (Zhang et al., 2025b), V*, InfoVQA (Mathew et al., 2022), and PosterSum (Saxena et al., 2025). For these images, we prompt Qwen3-VL-32B (Bai et al., 2025a) to generate question-answer pairs targeting local details. These images and questions are subsequently fed into Thyme-7B to produce complete interleaved reasoning samples.

To ensure data quality and reasoning necessity, we design a three-stage automated filtering pipeline. **(1) Reasoning evaluation.** Qwen3-VL-32B evaluates the full trajectory, discarding incorrect answers or flawed logic. **(2) Assistant image verification.** The same model checks each intermediate assistant image for consistency (aligning visually with the intermediate text) and sufficiency (providing enough evidence to derive the final answer), discarding any failures. **(3) Necessity filtering.** We evaluate whether Qwen2.5-VL-7B can correctly answer the question without access to the assistant images. If correct, we discard these simple samples. Ultimately, this rigorous pipeline yields the high-quality DeepLatent-180K dataset.

4 Experiments

4.1 Experimental Setup

Training Configuration. We build DeepLatent on top of Qwen2.5-VL-7B. In SFT Stage 1, the model is trained on the full DeepLatent-180K dataset. In SFT Stage 2, we train two variants: **DeepLatent**, which uses the full DeepLatent-180K mixture, and **DeepLatent***, which uses only the visual search subset to emphasize high-resolution localization. In the RL stage, we further optimize the model on a 10K subset of Thyme-RL. All training stages are conducted for one epoch. The maximum latent grid size is set to $M_H = M_W = 16$, yielding up to 256 tokens per latent block. Additional implementation details are provided in the supplementary material.

Evaluated Benchmarks. We evaluate DeepLatent on 13 benchmarks spanning three categories. *(1) High-Resolution Perception* includes V* (Wu and Xie, 2024), HRBench4K, HRBench8K (Wang et al., 2025d), MME-RealWorld, MME-RealWorld-Lite (Zhang et al., 2025d), and VisualProbe-Hard (Lai et al., 2025). *(2) General Evaluation* includes OCRBench (Liu et al., 2024), ChartQA (Masry et al., 2022), MMStar (Chen et al., 2024), and RealWorldQA (xAI, 2024). *(3) Visual Reasoning* includes VisuLogic (Xu et al., 2025b), TreeBench (Wang et al., 2025a), and BabyVision (Chen et al., 2026). We adopt the VLMEvalKit framework for fair evaluation.

4.2 Evaluation Results

We report results for both the intermediate SFT model and the final RL models, denoted as DeepLatent-SFT, DeepLatent-RL, and DeepLatent-RL* in Tab. 1. For brevity, we refer to the final RL models as DeepLatent and DeepLatent* below.

High-Resolution Perception. DeepLatent delivers strong gains on high-resolution perception. It achieves 84.8% on V*, 77.2% on HR4K, and 74.0% on HR8K, improving over Qwen2.5-VL-7B by 8.3%, 8.4%, and 8.7%, respectively. It also consistently outperforms prior latent reasoning methods and reaches or exceeds tool-augmented methods, surpassing Thyme by 0.2% on HR4K and 2.0% on HR8K, and DeepEyes by 3.6% on VisualProbe-Hard. On MME-RealWorld and MME-RealWorld-Lite, DeepLatent achieves 64.2% and 53.7%, improving over the base model by 6.9% and 9.4%. The DeepLatent* variant trained with visual search data further improves the results to 85.3% on V*, 77.9% on HR4K, and 74.2% on HR8K.

General Multimodal Understanding. While substantially improving visual perception and reasoning, DeepLatent also preserves strong general multimodal understanding. It achieves 86.4% on OCRBench and 86.4% on ChartQA, comparable to the base model, and further obtains 65.0% on MMStar and 71.5% on RWQA, improving over Qwen2.5-VL-7B by 3.1% and 4.1%, respectively.

Visual Reasoning. DeepLatent also brings consistent gains on visual reasoning tasks. It achieves 25.3% on VisuLogic, 39.0% on TreeBench, and 16.2% on BabyVision, improving over Qwen2.5-VL-7B by 5.3%, 2.0%, and 3.3%, respectively. Compared with DeepLatent*, which is trained only with visual search data, DeepLatent achieves more balanced performance and stronger results across all reasoning benchmarks.

We further evaluate DeepLatent on Jigsaw and Multi-view Reasoning from BLINK (Fu et al., 2024), and Visual Spatial Planning (Wu et al.,

Model	High-Resolution Perception						General Evaluation				Visual Reasoning		
	V*	HR4K	HR8K	MME-RW	MME-RW-Lite	Visual Probe-H	OCR Bench	Chart QA	MMStar	RWQA	Visu Logic	Tree Bench	Baby Vision
Open-source Models													
LLaVA-OneVision-7B	75.4	63.0	59.8	57.4	48.5	13.4	-	80.0	61.9	-	-	37.3	-
Qwen2.5-VL-7B	76.5	68.8	65.3	57.3	44.3	23.9	86.4	86.2	61.9	67.4	20.0	37.0	12.9
InternVL3-8B	81.2	70.0	69.3	61.2	48.6	-	**88.0**	86.0	**68.2**	70.8	24.9	38.5	12.9
Tool-based Models													
Pixel-Reasoner-7B	84.3	74.0	66.9	64.4	-	28.8	-	-	-	-	-	39.0	-
DeepEyes-7B	**85.6**	75.1	72.6	64.1	53.2	35.1	-	-	-	-	-	37.5	-
Thyme-7B	82.2	77.0	72.0	**64.8**	55.2	-	86.3	86.1	65.9	70.2	23.4	-	-
DyFo-7B	84.3	71.3	69.8	-	-	-	-	-	-	-	-	-	-
Latent Visual reasoning Models													
LVR-7B	81.7	70.8	63.0	-	50.6	-	-	-	-	67.7	-	39.0	-
Monet-7B	83.3	71.0	68.0	-	**55.5**	-	-	-	-	-	-	-	-
CoVT-7B	78.5	72.5	69.9	63.3	-	-	-	-	-	**71.8**	-	-	-
DeepLatent-SFT-7B	84.3	76.4	73.6	63.1	52.4	35.0	85.3	84.2	63.6	71.0	24.8	37.9	14.4
DeepLatent-RL-7B*	85.3	**77.9**	**74.2**	64.1	54.1	37.7	86.2	85.9	64.1	70.6	24.5	37.3	14.9
DeepLatent-RL-7B	84.8	77.2	74.0	64.2	53.7	**38.7**	86.4	**86.4**	65.0	71.5	**25.3**	**39.0**	**16.2**
Δ (*vs* Qwen2.5-VL-7B)	+8.3	+8.4	+8.7	+6.9	+9.4	+14.8	+0.0	+0.2	+3.1	+4.1	+5.3	+2.0	+3.3

Table 1: Performance comparison of different benchmarks. Best results are in **bold**, and the Δ row reports absolute gains of DeepLatent-RL-7B over Qwen2.5-VL-7B. DeepLatent* uses only visual search data of DeepLatent-180K in SFT Stage 2, while DeepLatent uses the full DeepLatent-180K dataset.

2024), as shown in Tab. 2. DeepLatent achieves 68.0% on Jigsaw and 59.4% on Multi-view, outperforming the best compared methods by 16.0% and 8.3%, respectively. On VSP, it reaches 83.7%, surpassing Mirage by 7.7%, demonstrating stronger spatial path reasoning ability.

Model	Jigsaw	Multi-view	VSP
Qwen2.5-VL-7B	52.0	45.9	13.5
InternVL3-8B	50.0	51.1	-
Pixel-Reasoner-7B	52.7	-	-
LVR-7B	52.0	46.6	-
Monet-7B	50.0	47.4	-
Mirage-7B	-	-	76.0
DeepLatent-RL-7B	68.0	59.4	83.7

Table 2: Performance on spatial visual reasoning tasks.

RL Gains. DeepLatent-SFT already provides a strong starting point, surpassing most prior methods, and the RL stage further improves most benchmarks. It brings a 3.7% gain on VisualProbe-Hard, improves MMStar and RWQA by 1.0% on average, and raises the three visual reasoning benchmarks by 1.1% on average. These gains show that reinforcement learning further strengthens DeepLatent after supervised training. Overall, DeepLatent demonstrates that the latent reasoning paradigm can reach the effectiveness of prior tool-augmented methods.

Visual Thought Efficiency. We compare visual thought generation efficiency in Tab. 3 under the same hardware and configuration. DeepLatent achieves a TPLT of 1.0ms and a total latent generation time of 86.1ms, 35× faster per latent token and 2.6× faster in total than Monet. This comes from generating all latent tokens in parallel within a single forward pass, even though DeepLatent produces 12.7× as many latent tokens per sample. Compared with tool-based methods, DeepLatent is over 27× faster than Thyme (2375ms) and over 38× faster than DeepEyes (3324ms).

Model	TPLT	Latent Tokens	Thought Time
Thyme-7B	-	-	2375ms
DeepEyes-7B	-	-	3324ms
Monet-7B	34.9ms	6.5	227ms
DeepLatent-RL-7B	1.0ms	82.7	86.1ms

Table 3: Visual thought generation efficiency on 200 samples. TPLT denotes time per latent token. Latent Tokens denotes average tokens per latent block. Thought Time measures one visual thought step: a tool call or complete latent block generation time.

4.3 Ablation Study

How Visual Latent Helps. We compare four SFT methods in Tab. 4: *Text-only CoT* removes intermediate images and keeps textual reasoning; *Visual-only CoT* keeps intermediate images and answers only; *Interleaved CoT (w/o distill)* uses interleaved data without latent distillation loss; and *Interleaved CoT* adds latent distillation. All variants improve high-resolution perception over Qwen2.5-VL-7B, while *Interleaved CoT* achieves the best results across all metrics, improving V* from 76.5% to 84.3% and HR4K from 68.8% to 76.4%. *Text-only CoT* consistently outperforms *Visual-only CoT*, showing that visual latents work best when coupled

with textual CoT rather than used alone. Removing distillation also degrades performance, confirming the importance of distillation supervision.

Configuration	V*	HR4K	MME-RW-Lite	MMStar
Qwen2.5-VL-7B	76.5	68.8	44.3	61.9
+ Text-only CoT	81.3	72.3	50.2	63.2
+ Visual-only CoT	80.7	71.1	47.5	57.5
+ Interleaved CoT†	82.1	73.4	51.3	63.3
+ Interleaved CoT	84.3	76.4	52.4	63.6

Table 4: Ablation study of CoT modality for DeepLatent-SFT. †: without latent distillation loss.

Parallel vs. Autoregressive Latent. Fig. 4 compares two latent generation paradigms and two scaling strategies. *Dynamic* predicts the latent size (H_z, W_z), and *Fixed* always uses a preset maximum size. Parallel generation improves with more latent tokens and nearly saturates at 256 tokens. Autoregressive generation peaks near 16 tokens, but additional tokens rapidly degrade performance, whereas parallel generation remains stable even with 576 tokens. This supports our proposed parallel latent generation. For scale allocation, *Fixed* is slightly better at small budgets, while *Dynamic* performs better at larger sizes by preserving the assistant image aspect ratio.

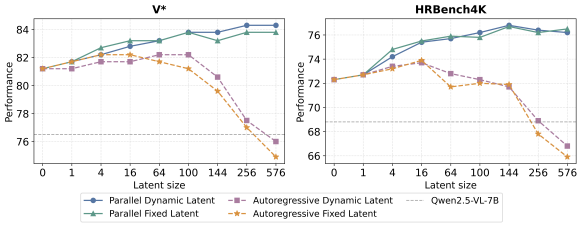


Figure 4: Parallel vs. autoregressive generation and dynamic vs. fixed scale under varying latent sizes.

Fig. 5 shows latent-to-image attention visualization at layers 8, 16, and 24 of LLM. DeepLatent consistently focuses on the queried ATTENTION sign across layers, while Monet spreads attention over background regions. This suggests that image-grounded parallel latents preserve clearer spatial evidence for fine-grained visual reasoning.

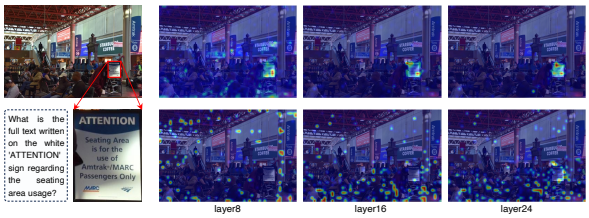


Figure 5: Latent-to-image attention visualization across LLM layers. Top: DeepLatent. Bottom: Monet.

LatentFormer Parameters. We examine the LatentFormer depth and modulation parameters on DeepLatent-SFT, as shown in Tab. 5. Removing LatentFormer feeds learnable latents directly into the LLM without directly extracting image features, causing drops on HR4K (-1.7%) and VisuLogic (-2.1%). One layer brings most of the performance improvement, while adding a second layer provides a slight gain. Between the two modulation parameters, latent modulation is more critical, causing an average drop of 1.4% when removed, while removing image modulation leads to 0.7% degradation.

Configuration	V*	HR4K	MMStar	VisuLogic
0 (w/o LatentFormer)	83.2	74.7	62.8	22.7
1 layer (default)	84.3	76.4	63.6	24.8
2 layers	84.3	76.8	63.7	25.0
w/o latent modulation	82.7	75.1	62.9	22.9
w/o image modulation	83.2	76.0	63.4	23.6

Table 5: Ablation study of LatentFormer parameters.

Continuous-Space GRPO. We compare RL optimization strategies in Tab. 6. *Text GRPO* applies GRPO only to text and `<latent>` tokens, while *Joint GRPO* additionally optimizes continuous latent modulation. Text GRPO brings no consistent improvement, and updating LatentFormer further degrades performance by 1.0% on average, suggesting that learning latent construction with standard GRPO may disrupt learned latent visual semantics. In contrast, Joint GRPO improves over DeepLatent-SFT by 0.7% on average, with the largest gain on MMStar (+1.4%). Freezing LatentFormer during Joint GRPO yields more stable gains.

Configuration	V*	HR4K	MMStar	VisuLogic
DeepLatent-SFT	84.3	76.4	63.6	24.8
+ Text GRPO	83.8	76.5	63.4	24.6
+ Text GRPO†	83.2	75.3	63.1	23.6
+ Joint GRPO	84.8	76.8	65.0	25.3
+ Joint GRPO†	84.3	76.1	64.7	24.9

Table 6: Ablation experiments on RL optimization strategies. †: LatentFormer is not frozen in RL stage.

5 Conclusion

We presented DeepLatent, a parallel latent visual reasoning framework enabling VLMs to think with images in latent space. By generating image-grounded latents through LatentFormer and optimizing continuous modulation with reinforcement learning, DeepLatent improves visual perception and reasoning while preserving general multimodal understanding. We introduced DeepLatent-180K for latent visual reasoning. Experiments show that DeepLatent outperforms prior latent methods, reaches or exceeds tool-augmented models, and provides more efficient visual thought generation.

Limitations

We summarize the limitations as follows:

(1) DeepLatent achieves strong results on image-based visual reasoning benchmarks. However, it has not yet been extended to video reasoning. Videos introduce temporal dynamics, event dependencies, and fine-grained motion changes. These factors require latent states to evolve across both space and time. In future work, we plan to build fine-grained video reasoning datasets. This will help fill the gap in latent reasoning for dynamic visual scenarios.

(2) The training pipeline is still relatively complex. Compared with methods such as Monet, DeepLatent simplifies training through parallel latent generation and direct alignment. Still, it requires two SFT alignment stages and one RL stage. This increases training cost and implementation effort. Encouragingly, our SFT model already surpasses most previous methods. It also performs well without RL, showing that RL is an effective enhancement rather than a strict requirement.

References

Ashutosh Adhikari and Mirella Lapata. 2026. Multimodal latent reasoning via predictive embeddings. *arXiv preprint arXiv:2604.08065*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Yiping Bao, and 1 others. 2026. Babyvision: Visual reasoning beyond language. *arXiv preprint arXiv:2601.06521*.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087.

Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*.

Ethan Chern, Zhulin Hu, Steffi Chern, Siqi Kou, Jiadi Su, Yan Ma, Zhijie Deng, and Pengfei Liu. 2025. Thinking with generated images. *arXiv preprint arXiv:2505.22525*.

Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. 2024. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*.

Xu Chu, Xinrong Chen, Guanyu Wang, Zhijie Tan, Kui Huang, Wenyu Lv, Tong Mo, and Weiping Li. 2025. Qwen look again: Guiding vision-language reasoning models to re-attention visual information. *arXiv preprint arXiv:2505.23558*.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, and 1 others. 2025. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*.

Shuai Dong, Siyuan Wang, Xingyu Liu, Chenglin Li, Haowen Hou, and Zhongyu Wei. 2025. Interleaved latent visual reasoning with selective perceptual modeling. *arXiv preprint arXiv:2512.05665*.

Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. 2025. Grit: Teaching mllms to think with images. *arXiv e-prints*, pages arXiv–2505.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer.

Xingyu Fu, Minqian Liu, Zhengyuan Yang, John Corring, Yijuan Lu, Jianwei Yang, Dan Roth, Dinei Florencio, and Cha Zhang. 2025. Refocus: Visual editing as a chain of thought for structured image understanding. *arXiv preprint arXiv:2501.05452*.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14953–14962.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

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-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. 2024a. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*, 37:139348–139379.

Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. 2024b. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9590–9601.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Xu Tang, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

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*.

Byungwoo Jeon, Yoonwoo Jeong, Hyunseok Lee, Minsu Cho, and Jinwoo Shin. 2026. Vision-aligned latent reasoning for multi-modal large language model. *arXiv preprint arXiv:2602.04476*.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, and 1 others. 2023. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026.

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. 2025. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. *arXiv preprint arXiv:2509.07969*.

Ang Li, Charles Wang, Deqing Fu, Kaiyu Yue, Zikui Cai, Wang Bill Zhu, Ollie Liu, Peng Guo, Willie Neiswanger, Furong Huang, and 1 others. 2025a. Zebra-cot: A dataset for interleaved vision language reasoning. *arXiv preprint arXiv:2507.16746*.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. 2025b. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025c. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*.

Geng Li, Jinglin Xu, Yunzhen Zhao, and Yuxin Peng. 2025d. Dyfo: A training-free dynamic focus visual search for enhancing lmms in fine-grained visual understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9098–9108.

Jiaze Li, Jingyang Chen, Yuxun Qu, Shijie Xu, Zhenru Lin, Junyou Zhu, Boshen Xu, Wenhui Tan, Pei Fu, Jianzhong Ju, and 1 others. 2025e. Xiaomi mimo-vl-miloco technical report. *arXiv preprint arXiv:2512.17436*.

You Li, Chi Chen, Yanghao Li, Fanhu Zeng, Kaiyu Huang, Jinan Xu, and Maosong Sun. 2026. Imagination helps visual reasoning, but not yet in latent space. *arXiv preprint arXiv:2602.22766*.

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

Bo Lv, Yasheng Sun, Junjie Wang, and Haoxiang Shi. 2026. Onelatent: Single-token compression for visual latent reasoning. *arXiv preprint arXiv:2602.13738*.

Jizheng Ma, Xiaofei Zhou, Geyuan Zhang, Yanlong Song, and Han Yan. 2025. Multimodal reasoning via latent refocusing. *arXiv preprint arXiv:2511.02360*.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706.

OpenAI. 2025. Thinking with images. https://openai.com/index/thinking-with-images/. Accessed: 2025.

Tan-Hanh Pham and Chris Ngo. 2025. Multimodal chain of continuous thought for latent-space reasoning in vision-language models. *arXiv preprint arXiv:2508.12587*.

Ji Qi, Ming Ding, Weihan Wang, Yushi Bai, Qingsong Lv, Wenyi Hong, Xu Bin, Lei Hou, Juanzi Li, Yuxiao Dong, and 1 others. 2025. Cogcom: A visual language model with chain-of-manipulations reasoning. In *International Conference on Learning Representations*, volume 2025, pages 11090–11110.

Yiming Qin, Bomin Wei, Jiaxin Ge, Konstantinos Kallidromitis, Stephanie Fu, Trevor Darrell, and XuDong Wang. 2025. Chain-of-visual-thought:

Teaching vlms to see and think better with continuous visual tokens. *arXiv preprint arXiv:2511.19418*.

Arijit Ray, Ahmed Abdelkader, Chengzhi Mao, Bryan A Plummer, Kate Saenko, Ranjay Krishna, Leonidas Guibas, and Wen-Sheng Chu. 2025. Mull-tokens: Modality-agnostic latent thinking. *arXiv preprint arXiv:2512.10941*.

Rohit Saxena, Pasquale Minervini, and Frank Keller. 2025. Postersum: A multimodal benchmark for scientific poster summarization. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1828–1844.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024a. Visual cot: Advancing multimodal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024b. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, and 1 others. 2025. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11888–11898.

Chameleon Team. 2024. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*.

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, and 1 others. 2025a. Traceable evidence enhanced visual grounded reasoning: Evaluation and methodology. *arXiv preprint arXiv:2507.07999*.

Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. 2025b. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*.

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

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. 2025c. Monet: Reasoning in latent visual space beyond images and language. *arXiv preprint arXiv:2511.21395*.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. 2025d. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7907–7915.

Lai Wei, Liangbo He, Jun Lan, Lingzhong Dong, Yutong Cai, Siyuan Li, Huijia Zhu, Weiqiang Wang, Linghe Kong, Yue Wang, and 1 others. 2026. Zooming without zooming: Region-to-image distillation for fine-grained multimodal perception. *arXiv preprint arXiv:2602.11858*.

Jialong Wu, Xiaoying Zhang, Hongyi Yuan, Xiangcheng Zhang, Tianhao Huang, Changjing He, Chaoyi Deng, Renrui Zhang, Youbin Wu, and Mingsheng Long. 2026a. Visual generation unlocks human-like reasoning through multimodal world models. *arXiv preprint arXiv:2601.19834*.

Linquan Wu, Tianxiang Jiang, Yifei Dong, Haoyu Yang, Fengji Zhang, Shichaang Meng, Ai Xuan, Linqi Song, and Jacky Keung. 2026b. Lavit: Aligning latent visual thoughts for multi-modal reasoning. *arXiv preprint arXiv:2601.10129*.

Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, Chengxiang Zhai, and Klara Nahrstedt. 2025. Vtool-r1: Vlms learn to think with images via reinforcement learning on multimodal tool use. *arXiv preprint arXiv:2505.19255*.

Penghao Wu and Saining Xie. 2024. V*: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094.

Qiucheng Wu, Handong Zhao, Michael Saxon, Trung Bui, William Yang Wang, Yang Zhang, and Shiyu Chang. 2024. Vsp: Assessing the dual challenges of perception and reasoning in spatial planning tasks for vlms. *arXiv preprint arXiv:2407.01863*.

xAI. 2024. Grok-1.5 vision preview. https://x.ai/news/grok-1.5v.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. 2025a. Llava-cot: Let vision language models reason step-by-step. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2087–2098.

Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, and 1 others. 2025b. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. *arXiv preprint arXiv:2504.15279*.

Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. 2025c. Visual planning: Let's think only with images. *arXiv preprint arXiv:2505.11409*.

Shuo Yang, Yuwei Niu, Yuyang Liu, Yang Ye, Bin Lin, and Li Yuan. 2026. Look-back: Implicit visual re-focusing in mllm reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 11694–11702.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. 2025. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, and 1 others. 2025a. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. *arXiv preprint arXiv:2509.18154*.

Xinlei Yu, Chengming Xu, Guibin Zhang, Zhangquan Chen, Yudong Zhang, Yongbo He, Peng-Tao Jiang, Jiangning Zhang, Xiaobin Hu, and Shuicheng Yan. 2025b. Vismem: Latent vision memory unlocks potential of vision-language models. *arXiv preprint arXiv:2511.11007*.

Huanyu Zhang, Wenshan Wu, Chengzu Li, Ning Shang, Yan Xia, Yangyu Huang, Yifan Zhang, Li Dong, Zhang Zhang, Liang Wang, and 1 others. 2025a. Latent sketchpad: Sketching visual thoughts to elicit multimodal reasoning in mllms. *arXiv preprint arXiv:2510.24514*.

Lu Zhang, Jiazuo Yu, Haomiao Xiong, Ping Hu, Yunzhi Zhuge, Huchuan Lu, and You He. 2025b. Finers: Fine-grained reasoning and segmentation of small objects with reinforcement learning. *arXiv e-prints*, pages arXiv–2510.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, and 1 others. 2025c. Thyme: Think beyond images. *arXiv preprint arXiv:2508.11630*.

YiFan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, and 1 others. 2025d. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? In *International Conference on Learning Representations*, volume 2025, pages 89655–89701.

Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Ming Li, Qilong Wu, Kaipeng Zhang, and Chen Wei. 2025. Pyvision: Agentic vision with dynamic tooling. *arXiv preprint arXiv:2507.07998*.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. 2025. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Appendix

This appendix provides supplementary materials in four parts. Sec. A.1 summarizes released resources, implementation settings, and RL details. Sec. A.2 reports additional evaluations and ablations. Sec. A.3 compares DeepLatent with related image-thinking and unified multimodal paradigms. Sec. A.4 describes dataset construction and qualitative inference examples.

A.1 Reproducibility and Training Details

A.1.1 Code, Model, and Data

Our code is available at `https://anonymous.4open.science/r/anonymous-emnlp3267/`. Due to the large size of model checkpoints and training data, they are hosted on an anonymous Hugging Face repository linked in the above project page. Please refer to `readme.md` for instructions on training, evaluation, and data usage.

A.1.2 Implementation Details

This section summarizes the main implementation and training hyperparameters used in our supervised fine-tuning (SFT) and reinforcement learning (RL) stages. We use Qwen2.5-VL-7B as the base model and train all models on 8× NVIDIA A800 GPUs. The SFT pipeline contains two stages: Stage 1 trains on interleaved image-text trajectories with explicit assistant images, and Stage 2 replaces these assistant images with latent blocks for latent distillation. Both SFT stages are trained with the `trl` framework, while RL training is conducted with the `verl` framework. We freeze the visual encoder in all training pipelines.

Hyperparameter	Value
learning rate	0.00001
batch size	1
gradient accumulation steps	16
weight decay	0.01
SFT Stage 1 steps	1 epoch
SFT Stage 2 steps	1 epoch
Max latent size	16×16
Max latent blocks	5
Stage 1 max question image pixels	$6000 \times 28 \times 28$
Stage 2 max question image pixels	$2500 \times 28 \times 28$
Stage 2 alignment loss weight	4.0
Stage 2 scale loss weight	0.5

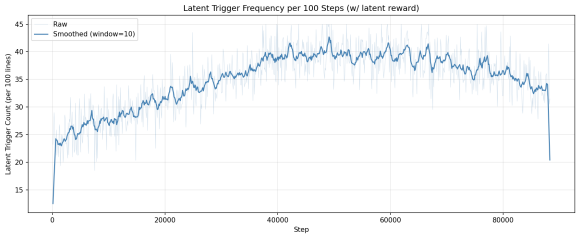
Table 7: Hyperparameters for SFT.

Hyperparameter	Value
learning rate	0.000001
batch size	64
weight decay	0.01
rollout size	8
temperature	0.5
max response length	2048
training samples	10,000
training steps	1 epoch
Max latent size	16×16
Max latent blocks	5
max question image pixels	$2500 \times 28 \times 28$
accuracy threshold	0.6

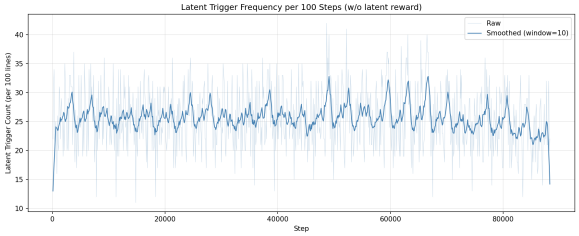
Table 8: Hyperparameters for RL.

Hyperparameter	Value
temperature	0.5
top_p	0.5
top_k	5
max_new_tokens	2048
max_image_pixels	$16384 \times 28 \times 28$

Table 9: VLMEvalKit evaluation configuration.



(a) With latent invocation reward.



(b) Without latent invocation reward.

Figure 6: Number of active latent invocations during RL training with and without the latent invocation reward.

A.1.3 Additional RL Details

Reward design. We use a weighted reward composed of an accuracy reward and a format reward. The accuracy reward is 1 if the final answer is correct and 0 otherwise. In our implementation, correctness is judged by a large VLM. The format reward encourages the model to use the `boxed{}` answer format. The final reward is computed as:

$$R = 0.9 \cdot R_{\text{Acc}} + 0.1 \cdot R_{\text{Format}}, \qquad (12)$$

where $R_{\text{Acc}} \in \{0, 1\}$ denotes the accuracy reward and $R_{\text{Format}} \in \{0, 1\}$ denotes the format reward.

We additionally test a latent-invocation reward that encourages active use of the latent interface. If a sampled response emits the `<latent>` token (a regular token in the text vocabulary), this reward is 1; otherwise, it is 0. This term is only used in the ablation below and is not included in the final reward in Eq. (1). We compare trigger counts with and without this term in Fig. 6.

The latent invocation reward clearly increases the model's tendency to use latent reasoning during RL. With the reward, the smoothed trigger count rises from about 25 to around 40 per 100 samples and remains at a high level for most of training. Without the reward, the trigger count stays mostly around 25 and shows no clear upward trend. In both settings, the trigger count shows a mild decreasing trend near the end of training. Although the latent invocation reward encourages more frequent latent usage, it does not lead to actual performance improvement. Instead, it often causes the model to generate multiple meaningless latent blocks within a single sampled response. Therefore, we do not adopt the latent invocation reward in the final RL setting.

Continuous-action ratio and KL. For continuous modulation actions, we compute policy ratios with diagonal Gaussian densities. Let $\mathbf{a}^{\text{old}}$ denote a rollout action (e.g., $\tilde{\mathbf{L}}^{\text{old}}$ or $\tilde{\mathbf{I}}^{\text{old}}$). The current and old policies define:

$$\begin{aligned} \pi_\theta(\mathbf{a}) &= \mathcal{N}(\mathbf{a}; \boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta^2), \\ \pi_{\text{old}}(\mathbf{a}) &= \mathcal{N}(\mathbf{a}; \boldsymbol{\mu}_{\text{old}}, \boldsymbol{\sigma}_{\text{old}}^2). \end{aligned} \qquad (13)$$

The ratio for each sampled action is:

$$r_t(\theta) = \exp\left(\log \pi_\theta(\mathbf{a}^{\text{old}}) - \log \pi_{\text{old}}(\mathbf{a}^{\text{old}})\right), \qquad (14)$$

with

$$\log \pi(\mathbf{a}) = \sum_j \log \mathcal{N}(a_j; \mu_j, \sigma_j^2). \quad (15)$$

Here, the action is the modulated output, while the Gaussian mean is parameterized by base variables. For latent-query modulation, $\boldsymbol{\mu}_\theta = \boldsymbol{\gamma}_{\theta,l} \odot \mathbf{L} + \boldsymbol{\beta}_{\theta,l}$; for image-feature modulation, $\boldsymbol{\mu}_\theta = \boldsymbol{\gamma}_{\theta,i} \odot \mathbf{I}_0 + \boldsymbol{\beta}_{\theta,i}$, following Eq. (3).

The KL term between the current policy and the SFT reference policy is computed in closed form:

$$\mathbb{D}_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}) = \frac{1}{2} \sum_j \left[\log \frac{\sigma_{\mathrm{ref},j}^2}{\sigma_{\theta,j}^2} + \frac{\sigma_{\theta,j}^2}{\sigma_{\mathrm{ref},j}^2} + \frac{(\mu_{\theta,j} - \mu_{\mathrm{ref},j})^2}{\sigma_{\mathrm{ref},j}^2} - 1 \right]. \quad (16)$$

We compute this KL per continuous action position and then average over positions and samples. For discrete text tokens (including <latent>, which is in the shared vocabulary), we use the standard categorical ratio and KL.

A.2 Additional Evaluation and Ablations

A.2.1 Low-Resolution Evaluation

In the main experiments, we set the maximum input budget to $16384 \times 28 \times 28$ pixels. While this setting preserves rich visual details, it may be less practical under strict compute or memory constraints. We therefore further evaluate both models at lower input budgets.

We set the maximum input pixels to $512 \times 28 \times 28$, $1024 \times 28 \times 28$, and $2048 \times 28 \times 28$, and evaluate Qwen2.5-VL-7B and DeepLatent on V* and HRBench4K. We choose these benchmarks because they are sensitive to fine-grained perception and high-resolution evidence. Results are shown in Fig. 7.

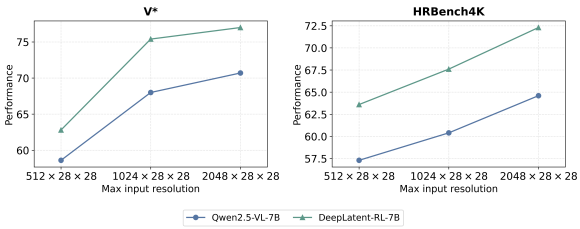


Figure 7: Performance comparison under different maximum input resolutions on V* and HRBench4K.

As the input budget increases, both models improve on V* and HRBench4K, confirming the dependence of these benchmarks on fine-grained visual evidence. DeepLatent consistently outperforms Qwen2.5-VL-7B across all tested budgets, with a clear margin in low-budget settings (e.g., $512 \times 28 \times 28$). The advantage persists at higher budgets, indicating that latent visual reasoning is effective in both compute-constrained and high-budget regimes.

A.2.2 SFT Data Ablation Study

Tab. 10 reports an ablation of SFT training data sources under the DeepLatent-SFT setting. DeepLatent-180K achieves the best results on HR4K and VisuLogic, improving over Thyme-SFT by 1.3% and 0.6%, respectively. On V*, it is on par with Thyme-SFT, while on MMStar it is slightly lower but still comparable. Monet-SFT-125K performs worse on all four benchmarks, suggesting that its supervision is less effective for parallel latent construction. Overall, this SFT-stage ablation indicates that the quality and coverage of DeepLatent-180K are important for learning useful intermediate visual states before RL.

Training Data	V*	HR4K	MMStar	VisuLogic
Thyme-SFT	84.3	75.1	63.9	24.2
Monet-SFT-125K	83.8	74.7	63.1	22.7
DeepLatent-180K	84.3	76.4	63.6	24.8

Table 10: SFT-stage data ablation across different training sources under the DeepLatent-SFT setting.

A.2.3 Adaptation to More Models

To further examine the generality of DeepLatent, we adapt it to Qwen3-VL-4B and Qwen3-VL-8B backbones. Since Qwen3-VL produces visual features from multiple encoder layers, we concatenate them along the channel dimension and apply an MLP projection to match the LatentFormer input dimension. This keeps the LatentFormer interface unchanged and requires minimal architectural modification.

As shown in Tab. 11, DeepLatent brings consistent improvements over the corresponding Qwen3-VL backbones, with the largest gains on high-resolution and fine-grained perception tasks. DeepLatent-RL-4B improves over Qwen3-VL-4B by up to 14.1% (VisualProbe-H) and averages +8.7% across V*, HR4K, HR8K, MME-RW, and VisualProbe-H. DeepLatent-RL-8B shows a similar pattern, averaging +8.0% on the same five benchmarks. On MMStar and BabyVision, the adapted models match or exceed all compared baselines, confirming that the latent interface generalizes across model scales without degrading general

Model	V*	HR4K	HR8K	MME-RW	VisualProbe-H	MMStar	BabyVision
GPT-4o (Hurst et al., 2024)	66.0	59.0	55.5	-	23.6	-	-
Qwen3-VL-4B (Bai et al., 2025a)	80.1	78.3	72.9	63.5	25.5	69.7	13.4
Qwen3-VL-8B (Bai et al., 2025a)	86.4	78.9	74.6	65.8	27.4	70.9	13.8
MiMo-VL-7B-RL (Li et al., 2025e)	81.2	74.4	72.9	63.4	-	73.5	16.2
MiniCPM-V-4.5-9B (Yu et al., 2025a)	70.2	69.9	63.6	58.2	-	67.9	15.0
DeepLatent-RL-4B	91.6	82.5	81.6	68.3	39.6	71.3	16.8
DeepLatent-RL-8B	92.7	84.7	83.5	69.6	42.5	73.5	17.0

Table 11: Performance of DeepLatent adapted to Qwen3-VL architecture.

multimodal understanding.

A.2.4 Math Reasoning Evaluation

We further evaluate DeepLatent-RL-7B on mathematical visual reasoning benchmarks (MathVista, MathVision, and WeMath) and compare with Qwen2.5-VL-7B, InternVL3-8B, Thyme-7B, and DeepEyes-7B.

As shown in Tab. 12, DeepLatent-RL-7B achieves competitive performance on all three benchmarks without degradation relative to its base model. Math problems are predominantly solved through reasoning in the language space (symbolic manipulation, multi-step calculation), where latent visual internalization offers limited additional benefit. This is consistent with DeepLatent's design goal of enhancing visually grounded perception rather than language-internal reasoning.

Model	MathVista	MathVision	WeMath
Qwen2.5-VL-7B	68.2	27.0	34.3
InternVL3-8B	70.4	26.3	31.7
Thyme-7B	70.0	27.6	39.3
DeepEyes-7B	70.1	26.6	38.9
DeepLatent-RL-7B	69.3	26.9	38.7

Table 12: Math reasoning evaluation compared with visual reasoning baselines on MathVista, MathVision, and WeMath.

A.3 Comparison with Related Paradigms

A.3.1 Comparison with Non-Image-Thinking Methods

Recent works explore alternative approaches to visual reasoning without explicit image thinking. CapImagine (Li et al., 2026) leverages caption-based imagination to enrich textual reasoning with visual context. ZWZ (Wei et al., 2026) distills region-level perception from a local-detail teacher into global-image inference. Both methods convert local visual evidence into fine-grained textual reasoning chains.

We view this direction as complementary to image thinking. Textual chains can improve fine-grained perception when the required evidence is explicitly describable. DeepLatent instead revisits the image through latent visual states, maintaining an internal visual interface that supports not only fine-grained perception but also tasks requiring spatial reasoning or multi-step visual evidence aggregation, such as complex path planning and multi-view spatial transformation.

Model	V*	HR4K	HR8K
CapImagine-7B	85.9	74.1	70.7
ZWZ-7B	88.5	75.4	73.3
DeepLatent-RL-7B	84.8	77.2	74.0

Table 13: Comparison with non-image-thinking methods on high-resolution perception benchmarks.

Model	Jigsaw	MMStar	BabyVision
CapImagine-7B	64.7	-	-
ZWZ-7B	-	63.4	16.0
DeepLatent-RL-7B	68.0	65.0	16.2

Table 14: Comparison with non-image-thinking methods on visual reasoning benchmarks.

As shown in Tab. 13, text-chain-based methods are competitive on perception-centric benchmarks, with ZWZ-7B obtaining the best V* score. However, DeepLatent-RL-7B achieves stronger results on HR4K and HR8K, where models must aggregate fine-grained evidence from high-resolution images. This indicates that revisiting the image through latent visual states is particularly beneficial when detailed visual evidence needs to be preserved across reasoning steps.

Tab. 14 further highlights the benefit of image-grounded latent reasoning. DeepLatent-RL-7B outperforms CapImagine-7B on Jigsaw and surpasses ZWZ-7B on MMStar, while remaining comparable on BabyVision. These results suggest that converting local details into textual chains is effective for perception, but less sufficient for tasks requir-

ing spatial reasoning or multi-step visual evidence aggregation.

A.3.2 Discussion on Unified Multimodal Generation

Another related direction is interleaved image-text generation with unified multimodal models. These methods can generate pixel-level intermediate images during reasoning, usually by alternating between language generation and image generation through diffusion heads or discrete visual tokenizers (Team, 2024; Deng et al., 2025; Chern et al., 2024; Li et al., 2025c; Chern et al., 2025; Xu et al., 2025c; Li et al., 2025a; Wu et al., 2026a). They are closely related to the broader goal of thinking with images, but they are more naturally categorized as unified multimodal generation models rather than conventional VLMs. For this reason, we do not make them the main focus of the comparison in the main text.

These approaches may ultimately achieve richer visual imagination because they explicitly materialize intermediate visual states in pixel space. However, current image generation procedures introduce substantial inference latency and require more computation than constructing compact latent states. DeepLatent instead focuses on a lightweight latent interface for VLMs, preserving the benefit of intermediate visual reasoning with only a single forward pass through the LatentFormer.

A.4 Dataset and Qualitative Examples

A.4.1 Training Dataset Construction

DeepLatent-180K provides supervision for latent visual reasoning. It is built from two sources: curated interleaved reasoning trajectories and newly generated high-resolution visual search data.

Curation from existing datasets. We collect samples from Zebra-CoT and Thyme-SFT. We keep samples whose assistant images can be derived from the question image through geometric annotations or object manipulations (e.g., drawing paths in mazes or moving target objects on board-like scenes). We discard samples that create new scenes or unrelated visual content. Each trajectory is limited to at most five assistant images.

High-resolution visual search data. We collect high-resolution images from SAM-1B, FineRS, InfoVQA, and other datasets. Most selected images have a resolution larger than 1,000,000 pixels and contain rich local details.

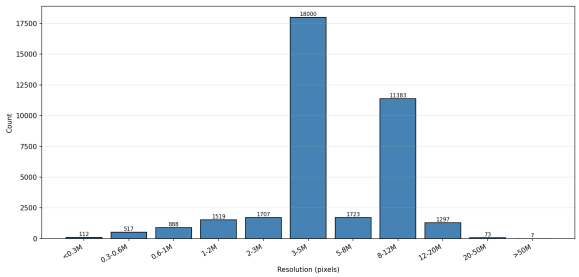


Figure 8: Resolution distribution of high-resolution images in DeepLatent-180K.

For each selected image, we use Qwen3-VL-32B to generate multiple-choice question-answer pairs focused on local details. The questions are designed to require precise localization rather than a quick global glance. We use the following prompt for generation:

> **Role.** You are an expert Visual Reasoning Examiner. Your task is to analyze an image and generate a high-quality, challenging multiple-choice question based on fine-grained visual details.
>
> **Objective.** Generate a question that tests fine-grained visual perception. The question should focus on details such as textures, background objects, subtle interactions, precise counts, or geometric patterns. The answer should require zooming in or carefully inspecting a specific image region.
>
> **Requirements.** The question should not be limited to fixed categories. It should target unique or easily overlooked details. Each question must include five options, A–E. Option E must be a negative option, such as "All the options above are incorrect." The correct answer can be A–D, or E for trick questions. The probability of E being correct should be no less than 10%.
>
> **Constraints.** The question should require localization and precision. Avoid generic questions about the main subject. Do not ask about specific words or numbers in the image. Do not ask subjective questions about emotions or intentions. If the image is too blurry, too dark, or lacks distinct fine-grained details, answer E.
>
> **Output Format.** Return a JSON object with the following fields: `visual_target`, `question`, `options`, `correct_answer`, and `reasoning`.

Filtering and trajectory generation. We verify generated question-answer pairs with three strong VLMs: Qwen3-VL-235B-A22B, InternVL3-78B, and GLM-4.5V (Hong et al., 2025). Only samples with at least 2 out of 3 correct judgments are kept. We then use Thyme-7B to generate interleaved trajectories with intermediate thinking images. Finally, we apply the three-stage filtering in Sec. 3.4 to all collected data, including both curated existing data and newly generated data. The remaining

high-quality samples are merged to form the final DeepLatent-180K training dataset.

Fig. 9 illustrates representative examples from DeepLatent-180K, covering visual perception, bounding box highlight, and jigsaw reasoning samples. They show how the dataset provides supervision for locating fine-grained visual evidence and constructing useful intermediate visual states during reasoning.

A.4.2 DeepLatent Inference Examples

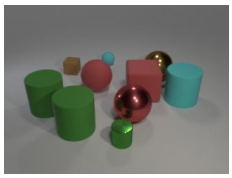
We provide inference examples to illustrate the behavior of DeepLatent. As shown in Fig. 10, the model invokes intermediate latent visual states during reasoning and uses them to support the final answer. The examples cover high-resolution visual perception, maze spatial planning, chart understanding, and jigsaw tasks.

User: What is the aircraft registration number visible on the rear fuselage of the Scandinavian Airlines plane? \nPut your final answer within \\boxed{}.

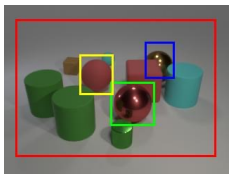
Assistant: The registration number is relatively small and may not be clearly visible in the original image due to its size and distance from the camera. To accurately read the text, it is necessary to zoom in and crop the specific region around the registration number. To achieve this, I will crop the image around the area where the registration number is located, which appears to be near the rear section of the fuselage.<latent></latent>\n

Assistant: By focusing on the cropped image, the registration number becomes more legible, allowing us to confirm that the aircraft registration number on the rear fuselage of the Scandinavian Airlines plane is LN-RCZ.
The answer is \\boxed{LN-RCZ}.

User: Subtract all shiny spheres. Subtract all big red matte spheres. How many objects are left? \nPut your final answer within \\boxed{}.

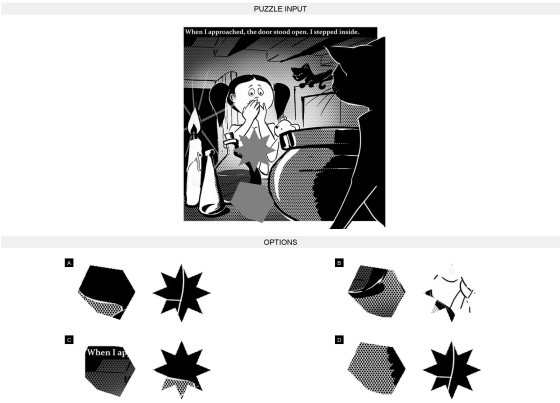


Assistant: First, locate the positions of all objects, luminous spheres, and a large matte red sphere. Then observe the quantity of each. \n<latent></latent>\n

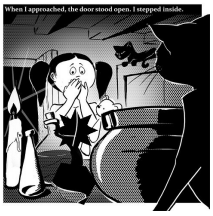


Assistant: The observed total number of objects is 10, with 2 luminous spheres and 1 large matte red sphere.
The number of objects left after subtracting all the luminous spheres and all the matte red spheres is 7 (10 - 2 - 1 = 7).
The answer is \\boxed{7}.

User: A multi-piece jigsaw puzzle has missing sections. Identify which set of pieces from the options provided correctly fills the gaps in the puzzle image.\nPut your final answer within \\boxed{}.

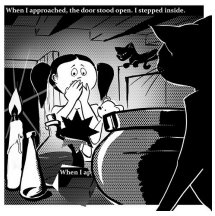


Assistant: Identify the shapes and locations of the missing pieces in the main puzzle image.\n Evaluate each option by considering if its two pieces match the shape and content of the missing areas. Visualize fitting the pieces. \n<latent></latent>\n"



Assistant: The first visual test shows that the star piece from option A fits the star-shaped gap. \n<latent></latent>\n"

Assistant: This image shows that the hexagonal piece from option B fits the hexagonal gap. \n<latent></latent>\n



Assistant: This image demonstrates that the text piece from option C fits the text-shaped gap. \n<latent></latent>\n

Assistant: Option B provides the hexagonal piece and the text piece, which together perfectly fill the corresponding missing areas in the puzzle.\n
The answer is \\boxed{B}.

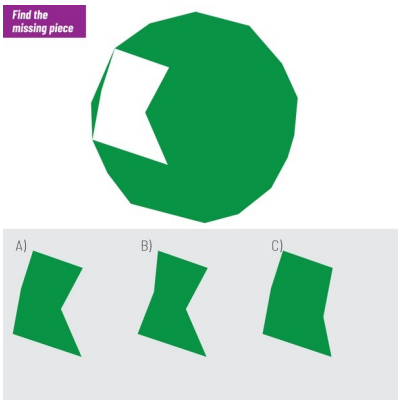
Figure 9: Examples from the DeepLatent-180K training dataset.

User: What is the color of the plastic stool? (A) black (B) blue (C) white (D) red\nPut your final answer within \\boxed{}.

Assistant: The stool is small and located near the bottom right corner, making its color difficult to determine without zooming. The image will be cropped to focus on the stool's location. \n <latent></latent>\n

Assistant: The cropped image shows a blue plastic stool near the bottom right corner. The answer is \\boxed{B}.

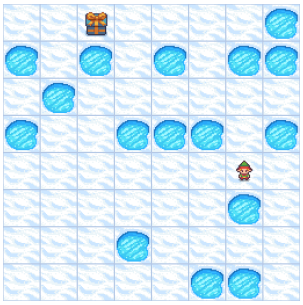
User: Select the appropriate option to fill in the blank and complete the figure.\nChoices:\n(A) A\n(B) B\n(C) C \nPut your final answer within \\boxed{}.



Assistant: The image shows a green circle with a white polygon cutout. The missing piece must fit this cutout. The options are three green pieces with varying shapes and sizes. The missing piece is a green polygon. Let's try putting A into the missing part . \n<latent></latent>\n

Assistant: Option A matches this description. The answer is \\boxed{A}

User: Find the minimum number of steps required for the little figure in the image to reach the box. Do not walk through the holes.\nPut your final answer within \\boxed{}.



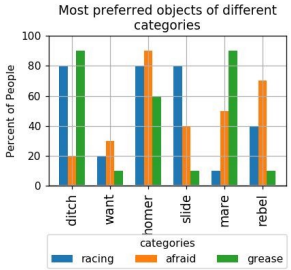
Assistant: In the diagram, the little figure is located in row 5, column 7, at coordinates (5, 7), while the box is located in row 1, column 3, at coordinates (1, 3). we will begin by moving upward.\n<latent></latent>\n

Assistant: After moving two steps upward, we reach (3, 7); at this point, the box is located to the upper left and needs to be moved leftward to (3, 4).\n<latent></latent>\n

Assistant: Next, Then, continue moving upward two steps to reach (1, 4).\n<latent></latent>\n

Assistant: At this point, we only need to take one step to the left to reach the box; therefore, we need to take a total of 2 + 3 + 2 + 1 = 8 steps. The answer is \\boxed{8}.

User: How many objects are preferred by more than 90 percent of people in at least one category? \nPut your final answer within \\boxed{}."



Assistant: The image displays a bar chart with categories and objects. The question asks how many objects are preferred by more than 90% in at least one category. The chart indicates no object exceeds 90% in any category. The answer is \\boxed{0}.

Figure 10: Examples of DeepLatent inference with intermediate latent visual reasoning.