Machine Mental Imagery: Empower Multimodal Reasoning with Latent Visual Tokens

Zeyuan Yang[1*] Xueyang Yu[1*] Delin Chen[1] Maohao Shen[2] Chuang Gan[1,3]
[1]University of Massachusetts Amherst [2]MIT [3]MIT-IBM Watson AI Lab

Abstract

*Vision-language models (VLMs) excel at multimodal understanding, yet their text-only decoding forces them to verbalize visual reasoning, limiting performance on tasks that demand visual imagination. Recent attempts train VLMs to render explicit images, but the heavy image-generation pretraining often hinders the reasoning ability. Inspired by the way humans reason with mental imagery—the internal construction and manipulation of visual cues—we investigate whether VLMs can reason through interleaved multimodal trajectories without producing explicit images. To this end, we present a Machine Mental Imagery framework, dubbed as **Mirage**, which augments VLM decoding with latent visual tokens alongside ordinary text. Concretely, whenever the model chooses to "think visually", it recasts its hidden states as next tokens, thereby continuing a multimodal trajectory without generating pixel-level images. Begin by supervising the latent tokens through distillation from ground-truth image embeddings, we then switch to text-only supervision to make the latent trajectory align tightly with the task objective. A subsequent reinforcement learning stage further enhances the multimodal reasoning capability. Experiments on diverse benchmarks demonstrate that Mirage unlocks stronger multimodal reasoning without explicit image generation.*

1. Introduction

Vision–language models (VLMs) jointly encode images and text and attain impressive results on visual-understanding benchmarks through text-only decoding [47]. Techniques such as chain-of-thought prompting and reinforcement-learning fine-tuning can lengthen these textual reasoning traces and yield extra gains. Nonetheless, VLMs still stumble on multimodal reasoning tasks such as spatial reasoning, which demand more than passive perception; they require a coherent understanding and manipulation of visual elements.

Consider the jigsaw puzzle in Fig. 1. Instead of textualizing every candidate piece, people picture how the two fragments might align and decide on the correct match. This reasoning unfolds in a native multimodal fashion, not through language alone. Recent studies [5, 10, 43, 45] have pretrained VLMs for large-scale image generation so a single model can produce both words and pictures. Yet the cognitive demands of logical reasoning differ sharply from the task of synthesizing pixels, and asking one model to master both goals often degrades its reasoning quality [46]. In addition, the image decoders cannot produce interleaved trajectories pertinent to input images. Consequently, fully exploiting the dormant multimodal reasoning capacity of VLMs remains an open challenge.

According to imagery theory, humans do not summon photorealistic pictures while thinking. We instead construct and manipulate mental images, simplified sketches that capture only task-relevant information, a process known as **mental imagery** [15, 23, 37]. In the jigsaw example, we examine fragment contours to decide whether two pieces fit. Likewise, when searching for misplaced keys, we recall the outline of the shelf edge rather than the full room. Inspired by this behavior, we ask whether VLMs can reason directly in their latent visual embedding space, weaving compact visual embeddings into the text stream and dispensing with the need for explicit image generation.

To this end, we present **Mirage**, a decoding mechanism that interleaves latent visual representations among text tokens. Prior studies have shown that LLMs can reason directly within the latent space. Building upon this insight, in our Mirage framework, when the model chooses to reason visually by producing a special token, it then reuses its current hidden state as a compact visual embedding and appends it to the context, skipping the language projection. These internal embeddings furnish focused visual cues for later reasoning steps. As shown in Fig. 1, Mirage yields a chain-of-thought trajectory without any external image decoder.

As illustrated in Fig. 3, we adopt a two-stage fine-tuning paradigm to equip the model with interleaved reasoning. In the first stage, with annotated interleaving trajectories, we supervise both modalities: the model predicts the next word while reconstructing a compact latent visual vector obtained from compressed image embeddings. This dual objective

[*]Equal Contribution.

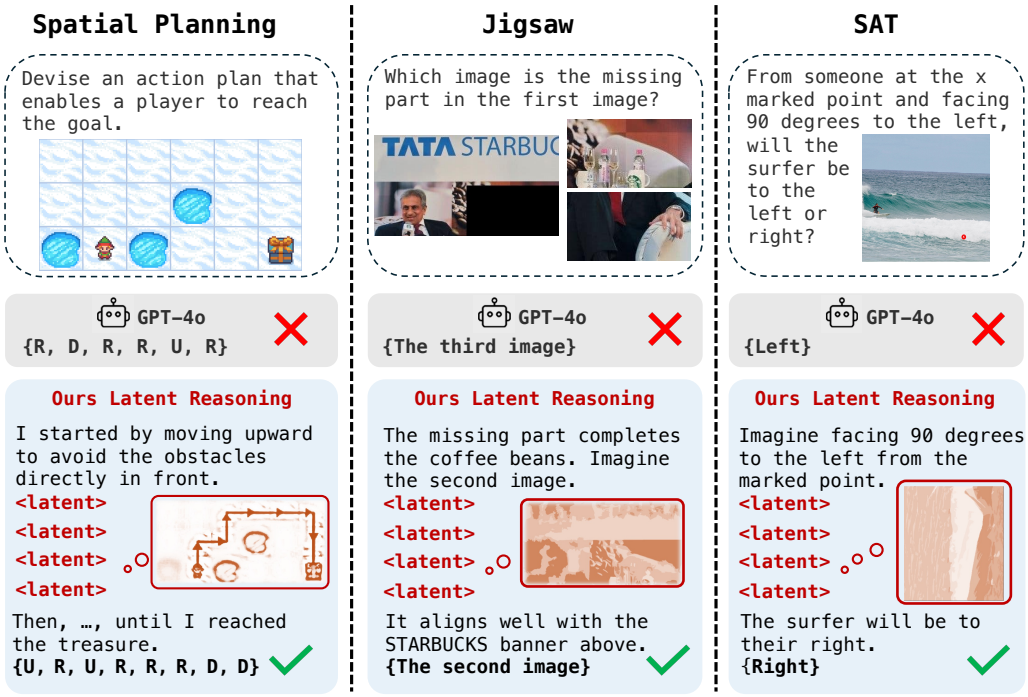

Figure 1. **Multimodal Reasoning Examples.** Mirage interleaves **latent visual tokens**, which represent compact imagery visual features, with explicit text tokens to solve diverse spatial reasoning multimodal tasks, boosting the reasoning performance without the full pixel-level image generation.

anchors the latent tokens in the visual subspace and teaches the model to weave visual cues into its output.

The second stage removes direct supervision on the latent vectors and optimizes only the text tokens, letting the model treat its autoregressively generated latent embeddings as priors that guide subsequent word generation. This relaxation yields a more flexible interleave reasoning trajectory without forcing the latent channel to match any predetermined embedding. After these two stages, we apply reinforcement learning to further boost the reasoning performance.

Extensive experiments and superior performance across multiple benchmarks demonstrate that our proposed Mirage significantly enhances the reasoning ability of VLMs compared with text-only decoding. More concretely, our contributions are threefold,

- We introduce Mirage, which enables VLMs to generate interleaved reasoning trajectories that mix latent visual tokens with ordinary text, without relying on external visual decoders.
- Our two-stage training paradigm empowers VLMs to produce stable yet flexible interleaved reasoning and shows that reinforcement learning can further boost performance.
- Mirage achieves consistent gains across diverse multimodal reasoning benchmarks. Further analysis reveals that the latent tokens embody meaningful visual cues, underscoring the potential to unlock deeper multimodal reasoning capabilities in VLMs.

2. Related Work

2.1. Multimodal Chain-of-Thought

Chain-of-Thought (CoT) prompting was first shown to elicit step-by-step reasoning in LLMs by supplying a few worked examples that include intermediate rationales [16, 50, 64]. Recent extensions of CoT to multimodal settings embed visual evidence directly into the reasoning trajectory. ICoT [67] interleaves attention-selected image crops with text tokens, yielding significant VQA gains, while Visual CoT [32] supplies 438 k bounding-box-grounded rationales to train VLMs that emit explicit visual tokens and improve spatial grounding. Beyond static VQA, several works [61, 63, 68] extend visual thoughts to embodied domains, demonstrating their effectiveness in planning and action execution. In VLM areas, recent works [8, 11, 14, 18, 22, 40, 51, 58, 70] further leverage external tools to supply visual cues that enrich multimodal CoT reasoning..

Recent works [5, 46] like Chameleon [10, 44] trains a unified token-based model that can emit arbitrary sequences of text and image tokens, but at the cost of large-scale pixel-level supervision and heavier decoding. In the VLM domain, multi-modal reasoning is a promising direction [7, 24, 28, 31, 49, 54, 62, 65, 69]. Multimodal-CoT [66]

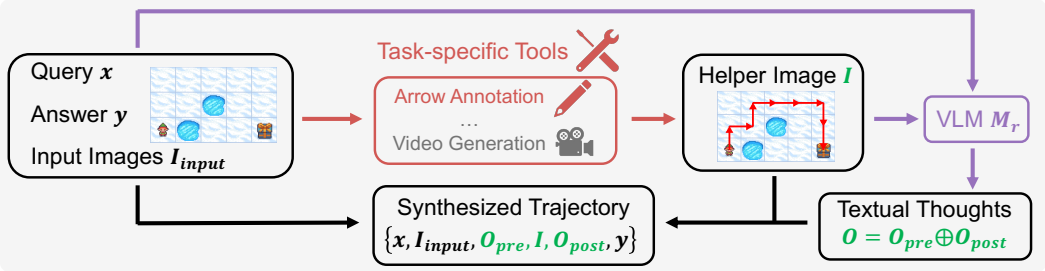

Figure 2. **Data-generation Pipeline.** For each question–answer pair, we first create a helper image with task-specific tools (here, annotate the map with arrows), then prompt a VLM to produce textual reasoning that embeds this image. The text and helper image together form the synthetic multimodal trajectory used for training.

aligns text and image features to generate auxiliary images that enhance reasoning. MVoT [25] further trains a unified model to directly produce image and text interleaving trajectories, but is absent of reasoning thoughts. Aurora [2] introduces an image de-tokenizer to explicitly generate perception tokens, thereby supporting multimodal perception. MMaDA [55] adopts a diffusion-based VLM to enable coherent reasoning and generation across modalities. In contrast, our Mirage framework differs by emitting compact latent visual tokens rather than real image patches or pixels, avoiding heavy image generation while still allowing fully interleaved visual–text reasoning.

2.2. Latent Reasoning in LLMs

Much recent work has highlighted the importance of intermediate hidden representations in Large Language Models (LLMs) [3, 56]. To better guide the latent reasoning process, several approaches introduce specialized tokens into the input sequence. [48] incorporate discrete `<plan>` tokens to control reasoning stages, while [20] propose inserting a `<pause>` token during pretraining to stabilize multi-step reasoning.

Another line of work [19, 30, 35, 38, 39, 41, 42, 71] seeks to internalize reasoning behavior by distilling chain-of-thought rationales into latent representations. [12] trains models to mimic CoT-style reasoning implicitly through hidden states, and [13] further improves inference efficiency by removing explicit intermediate steps altogether. [60] proposes to distill latent reasoning capabilities into a model by supervising it with data generated for complex reasoning. More recently, [21] go further by replacing CoT tokens with continuous latent embeddings, enabling unconstrained reasoning in the latent space to explore on complex tasks, including math and logical reasoning. While prior work primarily focuses on enhancing efficiency or structural planning within the LLM's latent space, our approach takes a different perspective: we treat latent tokens as a *bridge for exploring visual information* into the model.

3. Multimodal Reasoning with Latent Visual Tokens

Inspired by the cognitive process of mental imagery, we introduce Mirage, a framework that lets VLMs reason in interleaved multimodal trajectories. In contrast to prior unified models that integrate an external image decoder and pretrain on large-scale image generation, our method generates compact latent embeddings that serve as visual tokens. By sidestepping image generation, the model can devote its capacity to reasoning, producing only the essential visual cues and thereby echoing the concise, sketch-like representations humans employ during reasoning.

In this section, we first explain how we synthesize informative multimodal reasoning data (Sec. 3.1). Next, we introduce our first joint supervision training stage in Sec. 3.2. Finally, we explain the second stage, which applies text-only supervision while relaxing the latent constraints (Sec. 3.3).

3.1. Data Generation

Consider the multimodal reasoning task where the VLMs need to generate responses y to the input that consists of one or more images and a textual query. For simplicity, we denote the input that contains both image and text as $\boldsymbol{x}$.

Given VLMs naturally generating text tokens only, they require additional supervised fine-tuning to learn an interleaved reasoning pattern. We therefore begin by synthesizing a training corpus that pairs each input $\boldsymbol{x}$ with a task-specific helper image I (See Fig. 2). For example, in the navigation task, the helper image can be generated by taking the ground truth action list and manually drawing the corresponding path on the starting map with red arrows. Similarly, for the jigsaw task, we can concatenate the candidate fragments to form a composite image that captures the relationship among pieces. More details on the image generation procedures for different tasks can be found in the supplementary materials. In general, we obtain a help image that delivers precisely the visual cues needed to supervise latent reasoning.

With the helper image I prepared, we next synthesize

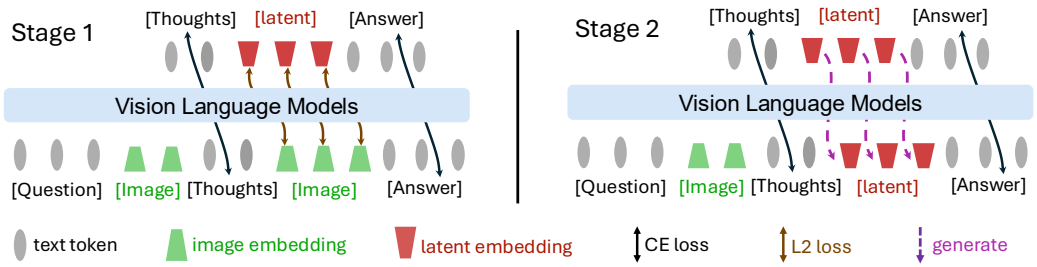

Figure 3. **Pipeline of Mirage Framework.** Stage 1 jointly supervises text and latent visual tokens, grounding the latter in the visual subspace; Stage 2 drops the latent supervision, anchoring the grounded latent tokens for subsequent text generation.

a reasoning chain where the LLM incorporates the helper image to generate the final solution. Specifically, we first feed a large reasoning VLM M with the original input $\boldsymbol{x}$, the ground-truth answer $\boldsymbol{y}$, and the helper image I and prompt it to generate a step-by-step reasoning that incorporates the helper image. For example, the prompt can be `Generate a step-by-step reasoning that leads to the ground-truth answer while properly incorporating the helper image in reasoning`. Denote the model response as

$$\boldsymbol{o} = M\left(\boldsymbol{x}, \boldsymbol{y}, I\right).$$

Here $\boldsymbol{o}$ is a step-by-step reasoning with the helper image embedded in the reasoning process. Since the helper image is embedded in the reasoning chain, it splits the reasoning chain into two parts. Without loss of generality, we represent $\boldsymbol{o} = \boldsymbol{o}_{\text{pre}} \oplus I \oplus \boldsymbol{o}_{\text{post}}$, where $\oplus$ is the concatenation operation, $\boldsymbol{o}_{\text{pre}}$ is the reasoning chain before the helper image while $\boldsymbol{o}_{\text{post}}$ is the reasoning chain after the helper image. By prompt the large reasoning VLM with different inputs, we can thus collect a training dataset $\mathcal{D} = \{\boldsymbol{x}^{(i)}, I^{(i)}, \boldsymbol{o}^{(i)}, \boldsymbol{y}^{(i)}\}_{i=1}^{N}$, where each $\boldsymbol{o}^{(i)}$ is a synthesized reasoning chain with text and image interleaved.

3.2. Joint Supervision for Latent Grounding

To teach the model an interleaved style of reasoning, one naive solution is to directly train a VLM on the data collected above. However, the effectiveness can be negatively affected by the model's limited capability of synthesizing helper images. Therefore, we propose a novel training strategy: pass the helper images to the VLM first to convert the helper images in the synthetic training data into patch-level features; then fine-tune the VLM to output such features as latent reasoning tokens, thus eliminating the need to generate helper images by the VLM.

Specifically, for each training example $(\boldsymbol{x}, I, \boldsymbol{o}, \boldsymbol{y}) \in \mathcal{D}$, we pass the helper image I through the VLM $f_{\boldsymbol{\theta}}(\cdot)$ with parameter $\boldsymbol{\theta}$ to obtain its patch-level features $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\} = f_{\boldsymbol{\theta}}(I)$. Rather than asking the model to reproduce every patch, we mimic human mental imagery by compressing these features into k salient vectors, $\{\hat{\boldsymbol{e}}_1, \ldots, \hat{\boldsymbol{e}}_k\} = \text{Compress}\big(\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}\big)$, that retain only task-critical visual cues. In this work, we realize $\text{Compress}(\cdot)$ with simple average pooling over the original patch features, a lightweight yet surprisingly effective strategy that supplies a concise visual summary for supervision. We then train our model to (1) generate the response $\boldsymbol{o}_{\text{pre}}$ conditioned on the input $\boldsymbol{x}$, (2) generate the latent tokens $\{\hat{\boldsymbol{e}}_1, \ldots, \hat{\boldsymbol{e}}_k\}$ conditioned on $\boldsymbol{x}$ and $\boldsymbol{o}_{\text{pre}}$, where the last layer hidden states at corresponding positions will be regarded as the generated latent tokens, and (3) generate the response $\boldsymbol{o}_{\text{post}}$ conditioned on the proceeding content.

For the training objective for latent token generation, we adopt the cosine similarity between the last layer hidden states of the model and the target latent tokens:

$$\mathcal{L}_{\text{visual}} = \ell_{\cos}\Big(\hat{\boldsymbol{e}}_j,\, g_\theta\big(\boldsymbol{o}_{\text{pre}},\, \hat{\boldsymbol{e}}_{1:j-1}\big)\Big), \tag{1}$$

where $g_\theta\big(\boldsymbol{o}_{\text{pre}},\, \hat{\boldsymbol{e}}_{1:j-1}\big)$ denotes the model's prediction for the j-th latent token conditioned on the preceding context. This loss grounds the latent tokens firmly in the visual representation space.

Meanwhile, we train the surrounding textual tokens using the standard cross-entropy loss for next token prediction. For the left segment $\boldsymbol{o}_{\text{pre}}$ the model conditions only on earlier words, whereas for the right segment $\boldsymbol{o}_{\text{post}}$ it also attends to the k compressed visual embeddings.

$$\begin{aligned}\mathcal{L}_{\text{text}} = &\sum_{i=1}^{|\boldsymbol{o}_{\text{pre}}|} \ell_{\text{CE}}\big(\boldsymbol{o}_{\text{pre},i},\, f_\theta(\boldsymbol{x}, \boldsymbol{o}_{\text{pre},<i})\big) \\ &+ \sum_{i=1}^{|\boldsymbol{o}_{\text{post}}|} \ell_{\text{CE}}\big(\boldsymbol{o}_{\text{post},i},\, f_\theta(\boldsymbol{x}, \boldsymbol{o}_{\text{pre}}, \{\hat{\boldsymbol{e}}_j\}_1^k, \boldsymbol{o}_{\text{post},<i}\big).\end{aligned} \tag{2}$$

Here $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ denotes the next token prediction probability conditioned on the input and $\{\hat{\boldsymbol{e}}_j\}_1^k$ is the set of the ground-truth latent tokens. The overall training objective in this stage combines this term with the visual-alignment loss $\mathcal{L}_1 = \mathcal{L}_{\text{visual}} + \gamma\mathcal{L}_{\text{text}}$, where the γ is the loss coefficient, thereby anchoring the latent tokens in visual space while teaching the model to weave them naturally into its textual thoughts.

VSP	Spatial Reasoning					Spatial Planning				
	Level 3	Level 4	Level 5	Level 6	Avg.	Level 3	Level 4	Level 5	Level 6	Avg.
Zero-Shot	0.32	0.23	0.40	0.32	0.32	0.10	0.08	0.05	0.01	0.06
Direct SFT	0.83	0.81	0.85	0.86	0.83	0.88	0.81	0.73	0.47	0.72
CoT SFT	0.88	0.86	0.80	0.83	0.84	0.68	0.53	0.35	0.31	0.47
GRPO	0.54	0.49	0.76	0.67	0.62	0.42	0.35	0.26	0.08	0.28
CoT SFT + GRPO	0.89	0.85	0.84	0.80	0.85	0.65	0.58	0.43	0.38	0.51
Anole	0.46	0.51	0.49	0.63	0.52	0.02	0.01	0.00	0.00	0.01
MVoT	0.53	0.64	0.67	0.59	0.61	0.21	0.11	0.08	0.03	0.11
Aurora	0.64	0.78	0.71	0.71	0.71	0.26	0.11	0.11	0.04	0.13
Ours (Direct)	0.86	0.84	**0.88**	0.87	0.86	**0.93**	**0.83**	**0.76**	**0.51**	**0.76**
Ours (CoT)	0.87	**0.92**	0.86	0.84	0.87	0.75	0.63	0.53	0.39	0.58
+ w/ GRPO	**0.92**	0.90	0.86	**0.88**	**0.89**	0.78	0.65	0.52	0.43	0.60

Table 1. **Results on Visual-Spatial Planning (VSP) tasks.** Mirage outperforms text-only baselines and achieves superior performance compared to interleave reasoning models.

3.3. Text-Only Supervision with Latent Relaxation

The first stage grounds the latent tokens by forcing the model to reconstruct the compressed image embeddings. Although effective for visual alignment, this can over-constrain the model, diverting capacity from its primary goal of answering the question correctly, degrading the reasoning performance. Therefore, in the second stage, we remove the cosine loss altogether and keep only the cross-entropy loss over text tokens.

Although the latent tokens no longer carry an explicit loss, we still anchor them so that they meaningfully guide the following thoughts. For each training instance, the model autoregressively produces its own latent tokens $\{\boldsymbol{e}_i\}_{i=1}^{k}$, with

$$\boldsymbol{e}_j = f_{\boldsymbol{\theta}}\big(\boldsymbol{x}, \boldsymbol{o}_{\text{pre}}, \boldsymbol{e}_{<j}\big). \tag{3}$$

These self-generated embeddings replace the compressed image vectors used in Stage 1 and serve as priors for the tokens that follow the image placeholder. Therefore, the training objective becomes

$$\begin{aligned}\mathcal{L}_{\text{text}} = &\sum_{i=1}^{|\boldsymbol{o}_{\text{pre}}|} \ell_{\text{CE}}\big(\boldsymbol{o}_{\text{pre},i},\ f_\theta(\boldsymbol{x}, \boldsymbol{o}_{\text{pre},<i})\big) \\ &+ \sum_{i=1}^{|\boldsymbol{o}_{\text{post}}|} \ell_{\text{CE}}\big(\boldsymbol{o}_{\text{post},i},\ f_\theta(\boldsymbol{x}, \boldsymbol{o}_{\text{pre}}, \{\boldsymbol{e}_j\}_1^k, \boldsymbol{o}_{\text{post},<i}\big).\end{aligned} \tag{4}$$

Due to the continuous property of $\{\boldsymbol{e}_i\}_{i=1}^{k}$, these self-generated latent tokens are fully differentiable. Since the next token prediction of $\boldsymbol{o}_{\text{post}}$ is a function of the latent tokens, the gradient can be propagated to these latent tokens when minimizing the above loss on the textual tokens. This allows us to optimize the generation of the latent tokens within the learned visual subspace, acting as flexible priors that guide subsequent text generation and yield a more adaptive, task-focused reasoning.

The overall framework of our two-stage pipeline is provided in Fig. 3. These two stages jointly endow VLMs with the ability to generate interleaved multimodal reasoning with latent visual tokens. Empirical results in Sec. 4.2 further validate the effectiveness of our latent reasoning over naive text-only decoding.

3.4. Reinforcement Learning

After the two supervised fine-tuning stages, the model has already learned to reason using both interleaved text and latent tokens. Here, we further explore whether the model's performance can be improved using reinforcement learning (RL), inspired by recent long-CoT language models [34, 53]. Specifically, we adopt group relative policy optimization (GRPO) [33] for RL training. For each input query in the training set, we sample multiple responses from the model. During RL, we explicitly optimize the probabilities of textual tokens while allowing gradients to flow through the latent tokens. Following LMM-R1 [27], we adopt two types of rewards: accuracy and format. We consider both accuracy and format rewards. For accuracy reward, we set $r_{\text{acc}}(\boldsymbol{o}, \boldsymbol{x}) = 1$ if the final answer is correct, and 0 otherwise. For the format reward, we check whether the thinking process is enclosed between "<think>" and "</think>" tags and whether the final answer format is formatted as "`\boxed{}`" in the output response $\boldsymbol{o}$. If the format is correct, the reward is 0.1; otherwise, it is 0.

4. Experiments

4.1. Experimental Settings

Benchmarks. We evaluate our approach on four different benchmarks. **VSP** [52] measures spatial planning in a simulated maze-navigation environment. In addition to its main task, we adopt its spatial reasoning subtask, which asks the model to predict the outcome of a prescribed action sequence. **BLINK-Jigsaw** [17] systematically evaluates the capacity of multimodal large language models to extrapolate global structural and semantic information from incomplete visual

inputs, thereby assessing their reasoning about spatial organization and maintaining perceptual coherence. **SAT** [29] evaluates both static and dynamic spatial relations. Additionally, we include the Mathematical Geometry subset of the recent **COMT-Geometry** [9] to assess formal spatial reasoning in mathematical contexts. Details are provided in Appx B.

Data Synthesis. For each task, we sample 1k training instances for fine-tuning and 1k instances for reinforcement learning. COMT provides interleaved multimodal reasoning trajectories, which we directly use as both helper images and reasoning supervision. For the other benchmarks, we synthesize data following the procedure outlined in Sec. 3.1. For VSP, the helper image is either the start map annotated with the red-arrow path or the agent's current state snapshot. In Jigsaw, we concatenate one candidate patch beside the reference image. For SAT, following MindJourney [57], we prompt fine-tuned `CogVideoX-5B` [59] to render a scene that matches the textual description. Full synthesis details are provided in Appx B.

Baseline Models. First, we fine-tune the model directly with answer labels and evaluate zero-shot reinforcement learning without any supervised warm-up. Next, using synthetic data, we perform CoT fine-tuning (CoT SFT) and then add reinforcement learning. In addition, we benchmark against a unified model **Anole** [10] and a reasoning model **MVoT** [25], both capable of generating textual actions and state images, with same multimodal supervision used for our method. We also compare against reasoning models that interleave either generated or processed images into reasoning trajectories, including Aurora [2] with implicit visual output, ViGoRL [31] and MindJourney [57] designed for spatial reasoning tasks, and MINT-CoT [6] specifically fine-tuned model for math geometry problems. For more baseline details, please refer to Appx C.2. To ensure fair comparison, we report baselin results with the best performance across zero-shot and fine-tuning setting.

Implementation Details. Unless stated otherwise, we use `Qwen2.5-VL 7B` as the base model, and we use a latent token size of $k = 4$ and a loss coefficient of $\gamma = 0.1$. The random seed is fixed at 42 to ensure reproducibility. For more implementation details, please refer to Appx C.1.

4.2. Experimental Results

We first evaluate the effectiveness of our method on the VSP benchmark. The results are shown in Tab. 1. First, adding latent visual tokens to the reasoning process significantly improves the reasoning capability of VLMs compared to text-only baselines. Compared to directly fine-tuning the VLM with the synthesized data, our method achieves 3% higher accuracy on the spatial reasoning task and 11% on the spatial planning task. Also, Mirage improves the CoT SFT + GRPO, by 2% and 7%, respectively, demonstrating the effectiveness of our two-stage training method. Moreover, reinforcement learning can further improve the performance of our method. By weaving latent visual tokens within the text trajectories, our model can naturally explore diverse sequences. After optimizing with GRPO, Mirage achieves extra gains (+2% accuracy) on VSP tasks. These results further confirm that interleaved latent cues provide informative guidance with flexible reasoning, highlighting the potential of our framework.

Method	Jigsaw	SAT Synthetic			SAT Real
		GoalAim	ObjM	Avg.	
Zero-Shot	0.45	0.50	0.38	0.44	0.51
Direct SFT	0.80	0.82	0.83	0.83	0.55
CoT SFT	0.59	0.73	0.88	0.71	0.54
GRPO	0.54	0.78	0.80	0.79	0.54
SFT + GRPO	0.72	0.82	0.85	0.84	0.52
ViGoRL	0.56	0.75	0.58	0.67	0.59
MindJourney	-	0.84	0.62	0.73	0.57
Ours	**0.85**	**0.85**	**0.93**	**0.89**	**0.64**

Table 2. **Experimental Results with Qwen2.5-VL 3B on Jigsaw and SAT tasks.** For each baseline, we report the best performance across zero-shot and fine-tuning settings, with and without CoT. Baseline implementations are described in Sec. C.2, and detailed results are provided in Sec. D.1.

Additionally, baselines such as Aurora and Anole, despite explicitly generating image tokens, perform poorly on interleaved text–image reasoning. After fine-tuning on the same data, they reach only 71% accuracy on the spatial reasoning task and 13% on the spatial planning task. We attribute this to the overhead of explicit image generation. Our reproduced MVoT [25] results are lower than those reported in their paper, likely due to training data differences. To ensure a fair comparison, we reproduce MVoT results using same data as used in our framework. Nonetheless, our model still outperforms their reported results, highlighting the advantage of our latent design.

We also evaluate our model on COMT tasks in Tab. 3. Mirage sft model achieves about 5% accuracy than best baselines. Furthermore, we report results on Jigsaw and SAT tasks in Tab. 2 using the `Qwen2.5-VL 3B`, showing that Mirage transfers effectively to smaller models, with performance improvements consistent with 7B models. Across both benchmarks,

Method	COMT
Zero-Shot	0.63
Direct SFT	0.71
CoT SFT	0.74
GRPO	0.69
SFT + GRPO	0.72
R1-VL	0.69
MINT-CoT	0.72
Ours	**0.77**

Table 3. **Results on COMT.**

Method	VSP Spatial Planning				
	3	4	5	6	Avg.
Ours	0.75	0.63	0.53	0.39	0.58
– w/o Stage 1	0.69	0.58	0.46	0.36	0.52
– w/o Stage 2	0.38	0.19	0.16	0.09	0.21

Table 4. **Ablation study of training stages on VSP Spatial Planning.** Both stages jointly improve reasoning performance.

our sft model outperforms all relevant text and image interleaved reasoning baselines, including MINT-CoT, which is explicitly trained on large-scale math data, and ViGoRL, which is trained on large-scale spatial reasoning data. Consistent improvements underscore that interleaving compact visual cues consistently strengthens reasoning ability. More detailed results are provided in Appx D.1.

We notice that on VSP spatial planning task, fine-tuning with synthesized reasoning thoughts performs significantly worse than training directly on answer labels, both with and without our latent design. Two factors likely contribute to this outcome. First, as noted in prior work [26], certain visual tasks that rely heavily on perception may not benefit from explicit reasoning during fine-tuning. Second, the synthesized thoughts are generated by `Qwen2.5-VL-32B`; although generally sound, they are not flawless, and any imperfections propagate into the base model. Likely, in SAT, the helper images are produced by a video generation model without ground-truth annotations, which can introduce further noise to the latent prior. Despite these challenges, our latent reasoning pipeline still closes much of the performance gap, highlighting its practical robustness.

4.3. Ablation Study

Effectiveness of Two Stage Design Tab. 4 reports the effect of removing each phase. Training with only the first phase, which jointly supervises text and latent tokens, anchors the latent embeddings but leaves them constrained and lowers performance, similar to the plight of unified models.

Training with only the second stage, which relies on text loss alone while letting latent tokens evolve freely, performs slightly better than text-only baseline. Without the grounding supplied by the first stage, the latent vectors drift into regions of the multimodal embedding space that do not aid reasoning. This outcome contrasts with findings on LLMs [21], where unsupervised latent vectors can benefit subsequent reasoning. The difference indicates that visual and textual subspaces in VLMs remain heterogeneous enough that a grounding phase is effective. This confirms that the first stage aligns latent tokens with visual features, the second stage allows them to adapt, and both steps are necessary.

Visual Latent Size and Hyper-parameters To delve deeper into the robustness of our framework, we investigate the influence of hyperparameters: latent token size k and the multimodal loss coefficient γ. As Tab. 5 shows, adjusting the loss coefficient γ has a moderate effect. A larger γ weights the latent-alignment loss less in the first stage. When γ approaches infinity, the first stage becomes equivalent to skipping visual supervision entirely, in other words, the second stage. This gives a poor initialization for subsequent training. Even so, after the second stage, each γ tested still obtains over 80% accuracy, which attests to the overall robustness of the framework. Additionally, we observe that varying the latent size k from 2 to 6 yields consistently strong performance, with $k = 6$ showing a slight improvement—highlighting the resilience of our latent design. However, increasing k to 8 results in a significant performance drop around 13%, likely due to error accumulation in longer latent sequences under autoregressive non-decoding generation. These observations are consistent with prior findings that optimal latent reasoning performance in LLMs typically occurs with fewer than 6 latent tokens [21].

k	γ	VSP Spatial Reasoning				
		3	4	5	6	Avg.
2	0.1	0.85	0.86	0.89	**0.93**	0.86
4	0.1	**0.87**	**0.92**	0.86	0.84	0.87
6	0.1	0.85	0.90	**0.91**	0.87	**0.88**
8	0.1	0.77	0.77	0.74	0.70	0.75
4	0.5	0.84	0.91	0.84	0.78	0.84
4	1.0	0.77	0.85	0.85	0.87	0.83

Table 5. **Ablation study of latent size k and loss coefficient γ.** The pipeline remains robust across hyperparameters.

5. Analysis

Synthesized Data Quality. Data quality plays a critical role in model performance. In this section, we investigate whether the helper images generated by various tools are genuinely informative for VLM reasoning. For the two VSP tasks, we supply the helper image as prior input and evaluate model performance in both zero-shot and fine-tuned settings. As shown in Fig. 4, providing the helper image leads both models to achieve nearly 100% accuracy on both tasks. Even in the zero-shot setting, we observe substantial performance gains on the spatial reasoning task. However, improvements on the spatial planning task are limited to simpler map layouts in the zero-shot setting. We attribute this to the inherent difficulty of extracting and leveraging spatial information from the helper image without task-specific fine-tuning. These results suggest that the synthesized helper images do indeed enhance VLM reasoning. Moreover, if the model's latent thoughts can fully internalize the information encoded in these images, it would represent a strong performance upper bound for our Mirage.

Latent Behavior Analysis. The model learns to reproduce compressed image embeddings in the first stage, anchoring

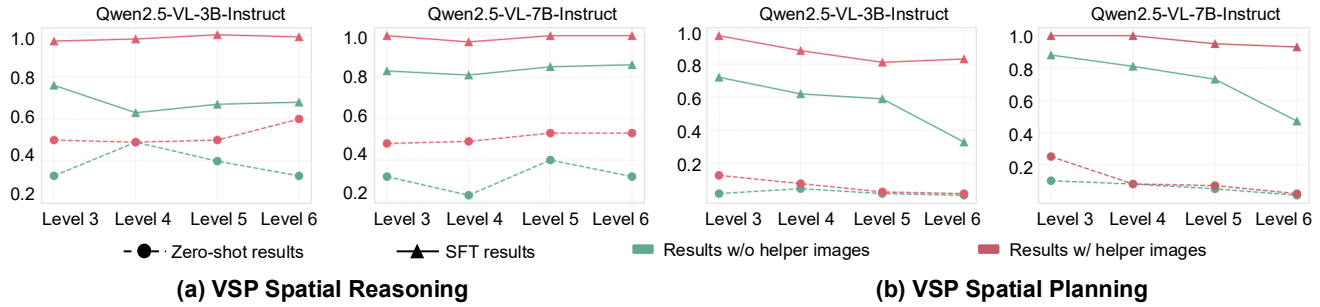

Figure 4. **Performance with Helper Images as Input Priors.** We evaluate model accuracy using synthesized helper images under both zero-shot and fine-tuned settings. The results highlight the informativeness of the generated images and confirm their high data quality.

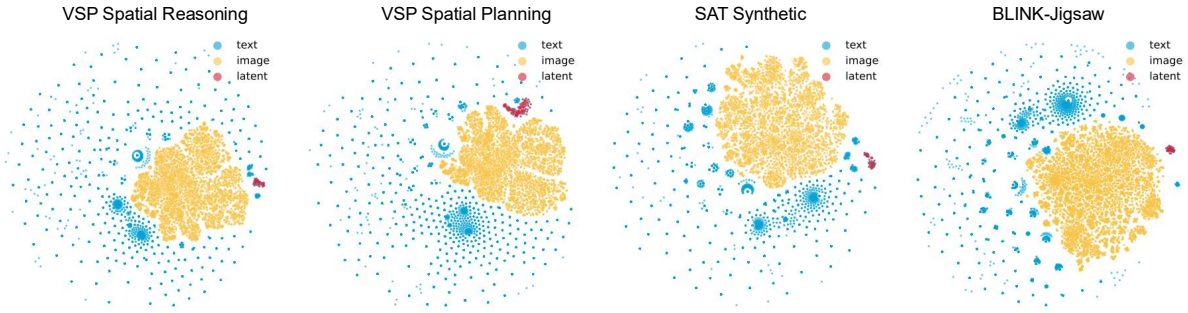

Figure 5. **Visualization of Latent Embeddings.** We visualize our latent tokens along with text and image embeddings with t-SNE. Latent tokens cluster near, yet just outside, the visual representation subspace, consistent with the two-stage training design.

its latent tokens in the visual subspace. However, after the second stage, these latent tokens receive no direct supervision. Therefore, it is unclear whether they still encode visual representations. By sampling 100 examples, we obtain the corresponding latent token vectors alongside the text and visual embeddings. We use t-SNE to embed all vectors into two dimensions for better visualization with a perplexity of 30, and initialize the embeddings via PCA. As shown in Fig. 5, the text embeddings (blue dots) fill the entire plot in a radial scattering pattern, while the visual token embeddings (yellow dots) cluster tightly inside a distinct visual subspace, consistent with previous findings. Our latent embeddings (red dots) form a compact cloud that sits just outside that visual cluster, shifted by the second training stage, which tailors the latent embeddings to answer generation. However, we notice that our latent tokens remain clearly separated from the text distribution and closer to the visual embedding across diverse tasks. This pattern shows that even without an explicit decoder, the latent tokens stay close to the visual manifold while retaining the flexibility introduced in the second stage, echoing the way mental imagery abstracts rather than reproduces visual input.

Multiple Latent Reasoning Steps. Beyond using a single helper image, we investigate the case where multiple steps of latent tokens appear in the reasoning trajectory. Interestingly, models trained with only one helper image naturally produce multiple latent visual reasoning steps occasionally during inference on structured, multi-step tasks such as VSP, reflecting the flexibility of our framework. We further extend experiments by explicitly using two helper images. Results in Appx. D.2 demonstrate that 3B models achieve an average 7% improvement on the VSP spatial reasoning task, suggesting that additional visual support can further benefit.

6. Conclusion

In this work, mimicking human mental imagery, we propose Mirage, a lightweight framework that interleaves compact latent visual tokens with text so a vision–language model can reason multimodally without ever generating pixel-level images. Specifically, our framework is trained in two stages: a joint supervision stage that anchors latent tokens to visual embeddings while learning the surrounding text, followed by a text-only supervision stage that lets those tokens adapt freely to support answer generation. A brief reinforcement-learning refinement further aligns the entire trajectory with task goals. Across four spatial-reasoning benchmarks, Mirage consistently outperforms text-only baselines, underscoring the effectiveness and potential of latent visual reasoning for multimodal models.

Acknowledgments. The authors gratefully acknowledge the support of the MIT-IBM Watson AI Lab. We also would like to thank Haoyu Zhen, Bairu Hou, Guangtao Zeng, Yuncong Yang, Jiaben Chen, Ziwei Liu, Zonghan Yang, Sunli Chen, Lixing Fang and many other frieds for their helpful feedback and insightful discussions.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 14

[2] Mahtab Bigverdi, Zelun Luo, Cheng-Yu Hsieh, Ethan Shen, Dongping Chen, Linda G Shapiro, and Ranjay Krishna. Perception tokens enhance visual reasoning in multimodal language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3836–3845, 2025. 3, 6

[3] Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. Hopping too late: Exploring the limitations of large language models on multi-hop queries. *arXiv preprint arXiv:2406.12775*, 2024. 3

[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016. 12

[5] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025. 1, 2

[6] Xinyan Chen, Renrui Zhang, Dongzhi Jiang, Aojun Zhou, Shilin Yan, Weifeng Lin, and Hongsheng Li. Mint-cot: Enabling interleaved visual tokens in mathematical chain-of-thought reasoning. *arXiv preprint arXiv:2506.05331*, 2025. 6

[7] Zhangquan Chen, Xufang Luo, and Dongsheng Li. Visrl: Intention-driven visual perception via reinforced reasoning. *arXiv preprint arXiv:2503.07523*, 2025. 2

[8] Zihui Cheng, Qiguang Chen, Xiao Xu, Jiaqi Wang, Weiyun Wang, Hao Fei, Yidong Wang, Alex Jinpeng Wang, Zhi Chen, Wanxiang Che, et al. Visual thoughts: A unified perspective of understanding multimodal chain-of-thought. *arXiv preprint arXiv:2505.15510*, 2025. 2

[9] Zihui Cheng, Qiguang Chen, Jin Zhang, Hao Fei, Xiaocheng Feng, Wanxiang Che, Min Li, and Libo Qin. Comt: A novel benchmark for chain of multi-modal thought on large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23678–23686, 2025. 6

[10] Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*, 2024. 1, 2, 6

[11] Ethan Chern, Zhulin Hu, Steffi Chern, Siqi Kou, Jiadi Su, Yan Ma, Zhijie Deng, and Pengfei Liu. Thinking with generated images. *arXiv preprint arXiv:2505.22525*, 2025. 2

[12] Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023. 3

[13] Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*, 2024. 3

[14] Irving Fang, Juexiao Zhang, Shengbang Tong, and Chen Feng. From intention to execution: Probing the generalization boundaries of vision-language-action models. *arXiv preprint arXiv:2506.09930*, 2025. 2

[15] Martha J Farah. Psychophysical evidence for a shared representational medium for mental images and percepts. *Journal of Experimental Psychology: General*, 114(1):91, 1985. 1

[16] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36:70757–70798, 2023. 2

[17] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024. 5

[18] Timin Gao, Peixian Chen, Mengdan Zhang, Chaoyou Fu, Yunhang Shen, Yan Zhang, Shengchuan Zhang, Xiawu Zheng, Xing Sun, Liujuan Cao, et al. Cantor: Inspiring multimodal chain-of-thought of mllm. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 9096–9105, 2024. 2

[19] Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025. 3

[20] Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*, 2023. 3

[21] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024. 3, 7

[22] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *arXiv preprint arXiv:2406.09403*, 2024. 2

[23] Stephen M Kosslyn. *Image and brain: The resolution of the imagery debate*. MIT press, 1996. 1

[24] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025. 2

[25] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*, 2025. 3, 6, 15

[26] Ming Li, Jike Zhong, Shitian Zhao, Yuxiang Lai, and Kaipeng Zhang. Think or not think: A study of explicit thinking in

rule-based visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.16188*, 2025. 7

[27] Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *arXiv preprint arXiv:2503.07536*, 2025. 5

[28] Tan-Hanh Pham and Chris Ngo. Multimodal chain of continuous thought for latent-space reasoning in vision-language models. *arXiv preprint arXiv:2508.12587*, 2025. 2

[29] Arijit Ray, Jiafei Duan, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, Kuo-Hao Zeng, et al. Sat: Spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*, 2024. 6

[30] Yangjun Ruan, Neil Band, Chris J Maddison, and Tatsunori Hashimoto. Reasoning to learn from latent thoughts. *arXiv preprint arXiv:2503.18866*, 2025. 3

[31] Gabriel Sarch, Snigdha Saha, Naitik Khandelwal, Ayush Jain, Michael J Tarr, Aviral Kumar, and Katerina Fragkiadaki. Grounded reinforcement learning for visual reasoning. *arXiv preprint arXiv:2505.23678*, 2025. 2, 6

[32] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning, 2024. 2

[33] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 5, 14

[34] Maohao Shen, Guangtao Zeng, Zhenting Qi, Zhang-Wei Hong, Zhenfang Chen, Wei Lu, Gregory Wornell, Subhro Das, David Cox, and Chuang Gan. Satori: Reinforcement learning with chain-of-action-thought enhances llm reasoning via autoregressive search. *arXiv preprint arXiv:2502.02508*, 2025. 5

[35] Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. Codi: Compressing chain-of-thought into continuous space via self-distillation. *arXiv preprint arXiv:2502.21074*, 2025. 3

[36] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024. 14

[37] Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. 1

[38] Teng Shi, Weicong Qin, Weijie Yu, Xiao Zhang, Ming He, Jianping Fan, and Jun Xu. Bridging search and recommendation through latent cross reasoning. *arXiv preprint arXiv:2508.04152*, 2025. 3

[39] DiJia Su, Hanlin Zhu, Yingchen Xu, Jiantao Jiao, Yuandong Tian, and Qinqing Zheng. Token assorted: Mixing latent and text tokens for improved language model reasoning. *arXiv preprint arXiv:2502.03275*, 2025. 3

[40] Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, et al. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025. 2

[41] Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Jian Luan, and Ruihua Song. Think silently, think fast: Dynamic latent compression of llm reasoning chains. *arXiv preprint arXiv:2505.16552*, 2025. 3

[42] Jiakai Tang, Sunhao Dai, Teng Shi, Jun Xu, Xu Chen, Wen Chen, Jian Wu, and Yuning Jiang. Think before recommend: Unleashing the latent reasoning power for sequential recommendation. *arXiv preprint arXiv:2503.22675*, 2025. 3

[43] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 1

[44] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2025. 2

[45] Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. *arXiv preprint arXiv:2412.14164*, 2024. 1

[46] Dianyi Wang, Wei Song, Yikun Wang, Siyuan Wang, Kaicheng Yu, Zhongyu Wei, and Jiaqi Wang. Autoregressive semantic visual reconstruction helps vlms understand better. *arXiv preprint arXiv:2506.09040*, 2025. 1, 2

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

[48] Xinyi Wang, Lucas Caccia, Oleksiy Ostapenko, Xingdi Yuan, William Yang Wang, and Alessandro Sordoni. Guiding language model reasoning with planning tokens. *arXiv preprint arXiv:2310.05707*, 2023. 3

[49] Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*, 2025. 2

[50] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. 2

[51] Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. *arXiv preprint arXiv:2506.09965*, 2025. 2

[52] Qiucheng Wu, Handong Zhao, Michael Saxon, Trung Bui, William Yang Wang, Yang Zhang, and Shiyu Chang. Vsp: Assessing the dual challenges of perception and reasoning in spatial planning tasks for vlms. *arXiv preprint arXiv:2407.01863*, 2024. 5

[53] Zhihui Xie, Liyu Chen, Weichao Mao, Jingjing Xu, Lingpeng Kong, et al. Teaching language models to critique via reinforcement learning. *arXiv preprint arXiv:2502.03492*, 2025. 5

[54] Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. Visual planning: Let's think only with images. *arXiv preprint arXiv:2505.11409*, 2025. 2

[55] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025. 3

[56] Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? *arXiv preprint arXiv:2402.16837*, 2024. 3

[57] Yuncong Yang, Jiageng Liu, Zheyuan Zhang, Siyuan Zhou, Reuben Tan, Jianwei Yang, Yilun Du, and Chuang Gan. Mindjourney: Test-time scaling with world models for spatial reasoning. *arXiv preprint arXiv:2507.12508*, 2025. 6

[58] Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3d-mem: 3d scene memory for embodied exploration and reasoning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17294–17303, 2025. 2

[59] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 6

[60] Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*, 2024. 3

[61] Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, and Xing Wei. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *arXiv preprint arXiv:2505.17685*, 2025. 2

[62] Yufei Zhan, Ziheng Wu, Yousong Zhu, Rongkun Xue, Ruipu Luo, Zhenghao Chen, Can Zhang, Yifan Li, Zhentao He, Zheming Yang, et al. Gthinker: Towards general multimodal reasoning via cue-guided rethinking. *arXiv preprint arXiv:2506.01078*, 2025. 2

[63] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025. 2

[64] Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Advances in Neural Information Processing Systems*, 37:333–356, 2024. 2

[65] Yizhen Zhang, Yang Ding, Shuoshuo Zhang, Xinchen Zhang, Haoling Li, Zhong-zhi Li, Peijie Wang, Jie Wu, Lei Ji, Yelong Shen, et al. Perl: Permutation-enhanced reinforcement learning for interleaved vision-language reasoning. *arXiv preprint arXiv:2506.14907*, 2025. 2

[66] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023. 2

[67] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models, 2024. 2

[68] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025. 2

[69] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025. 2

[70] Qiji Zhou, Ruochen Zhou, Zike Hu, Panzhong Lu, Siyang Gao, and Yue Zhang. Image-of-thought prompting for visual reasoning refinement in multimodal large language models. *arXiv preprint arXiv:2405.13872*, 2024. 2

[71] Rui-Jie Zhu, Tianhao Peng, Tianhao Cheng, Xingwei Qu, Jinfa Huang, Dawei Zhu, Hao Wang, Kaiwen Xue, Xuanliang Zhang, Yong Shan, et al. A survey on latent reasoning. *arXiv preprint arXiv:2507.06203*, 2025. 3

A. Limitations and Future Works.

While effective, our framework has certain limitations: *Synthetic Data Quality*: The performance of Mirage depends on the quality of the generated multimodal trajectories. High-quality datasets for unified reasoning models are an important next step. *Extend to Unified Models*: Despite current limitations in interleaved generation performance, whether the aligned feature space of unified models can be leveraged to further improve latent reasoning design remains an open question. *Extend to Larger Models*: Currently, our evaluation is limited to `Qwen2.5-VL 7B` and `Qwen2.5-VL 3B` models. Extending Mirage to larger models remains an open direction. *Latent Visualization*: Currently, the generated latent embeddings can only be visualized using t-SNE, lacking a semantic visualization. Developing methods to visualize these embeddings semantically would further enhance the explainability of Mirage.

B. Datasets

B.1. Help Image Generation

Diverse task-specific tools are employed to generate the helper images used in fine-tuning. In this section, we will detail the generation pipeline for each task.

VSP Spatial Reasoning. To assist in inferring the final state after a sequence of actions, we leverage the map layout visualization as the helper image, including the agent position after part of the action trajectory. Following the VSP implementation, we render this state with the OpenAI Gym package [4], using the initial map and the action sequence as inputs.

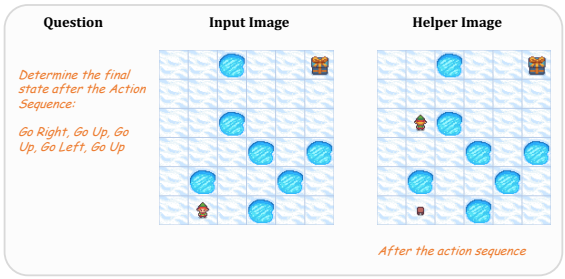

Figure 6. An example of the helper image of the VSP Spatial Reasoning task.

VSP Spatial Planning. For the planning task, we provide a map annotated with the ground-truth path, turning the problem into simply reading the highlighted trajectory. Specifically, we select one valid action sequence for each sample and highlight its steps as a red arrow that begins at the agent's start position and ends at the goal.

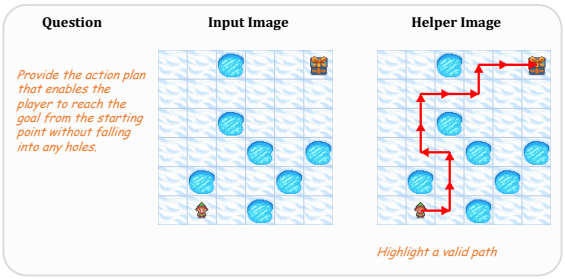

Figure 7. An example of the helper image of the VSP Spatial Planning task.

Blink Jigsaw. The Jigsaw task asks which candidate patch completes the reference image. For each instance we create a helper image by inserting one randomly chosen candidate patch into the masked region. The model then can judge whether the composite looks seamless: if the patch blends smoothly, it is the correct answer; if not, the other candidate should be chosen.

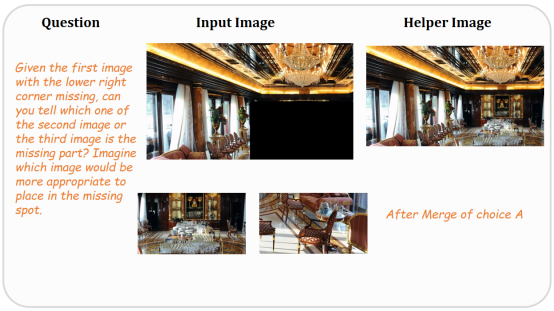

Figure 8. An example of the helper image of the BLINK task.

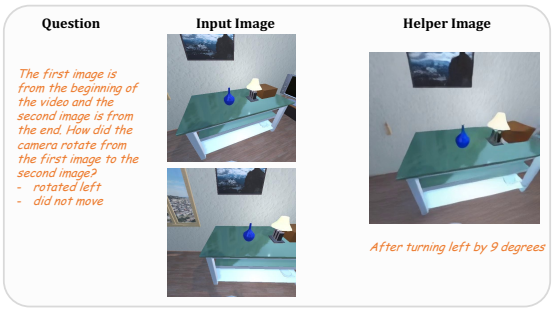

Figure 9. An example of the helper image of the SAT task.

SAT. For the SAT task, we focus on the GoalAim and ObjM subtasks, which require reasoning about a specified camera pose movement. Providing the target view as a helper image would ease the model's spatial reasoning burden. Therefore, given the recent advance in world model

Table 6. **Data example of VSP Spatial Reasoning.**

Thoughts Generation System Prompt:

You are analysing *one move* in FrozenLake.
Tiles (numeric): 1=Start point, 0=Ice, -1=Hole, 2=Target.
End-states: success, fail in hole, fail on ice.

Task: Given the map, current position, and one action, write *one short sentence thought* that states: (1) the square reached $[(row, col), \text{TileType}]$; (2) how that move shifts the likelihood of the three end-states.
No bullets, no extra lines, just the single sentence.

Example 1:
Numeric map: `[[1,0,0],[0,-1,0],[0,0,2]]`
At (0,0) action RIGHT
Thought: I slide onto $[(0, 1), \text{Ice}]$ staying safe and inching closer to the target, so success is a bit likelier while hole-failure remains unchanged.
Example 2:
Numeric map: `[[1,0,-1],[0,-1,0],[0,0,2]]`
At (1,0) action RIGHT
Thought: Moving to $[(1, 1), \text{Hole}]$ ends the game instantly in a hole, making "fail in hole" certain and wiping out other outcomes.

Example Input:
Task: FrozenLake. Given the initial state and action sequence, visualize the final state of the frozen lake map and determine whether the agent can safely reach the gift.
Full Action Sequence: Go Left, Go Down.
The reasoning process must be enclosed within `<think>...</think>` tags. The final answer must be put in `\boxed{}`.

Example Output:
`<think>` Moving to $[(1, 3), \text{Hole}]$ ends the game instantly... `</think>`
The answer is `\boxed{A}`.

research, we adopt a high-quality video generation model **CogVideoX-5B** to generate this image. To further ensure the image quality, we restrict the action condition for generation to three primitives: move forward, turn left, and turn right. Sampling 9 frames along each trajectory, we instruct a VLM to choose the most informative frame. The chosen frame is then used as the helper image.

B.2. Textual Thoughts Generation

For each task, we generate the textual thoughts instead of leveraging closed-source outputs. We feed the helper image and the ground truth answer to a large reasoning model `Qwen2.5-VL 32B`. Task-specific prompts are applied. Simplified prompts and one illustrative example per task are provided in Tab. 6–9. To encourage diversity in model outputs, three distinct reasoning trajectories are generated per helper image.

The generated thoughts and the associated helper image serve as the supervision for fine-tuning, and the quality of these explanations sets an upper bound on our model's performance. Our current approach relies on straightforward prompts, which occasionally yield subpar reasoning trajectories. Developing richer prompts or otherwise curating higher-quality trajectories remains an important future work.

B.3. Data Configuration

For the Mathematical Geometry subset of COMT, we randomly sample 200 examples for evaluation and use the remaining 820 for both fine-tuning and reinforcement learning.

Table 7. **Data example of VSP Spatial Planning.**

VSP Spatial Planning

Thoughts Generation System Prompt:
You are a concise maze navigator.
Input: (1) an image of a maze, (2) a second image, and (3) a proposed action sequence.
Task: Write 1–2 English sentences that logically justify why the given action sequence reaches the treasure.
– Pretend you derived the sequence yourself; *never* say it was supplied.
– *Never* mention or allude to the second image or any "highlighted route".
– Do not output anything else.

Example Input:
As a professional maze solver, your task is to analyze a grid-based map and devise an action plan that enables a player to reach the goal from the starting point without falling into any holes, using the fewest possible moves.
Game Setup
– The game presents a fully observable grid-based map.
– The player starts at a specified grid square, with the goal located elsewhere on the map.
– Each grid square is either safe or contains a hole.
– Your goal is to guide the player to the goal while avoiding holes.
– For example, your final answer should be like "L,L,R,U,D", which means an action plan of left, left, right, up, and down.
Moving Rules
– The action plan involves a series of moves: 'L' (left), 'R' (right), 'U' (up), or 'D' (down).
– Each move transfers the player to the adjacent square in that direction, provided it is a safe square. The player cannot move more than one square at a time.
– Moving off the edge of the map has no effect. The player will remain at the same square.
– *Do not move into a hole*! Falling into a hole results in defeat.
– Locating at the grid containing the goal results in victory.
Here is the map: `<image>`
Please provide your action plan.
The reasoning process must be enclosed within `<think>...</think>` tags. The final answer must be put in `\boxed{}`.

Example Output:
`<think>` Moving RIGHT is the best choice because it allows me to avoid the hole directly below me and keeps me on a path moving closer to the goal 'G'. Moving up would lead me to a frozen square but doesn't advance me towards the goal, while moving down would immediately result in falling into a hole. Therefore, moving RIGHT is the safest and most efficient option at this moment. Next, moving DOWN is a strategic choice because it avoids the immediate danger of the hole to the right and moves the agent closer to the goal without immediate threats. Next, moving DOWN avoids nearby hazards while keeping progress toward 'G'. Next, moving DOWN continues progressing safely toward the goal. `<image>` `</think>` The answer is `\boxed{RIGHT, DOWN, DOWN, DOWN}`.

For each other benchmark, we fine-tune on 1,000 samples and use another 2,000 non-overlapping samples for reinforcement learning. Blink-Jigsaw and SAT samples are drawn at random from their official training splits. VSP provides no training set, so we follow its data generation recipe and synthesize our own data. We ensure that each map contains a valid path, no more than 20% trap blocks, and grid sizes 3–6 are produced in a 1:2:3:4 ratio (100, 200, 300, and 400 examples, respectively, for fine-tuning). Additionally, for each sample in VSP, we generate three distinct reasoning trajectories to encourage diversity. Full dataset statistics are provided in Tab. 10.

Table 8. **Data example of BLINK Jigsaw.**

BLINK Jigsaw

Thoughts Generation System Prompt:
You are a concise visual inspector. When shown: (1) a jigsaw with a missing section, (2) two candidate pieces, and (3) the fully assembled image using the correct piece, you must give a brief justification of why the chosen piece fits. Write 1–2 English sentences, focusing on visual cues such as edge continuity, color/lighting consistency, perspective, and semantics.
Example Input:
Given the first image with the lower right corner missing, can you tell which one of the second image or the third image is the missing part? Imagine which image would be more appropriate to place in the missing spot. You can also carefully observe and compare the edges of the images.
Select from the following choices:
(A) the second image
(B) the third image
`<image><image><image>`

The reasoning process must be enclosed within `<think>...</think>` tags. The final answer must be put in `\boxed{}`.
Example Output:
`<think>` Option (B) shows a person in the foreground with a light source, which disrupts the continuity of the crowd and lighting in the main image. `</think>` `<image>` `<think>` The lighting and shadows in the main image suggest a more uniform and darker environment, which is not consistent with the brighter, focused light in the wrong piece. `</think>` `\boxed{A}`.

Table 9. **Data example of SAT.**

SAT

Thoughts Generation System Prompt:
You are a concise spatial-reasoning expert. Input items may include one or more scene images, a question, an additional helper image, the available answer choices, and a special field called `TargetAnswer`.
– Use any information you need internally, but *never* mention or allude to the additional image or the `TargetAnswer`.
– Write one short English sentence that logically deduces the answer.
– Embed the additional image exactly once using the literal token `<image>`.

Example Input:
You will be given one or two images and a spatial reasoning question. Your goal is to answer the spatial question correctly from the provided choices. Now, according to the following image(s), answer the question from the provided choices.
Question: If I turn right by 33 degrees, will I be facing away from *dark frame red plaid blanket red* (near the mark 2 in the image)?
Answer choices: yes; no.
`<image>`

The reasoning process must be enclosed within `<think>...</think>` tags. The final answer must be put in `\boxed{}`.

Example Output:
`<image>` `<think>` Turning right by 33 degrees will not make you face away from the bed. `</think>` The answer is `\boxed{no}`.

Table 10. **Dataset statistics.**

Task	#SFT	#RL	#Test
VSP Spatial Reasoning	3,000	2,000	400
VSP Spatial Planning	3,000	2,000	400
Blink Jigsaw	1,000	2,000	150
SAT	1,000	2,000	500
COMT	820	820	200

Table 11. **Implementation details of supervised fine-tuning.**

Config	Value	Config	Value
Optimizer	Adam	Batch size	8
Momentum β_1	0.9	Gradient accumulation steps	2
Momentum β_2	0.95	Warmup steps	10
Weight decay	0.01	Training epochs	10
Learning rate	1e-5	Loss weight γ	10

C. Implementation Details

C.1. Implementation Details

Fine-tuning. We adopt `Qwen2.5-VL-7B-Instruct` [1] as our base VLM. The detailed training configurations are provided in Table 11. We perform supervised fine-tuning using a batch size of 8 and a cosine learning rate scheduler with an initial learning rate of 1e-5 for both stages. During fine-tuning, all components of the model are trainable except for the vision encoder. The training objective combines a cross-entropy loss for next-token prediction with a cosine similarity loss for aligning latent visual tokens, as described in Sec. 4.1. The loss weight γ for the visual alignment loss is set to the default value of 0.1. Both the training stage 1 and the training stage 2 employ the same configurations.

Table 12. **Implementation details of reinforcement learning.**

Config	Value	Config	Value
Prompt length limit	1024	Response length limit	1024
Learning rate	1e-6	Batch size	32
Gradient accumulation	4	Rollout number	5
Training epochs	15	Mini-batch size	8
σ_f	0.1	σ_c	0.9
λ_{kl}	0.01	λ_{en}	0.0

Reinforcement Learning. We adopt **VERL** [36] as the RL framework, and provide the detailed training settings in Tab. 12. Specifically, we utilize **Group Relative Policy Optimization (GRPO)** [33] for reinforcement learning. The reward function consists of a *format reward* and a *correctness reward*, weighted by σ_f and σ_c, respectively. For the accuracy reward, we set $r_{\text{acc}}(\boldsymbol{o}, \boldsymbol{x}) = 1$ if the final answer is correct, and 0 otherwise. For the format reward, we check whether the thinking process is enclosed between "`<think>`" and "`</think>`" tags and whether the final answer format is formatted as "`\boxed{}`" in the output response $\boldsymbol{o}$. If the format is correct, the reward is 0.1; otherwise, it is 0. KL regularization is applied with a coefficient of λ_{kl}, while entropy regularization is disabled in the policy loss by setting $\lambda_{en} = 0$. For our Mirage, the KL divergence on latent visual tokens is omitted during RL training.

C.2. Baselines

Anole We finetune the `Anole-7B` model with the officially released code (`https://github.com/GAIR-`

Method	Jigsaw	SAT Synthetic			SAT Real
		GoalAim	ObjM	Avg.	
Zero-Shot	0.58	0.50	0.63	0.57	0.49
Direct SFT	0.87	0.95	0.95	0.95	0.67
CoT SFT	0.83	0.97	0.90	0.94	0.66
GRPO	0.85	0.85	0.80	0.83	0.71
SFT + GRPO	0.86	0.93	0.85	0.89	0.65
Ours	**0.88**	**0.98**	**0.98**	**0.98**	**0.72**

Table 13. **Results with `Qwen2.5-VL 7B` on Jigsaw and SAT tasks.**

NLP/anole). Results are reported with the best hyper-parameters (learning rate, epochs, e.t.c.) searched in 10 runs.

MVoT We reproduce the results following the original paper [25] and released code (https://github.com/chengzu-li/MVoT). For fairness, we finetune the base model (`Anole-7B`) using the same data as in our framework, following their recipe with hyper-parameters stated in `MVoT` paper and used in the repo.

Aurora We follow the official repo (https://github.com/mahtabbigverdi/Aurora-perception) for fine-tuning Aurora models. The visual tokens are obtained from a VQGAN model trained on ImageNet with 1024 codebook dim and a downsample ratio 16. To ensure fair comparison, we change the VLM backbone from `LLaVA-1.5-13B` to `Qwen2.5-VL 7B`.

R1-VL We follow the official implementation released at https://github.com/jingyi0000/R1-VL. While the original work uses `Qwen2-VL 7B`, we reproduce results with `Qwen2.5-VL 7B` to ensure a fair comparison.

MINT-CoT Similar to R1-VL, MINT-CoT originally uses `Qwen2-VL 7B` model as the base model. We follow the official implementation released at https://github.com/xinyan-cxy/MINT-CoT#, and reproduce results with `Qwen2.5-VL 7B`, training on the provided math corpus, to ensure a fair comparison.

ViGoRL We use the official implementation at https://github.com/Gabesarch/grounded-rl. The original work adopts `Qwen2.5-VL 7B`. For reproduction, we directly use their checkpoint `ViGoRL-7B-Spatial` released at https://huggingface.co/gsarch/ViGoRL-7b-Spatial, which is specifically trained on spatial data and serves as the base model for our tasks.

Method		Jigsaw	SAT Synthetic			SAT Real
			GoalAim	ObjM	Avg.	
ViGoRL-7b-spatial	Zero-shot	0.61	0.85	0.62	0.74	0.57
	Direct SFT	0.65	0.80	0.81	0.81	0.65
	CoT SFT	0.67	0.81	0.74	0.78	0.64
ViGoRL-3b-spatial	Zero-shot	0.56	0.75	0.55	0.65	0.49
	Direct SFT	0.54	0.74	0.63	0.69	0.57
	CoT SFT	0.57	0.73	0.58	0.66	0.59

Table 14. **Detailed Results of ViGoRL on Jigsaw and SAT tasks.**

MindJourney We follow their official implementation at https://github.com/UMass-Embodied-AGI/MindJourney, with `InternVL3-14B` as the base model, larger than our original base model.

D. Experiments

D.1. Detailed Experimental Results

In this section, we provide supplementary detailed experimental results. In Tab. 13, we report results of `Qwen2.5-VL 7B` models on Jigsaw and SAT tasks. The outcomes are consistent with our findings on VSP tasks and 3B models, confirming the effectiveness of our framework. Across both benchmarks, Mirage achieves stronger performance compared with all baselines, yielding consistent performance improvement.

In the main paper, we report the best performance across both zero-shot and fine-tuning settings for each baseline. Here we provide the detailed baseline results. As shown in Tab. 14 and Tab. 15, our model universally outperforms prior baselines across Jigsaw, SAT, and CoMT tasks. Notably, these baselines have already been trained on large-scale, same-domain data for spatial reasoning or math, which explains their relatively strong zero-shot performance. Nevertheless, our method consistently achieves superior results, confirming its effectiveness.

Method	CoMT	
	Zero-Shot	CoT SFT
R1-VL 7B	0.69	0.67
Mint-CoT-7B	0.71	0.72

Table 15. **Detailed Results of R1-VL and MINT-CoT on COMT**

	3	4	5	6	Avg.
Single Image	0.64	0.70	0.71	0.68	**0.68**
Multiple Images	0.75	0.78	0.75	0.73	0.75

Table 16. **Results on VSP spatial reasoning with multiple helper images** on `Qwen2.5-VL 3B` model.

D.2. Other Ablation Study

Our framework is designed to support reasoning trajectories that interleave multiple images, though we synthesized trajectories with only one helper image per task due to the lack of high-quality interleaved datasets. To explore the impact of richer visual support, we extended experiments on the VSP spatial reasoning data with two helper images and observed a 7% average improvement on `Qwen2.5-VL 3B` models, suggesting that additional step-wise guidance benefits. Results are provided in Tab. 16. Interestingly, even when trained with a single helper image, the model can naturally generate multiple latent steps at test time when tasks demand structured, multi-step reasoning, highlighting the framework's flexibility.

		3	4	5	6	Avg.
Random Pooling	Stage 1	0.29	0.18	0.12	0.07	0.16
	Stage 2	0.70	0.57	0.49	0.34	0.52
Average Pooling	Stage 1	0.38	0.19	0.16	0.09	**0.21**
	Stage 2	0.75	0.63	0.53	0.39	**0.58**

Table 17. **Comparison results on VSP spatial planning tasks** with `Qwen2.5-VL 7B` models.

In our framework, the compression strategy is designed to condense multiple, and often semantically sparse, image tokens into compact embeddings, thereby providing stronger supervision for the first training stage. The choice of compression method directly affects the initialization for the second stage and downstream performance. Currently, we selected average pooling because it is simple, efficient, and widely adopted in feature extraction. To further investigate the impact of the compression method choice, we conducted experiments using random pooling, reported in Tab. 17. The results show that after the first training stage, average pooling outperforms random pooling by approximately 5%, and this advantage persists after the second training stage. This suggests that better supervision leads to better final performance, and that average pooling is a reasonable strategy.

D.3. Efficiency Analysis

Both training stages of Mirage are conducted on a single NVIDIA H100 GPU. Taking the VSP spatial reasoning task as an example, Stage 1 completes in approximately 3.5 hours, while Stage 2 takes around 7.2 hours. For reference, text-only CoT SFT on the same hardware requires about 5.5 hours.

During inference, our framework forwards latent embeddings directly, bypassing the need for decoding and thus saving computation in some cases. The only additional overhead comes from latent mode checking and potentially longer reasoning sequences. We compared the inference speed with text-only baselines using 7B models on both VSP tasks. As shown in Tab. 18, our framework achieves nearly identical GPU time usage on average, only 1% slower, and even slightly faster on the vsp spatial planning task. These results demonstrate that our framework incurs no extra inference cost, staying efficient and effective.

	VSP reasoning	VSP planning	Avg.
Text Baseline	1.00	1.00	1.00
Ours	1.06	0.96	1.01

Table 18. **Inference time comparison**