PROBING INTERNAL WORLD MODELS FOR INTERLEAVED VISUAL-TEXTUAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Humans often solve spatial problems by mentally simulating visual transformations. In contrast, conventional vision-language models (VLMs) reason primarily through language. We investigate whether VLMs can solve spatial problems by reasoning with both text and generated visual states. To this end, we introduce `WM-VLM`, which augments a pretrained VLM with a lightweight world model branch for generating intermediate visual states. Our two-stage training first teaches the model to generate the next visual state and then to use that state for reasoning. We programmatically construct spatial reasoning tasks with verifiable intermediate visual states. These tasks allow us to evaluate how well the model generates visual states and how much it relies on them to answer the question. On 2D and 3D mental rotation tasks, `WM-VLM` consistently outperforms the supervised fine-tuned backbone, with gains of up to 39.25 percentage points. Ablations suggest that these gains depend on the generated visual states, as removing or corrupting them sharply reduces performance. Together, these results suggest that internal world models offer a promising path toward VLMs that reason in both language and visual space.

1 INTRODUCTION

Spatial reasoning is a fundamental human ability that often involves mentally simulating visual transformations (Kosslyn et al., 1978; Shepard & Metzler, 1971). For example, to determine whether turning right would bring them closer to a chair, people can mentally simulate the turn and its visual outcome. However, prior work has documented persistent weaknesses in the spatial reasoning capabilities of VLMs (Kamath et al., 2023; Stogiannidis et al., 2025; Zhang et al., 2026b; Yang et al., 2025a; Góral et al., 2024). One possible limitation is that conventional VLMs perform intermediate reasoning primarily through text, even for visually grounded problems. Although recent reasoning VLMs perform well on mathematical and chart-based tasks (Huang et al., 2026; Yang et al., 2025b; Chen et al., 2025; Zha et al., 2026; Masry et al., 2025), many of these tasks can be solved primarily through language. Reasoning through text alone may fail to preserve the visual information needed to solve spatial problems.

Recent work seeks to enable VLMs to reason in both visual and textual spaces. Some methods use cropped or highlighted images as intermediate visual cues (Li et al., 2026a; Liu et al., 2026; Li et al., 2026b; Zhang et al., 2026a; Gao et al., 2025). These methods manipulate existing visual inputs rather than predict new visual states. Other methods interleave textual reasoning with generated visual states (Gu et al., 2026; Li et al., 2025; Yang et al., 2026b; Wang et al., 2026; Hu et al., 2026), but they are often evaluated on planning tasks, where state prediction and action planning jointly determine performance. A separate line of work uses VLMs as reasoning policies that query external world models to generate imagined observations (Yang et al., 2026a; Yu et al., 2026; Zhu et al., 2026). Although effective, these approaches delegate visual state prediction to a separate model, leaving open whether a VLM itself can learn to predict intermediate visual states that improve its spatial reasoning.

Motivated by work that frames visual generation as a form of world modeling (Wu et al., 2026; Jin et al., 2026), we study whether an internal world model can improve spatial reasoning in VLMs. Standard VLM training typically supervises only text outputs, providing no direct objective for predicting intermediate visual states. To address this limitation, we introduce `WM-VLM`, a lightweight

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

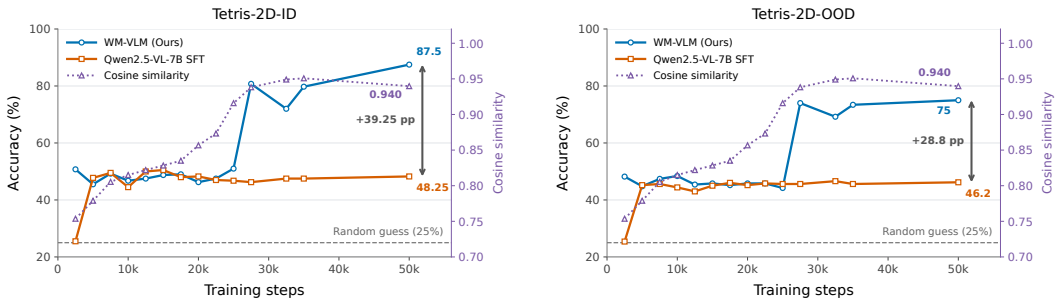


Figure 1: The internal world model module and training objective positively contributes to solving spatial-related problems, which finally helps outperform the finetuned base VLM. `WM-VLM` exhibits a sudden performance transition at around step 25k, then plateaued, yielding a 39.25 points gain on ID set and 28.8 points gain on OOD set. We train `WM-VLM` with the middle-layer configuration and evaluate each checkpoint on Tetris-2D-ID (left) and Tetris-2D-OOD (right) splits. Training step indicates the training steps taken in Stage 1. All data points are obtained following an additional Stage 2 training. Each panel compares the accuracy of `WM-VLM` and Qwen2.5-VL-7B SFT (left axis). We also show the cosine similarity between the generated visual states and ground truth embedding during training (right axis, in purple). The dashed horizontal line indicates random-guess accuracy (25%).

variant of the Mixture-of-Transformers architecture (Liang et al., 2024) with a shallow generation branch. We train the model with both vision-language understanding and visual-state generation objectives. Training proceeds in two stages. The model first learns to predict intermediate visual states and then learns to use its own predictions for reasoning. To train and evaluate these capabilities, we programmatically construct two diagnostic datasets, Tetris-2D and Tetris-3D, with explicit, verifiable intermediate states and reasoning chains.

Our experiments show that `WM-VLM` outperforms the fine-tuned backbone by 16.30–39.25 percentage points across Tetris-2D and Tetris-3D. Replacing or corrupting the generated visual states causes substantial performance drops, indicating that the model relies on them for reasoning. Training analyses further highlight the importance of the two-stage curriculum. Generated visual states become useful only after reaching sufficient quality, while joint training from scratch fails under the tested setting. Finally, a lightweight four-layer generation branch performs comparably to the full-depth MoT architecture. Our findings therefore suggest that effective visual reasoning depends on generating informative visual states and learning to use them.

2 Related Work

Visual Reasoning with VLMs Vision-language models are first trained to reason in textual space, where model generates long chain-of-thought textual reasoning to solve a problem (Huang et al., 2026; Yang et al., 2025b; Chen et al., 2025; Zha et al., 2026; Masry et al., 2025). Previous work usually evaluate their model on STEM, charts or common sense related benchmarks, e.g., MathVista (Lu et al., 2024), MMMU (Yue et al., 2024), ChartQA (Masry et al., 2022). However, not all vision-related tasks are well suited to textual reasoning. Prior work suggests that visuospatial tasks involving transformation, localization, or state tracking benefit more from visual intermediates or pixel-space operations than from purely textual reasoning (Larkin & Simon, 1987; Hu et al., 2024; Su et al., 2026).

Interleaved Visual-Textual Reasoning Prior work represents intermediate visual states in several forms and generates them through different mechanisms. Visual Sketchpad (Hu et al., 2024) and Pixel Reasoner (Su et al., 2026) use external tools to edit input images (e.g., by zooming or cropping), treating the resulting images as intermediate states. MindJourney (Yang et al., 2026a) and DreamPlan (Jia et al., 2026) invoke external world models to simulate future scenarios. Other methods use unified models, such as Anole (Chern et al., 2024) and BAGEL (Deng et al., 2025), to generate intermediate reasoning images (Chern et al., 2025; Gu et al., 2026). Mirage (Yang et al.,

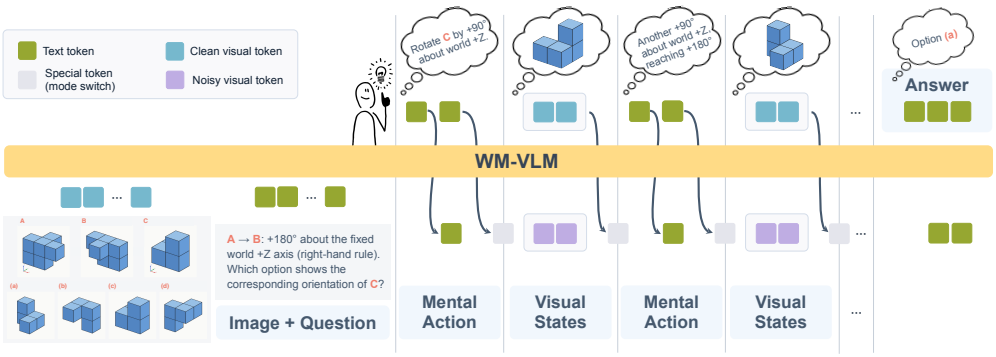


Figure 2: WM-VLM performs interleaved visual-textual reasoning by alternating between textual mental actions and generated visual states, before producing the final answer.

2026b), LVR (Li et al., 2026a), and Monet (Wang et al., 2026) train base VLMs to produce continuous visual latents, whereas LatentUM (Jin et al., 2026) produces discrete visual tokens. Recent analyses, however, question whether models with interleaved visual-textual reasoning causally depend on these visual intermediates. Viveiros et al. (2026) find that removing or corrupting latent visual tokens often has little effect, attributing this behavior to redundant intermediate supervision that enables latent bypass, inference-time representation collapse, and a substantial oracle-generation gap.

3 Method

3.1 Task Generation

To study internal world models in interleaved visual-textual reasoning, we require datasets to meet three criteria. First, solving each task should require predicting a new visual state. Second, each example should include a question, an input image, an interleaved sequence of textual actions and visual states, and a final answer. Third, both the intermediate states and the final answer should be verifiable. However, datasets with all three properties remain scarce.

Following Viveiros et al. (2026), we programmatically construct 2D and 3D spatial-reasoning datasets that satisfy these criteria. Each example contains an interleaved visual-textual reasoning trace in which intermediate visual states are needed to answer a verifiable multiple-choice question.

Each sample $s^{(i)}$ in the dataset $\mathcal{D}$ is represented as

$$s^{(i)} = \left(Q^{(i)}, I_q^{(i)}, R^{(i)}, A^{(i)}\right), \tag{1}$$

where $Q^{(i)}$ is the initial question and $I_q^{(i)}$ is the corresponding question image. The reasoning trace is denoted as $R^{(i)} = \left((T_1, I_1)^{(i)}, (T_2, I_2)^{(i)}, ..., (T_n, I_n)^{(i)}, T_{n+1}^{(i)}\right)$, where T_1 is the first mental action on the original state of the image $I_q^{(i)}$, and I_1 is the resulted visual state after applying the mental action T_1. I_j is the resulted visual state after applying the mental action T_j on the previous visual state I_{j-1}, where $j > 1$. The textual summary of the interleaved visual and textual reasoning is denoted as $T_{n+1}^{(i)}$. The final answer is denoted as $A^{(i)}$.

3.2 Interleaved Visual-Textual Reasoning with World Model

We formulate interleaved visual-textual reasoning as a Markov process (Figure 2). Let E denote the vision encoder, with $z_q = E(I_q)$ and $z_j = E(I_j)$ denoting the continuous visual embeddings of the initial image and the intermediate reasoning images, respectively. Before step j, the reasoning state H_j contains the question and the complete interleaved history, with $H_1 = (Q, z_q)$:

$$H_j = (Q, z_q, (T_1, z_1), \ldots, (T_{j-1}, z_{j-1})). \tag{2}$$

At each step, the policy model π_θ generates a piece of text including the mental action T_j. An internal world model p_θ^{wm} then predicts the resulting visual state in the embedding space:

$$T_j \sim \pi_\theta(\cdot \mid H_j), \quad \hat{z}_j \sim p_\theta^{\text{wm}}(\cdot \mid H_j, T_j). \tag{3}$$

During training, $E(I_j)$ provides the target embedding; during inference, the internal world model generates the embedding directly. The reasoning state is then updated as

$$H_{j+1} = H_j \oplus (T_j, \hat{z}_j), \tag{4}$$

where $\oplus$ denotes sequence concatenation. Thus, subsequent textual reasonings (including mental actions) are conditioned on the predicted visual outcomes of earlier steps. The model performs reasoning in this way after n steps. Then the model generates the textual summary T_{n+1} and the final answer A from the accumulated reasoning state.

3.3 VLM with An Internal World Model

We use the Mixture-of-Transformer (MoT) architecture (Liang et al., 2024) that accomodates both visual understanding and world modeling capabilities. Our model contains a understanding branch and a generation branch. The understanding branch is initialized with a pre-trained vision-language model (e.g., Qwen2.5-VL) and the generation branch weights are intialized from the corresponding understanding branch layer.

Each token x_i is routed according to its modality m_i, where $m_i \in \{\text{text}, \text{clean image}, \text{noisy image}\}$. Text and clean image tokens are routed to the understanding branch, whereas noisy image tokens are routed to the generation branch. Clean image tokens comprise the encoded question-image tokens and the visual tokens generated by the generation branch. The generation branch transforms noisy image tokens into clean image tokens.

In practice, text and clean image tokens share one set of parameters, whereas noisy image tokens are processed by a separate parameter set in the generation branch. After generation, each noisy image token is replaced by its corresponding clean image token. Attention is computed globally across tokens from both branches.

To improve training efficiency, we introduce a **light MoT** architecture (Figure 3) that retains the full understanding branch but uses a shallow generation branch. We align its k generation layers with k consecutive layers of the understanding branch. Generation tokens are processed only in these aligned layers, where the model applies global attention across tokens from both branches.We set $k = 4$ by default.

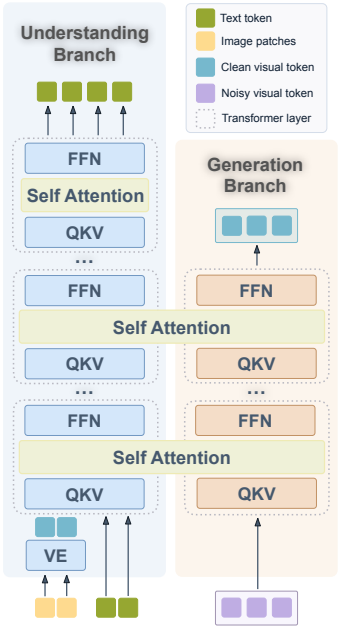


Figure 3: Architecture of WM−VLM.

3.4 Two-stage Training

Training proceeds in two stages. In the first stage, the generation branch learns to predict the next visual state conditioned on a mental action. In the second stage, the VLM learns to use the generated visual latents for downstream reasoning.

In Stage 1, we freeze the entire understanding branch, including the vision encoder and language decoder, and train only the generation branch. In Stage 2, we keep the vision encoder frozen and jointly train the language decoder and generation branch.

We optimize the generation branch with a rectified flow objective (Liu et al., 2023) and the understanding branch with a cross-entropy objective:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{flow}} + \beta \mathcal{L}_{\text{CE}}, \tag{5}$$

where α and β balance the two scalar loss terms.

Specifically, given a clean target latent $z^{(1)}$ and conditioning information c, we construct an interpolated latent $z^{(t)} = (1-t)\epsilon + tz^{(1)}$, where $\epsilon \sim \mathcal{N}(0, I)$ and $t \in [0, 1]$. Thus, $t = 0$ corresponds to noise and $t = 1$ to the clean target latent. For visual latents, parenthesized superscripts denote flow time, while subscripts denote reasoning-step indices. The generation branch predicts the velocity along this path, yielding the rectified flow objective:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{z^{(1)},c,t,\epsilon}\left[\left\| v_\theta^{\text{gen}}(z^{(t)}, t; c) - (z^{(1)} - \epsilon)\right\|_2^2\right], \tag{6}$$

where v_θ^{gen} denotes the velocity field predicted by the generation branch. Here, $z^{(1)}$ is the visual embedding obtained by passing the intermediate reasoning image through the vision encoder, and the conditioning information c includes the question and previous reasoning steps. In our experiments, we set $\alpha = 1, \beta = 0$ in Stage 1, and $\alpha = 0, \beta = 1$ in Stage 2.

At inference time, we generate $\hat{z}_j$ by integrating the learned velocity field conditioned on $c_j = (H_j, T_j)$. We then append $\hat{z}_j$ and T_j to the reasoning history to condition subsequent steps. We provide the numerical integration procedure in Appendix A.1.2.

4 Experiments

4.1 Implementation Details

4.1.1 Data Construction

We use programs to generate the interleaved visual-textual reasoning dataset. It guarantees that each sample has at least one intermediate reasoning image that is essential for answering the final question. Specifically, following Viveiros et al. (2026), we intialize 2D and 3D shapes with random number of atomic squares and cubes, respectively. We then apply a series of rotation to create different views of these shapes. Each question first presents two views of the same reference shape that illustrate a rotation. The model is then shown a new shape and asked to identify the view produced by applying the same rotation. A model capable of mental rotation should solve the task with only a short reasoning chain. We name the 2D rotation dataset as *Tetris-2D* and the 3D rotation dataset as *Tetris-3D*. Tetris-2D contains 4k training samples, 400 in-distribution (ID) and 500 out-of-distribution (OOD) test cases, respectively. Tetris-3D contains 16k training samples. Because 3D rotation is more complex and harder than 2D rotation. We include 400 easy in-domain samples (Tetris-3D-SC-ID), where the same cube shape and count appear in the training data. Additional 400 hard in-domain samples (Tetris-3D-C-ID) include seen cube count but unseen cube shapes. The 500 OOD test cases include cube shapes or counts that never appear in the training data. More details of building Tetris-2D and Tetris-3D are shown in Appendix A.2.1. We also train on ThinkMorph-SpatialNavigation (Gu et al., 2026), which combines state prediction with planning. Given a map, a starting point, and a destination, the model must find a path that avoids ice holes. At each step, it generates a visual state representing the agent's current position and plans the next move. Following Gu et al. (2026), we use the maze navigation task in the VSP benchmark (Wu et al., 2024) as our testbed.

4.1.2 Model and Training Details

We use the light mot architecture introduced in Section 3.3. The understanding branch in light mot is intialized with Qwen2.5-VL-7B-Instruct (Bai et al., 2025). For each dataset, we start with freezing the understanding branch and only train the generation branch until convergence. Then we unfreeze the understanding branch and jointly train both branches. We use an online training setting in which the understanding branch receives visual tokens generated by the generation branch rather than ground truth tokens. We keep the vision encoder frozen during the entire training. Hyperparameters are shown in Table 8, in the appendix.

For baselines, we first compare `WM-VLM` with a supervised fine-tuned Qwen2.5-VL-7B-Instruct model (Qwen2.5-VL-7B-Instruct SFT), as it provides the most direct comparison. For fairness, we optimize Qwen2.5-VL with the same steps as `WM-VLM`. Qwen2.5-VL is a VLM trained with a standard autoregressive (AR) objective and optimized using cross-entropy (CE) loss. Next, we

include LatentUM (Jin et al., 2026), which generates discrete visual tokens autoregressively. We use their pre-trained model (LatentUM-Base[1]) and conduct continual training for learning the world model and training the reasoning capability. We also include Mirage (Yang et al., 2026b), which generates continuous visual tokens autoregressively. We reuse their two-stage training approach on our datasets. Finally, we include ThinkMorph (Gu et al., 2026), which generates image pixels and is trained with a flow-matching loss. ThinkMorph is built on the unified model BAGEL, which has already been trained on large-scale interleaved visual-textual data. Following Gu et al. (2026), we simply finetune BAGEL on each of the datasets and then do evaluation. All experiments use 8 H200 GPUs if not otherwise specified.

Table 1: Performance comparison of methods with different reasoning types and training objectives. All models in this table are fine-tuned on Tetris-2D or Tetris-3D, except Qwen2.5-VL-7B-Inst. Results are accuracy (%); Δ denotes the absolute improvement of `WM-VLM` over Qwen2.5-VL-7B-Instruct SFT in percentage points. AR, FM, CE, and MSE denote autoregressive, flow matching, cross-entropy, and mean squared error, respectively. *Fine-tuned BAGEL (ThinkMorph) fails to emit images and generate answers on Tetris-3D, resulting in 0% accuracy.

Method	Visual Reasoning Type	Objective	Tetris-2D		Tetris-3D		
			ID	OOD	SC-ID	C-ID	OOD
Qwen2.5-VL-7B-Inst.	(Text only)	AR-CE	22.00	23.80	21.25	23.00	21.40
+ SFT	(Text only)	AR-CE	48.25	46.20	71.75	44.50	55.50
LatentUM	Discrete Visual Tokens	AR-CE	29.25	23.00	58.50	37.75	39.00
Mirage	Continuous Visual Tokens	AR-Cosine	47.75	43.00	66.25	46.50	49.80
ThinkMorph	Generated Image	FM-MSE	22.50	21.60	0.00*	0.00*	0.00*
`WM-VLM` (Ours)	Continuous Visual Tokens	FM-MSE	**87.50**	**75.00**	**91.00**	**66.25**	**71.80**
$\Delta_{+\text{SFT}\rightarrow\text{Ours}}$	/	/	+39.25	+28.80	+19.25	+21.75	+16.30

4.2 Results and Analysis

Vision-language models learn from image-text data using a next-token prediction objective, which does not take full advantage of the rich supervision provided by visual inputs. Consequently, VLMs fail to learn effective internal world models that can predict future visual states conditioned on the current state and mental action.

We run a comparative study to show that adding a world modeling objective is essential. Starting with Tetris-2D, we first train the generation branch in `WM-VLM` with the flow matching loss, enabling its world modeling capability (Stage 1). Then we train both the understanding and generation branch to unlock the interleaved visual-textual reasoning capability (Stage 2). We evaluate the final checkpoint on both Tetris-2D-ID and Tetris-2D-OOD.

World modeling objective boosts performance on task requires imagination. Training dynamics and evaluation results are shown in Figure 1 and Table 1, respectively. Figure 1 shows a clear performance transition at around 25k training steps. Meanwhile, the cosine similarity between the generated visual states and the ground-truth embedding also increases more rapidly around the same steps. Our assumption is that the model is only able to benefit from the world model once its generation capability exceeds a certain threshold. Qwen2.5-VL-7B-Instruct SFT demonstrates similar training dynamics before step 25k and exhibits downstream performance comparable to `WM-VLM`. However, its performance plateaus after step 25k, while `WM-VLM` enters the transition and ultimately outperforms Qwen2.5-VL-7B-Instruct SFT by 39.25 points on Tetris-2D-ID and 28.8 points on Tetris-2D-OOD.

Similar performance gains are observed on the harder Tetris-3D evaluation sets, where `WM-VLM` outperforms Qwen2.5-VL-7B-Instruct SFT by 19.25 points on Tetris-3D-SC-ID, 21.75 points on Tetris-3D-C-ID, and 16.3 points on Tetris-3D-OOD. These results show that adding a world modeling objective is essential for solving spatial reasoning tasks that require imagination. `WM-VLM` also outperforms other visual latent reasoning and interleaved visual-textual reasoning methods on Tetris-2D and Tetris-3D.

[1] `https://huggingface.co/SJTU-DENG-Lab/LatentUM-Base`

Table 2: Ablations on alternating generated image pixels on Tetris-2D. We report accuracy (%). Δ denotes the change relative to standard inference with generated pixels.

Metric	all-white pixel	shuffle 19×19 patches	Default
Tetris-2D-ID	49.25	48.25	47.25
Δ	+2.00	+1.00	–
Tetris-2D-OOD	46.60	46.20	46.20
Δ	+0.40	0.00	–

Table 3: Ablations on alternating generated visual tokens onTetris-2D. Results are accuracy (%); Δ denotes the change relative to evaluation with the default setup.

Metric	all-zero embedding	randomized tokens	shuffle visual tokens	Default
Tetris-2D-ID	14.00	23.50	81.50	87.50
Δ	-73.50	-64.00	-6.00	–
Tetris-2D-OOD	15.40	20.80	72.60	75.00
Δ	-59.60	-54.20	-2.40	–

Table 4: Ablations on alternating generated visual tokens on Tetris-3D. Results are accuracy (%); Δ denotes the change relative to evaluation with the default setup.

Metric	all-zero embedding	randomized tokens	shuffle visual tokens	default
Tetris-3D-SC-ID	15.50	23.25	81.00	91.00
Δ	-75.50	-67.75	-10.00	–
Tetris-3D-C-ID	16.00	25.00	60.25	66.25
Δ	-50.25	-41.25	-6.00	–
Tetris-3D-OOD	14.40	20.40	64.20	71.80
Δ	-57.40	-51.40	-7.60	–

Do generated visual tokens improve reasoning? The model may learn shortcuts that allow it to answer questions without relying on the generated visual tokens. To test this possibility, we conduct a retrieval experiment to evaluate the contribution of these tokens. As we have all the intermediate reasoning images in the eval set, we first encode these images to obtain the ground truth visual embeddings. Then, we get the `WM-VLM`'s generated visual embeddings, and compute the cosine similarity between the generated and ground truth embeddings. We select the top-1 retrieved ground truth embedding for each generated embedding. As shown in Table 5, higher retrieval rate leads to higher answer accuracy. Correlation analysis shows that the retrieval quality is positively correlated with the answer accuracy.

Further, we manipulate the generated visual states to see how it affects the final performance. We conduct three types of manipulation: (1) replace the generated visual tokens with all-zero embeddings, (2) replace the generated visual tokens with randomized embeddings, and (3) shuffle the generated visual tokens. As shown in Table 3 and 4, replacing the visual tokens with all-zero embeddings or randomized embeddings leads to a significant drop in accuracy. This indicates that the generated visual tokens are non-trivial and essential for assisting the model to answer the final question. It is expected that shuffling the visual tokens has a smaller impact, because visual tokens do bi-directional attention with each other in the generation branch. They already have positional information and shuffling tokens does not change that.

Does the number of visual tokens affect model performance? As the size of the reasoning images in the dataset is 512×512, the default number of visual tokens is 361 (19×19) in our experiments. We conduct an ablation study to see how the number of visual tokens affects model performance. We change the number of visual tokens by simply resizing the reasoning images. As shown in Figure 4, smaller number of visual tokens leads to lower MSE and higher cosine similarity. However, the final answer accuracy does not show the same trend. ID performance increases as the number of visual tokens increases, while OOD performance peaks at 4 (2×2) visual tokens.

What is the optimal loss ratio in Stage 2? By default, we only use the cross-entropy loss in Stage 2 ($\alpha = 0, \beta = 1$), to optimize both the understanding and generation branches. To see how the ratio between the cross-entropy and flow-matching losses affects model performance, we conduct an ablation study by varying α and β. Starting from the same checkpoint in Stage 1, we train the model with different loss ratios in Stage 2. We set $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.9\}$, with $\beta = 1 - \alpha$. As shown in Figure 5, larger weight of the flow-matching loss in Stage 2 improves the generated visual

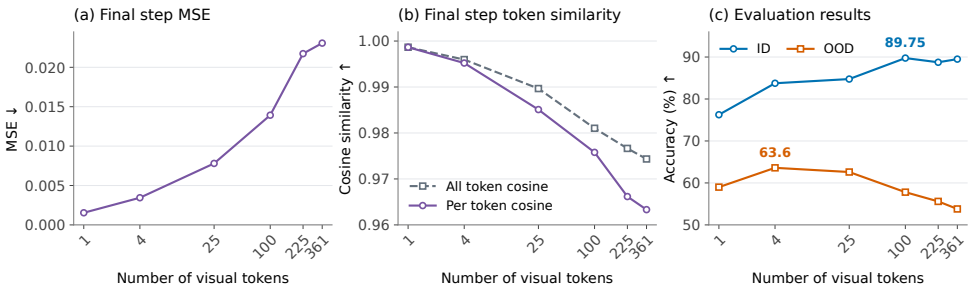


Figure 4: Effect of visual-token count on model performance. We report final-step MSE, training-time token similarity, and performance on Tetris-2D-ID and Tetris-2D-OOD. MSE compares the generated and ground-truth visual embeddings. All-token cosine is computed after concatenating all tokens in each block, whereas per-token cosine averages the similarities between corresponding generated and ground-truth tokens.

token quality. But the improved visual token quality does not necessarily lead to better downstream performance. We observe that the best overall performance is achieved when $\alpha = 0.1$ and $\beta = 0.9$, showing cross-entropy dominating the final downstream reasoning capability.

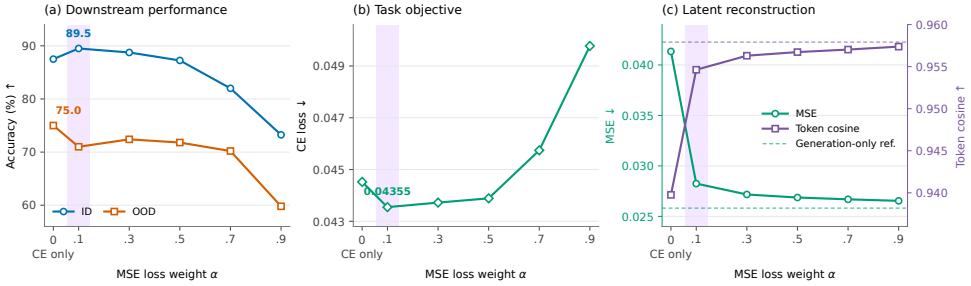


Figure 5: Effect of the ratio between cross-entropy and flow-matching losses on model performance.

Is the two-stage training necessary? Since it's possible to optimize both branches with both losses, we want to see if a single-stage approach can achieve similar performance as in the two-stage setting. From the previous experiments, we find that the optimal loss ratio is $\alpha = 0.1$ and $\beta = 0.9$. We conduct a single-stage training with this loss ratio from scratch. As shown in Figure 10, the model performance remains at random guess level and does not show any improvement. Meanwhile, both Qwen2.5-VL-7B-Instruct SFT and `WM-VLM` achieve more than 40% accuracy on the benchmarks at similar training steps. This indicates that the two-stage training is necessary for the model to first learn the world model and then learn to reason with the generated visual tokens.

How many generation layers are needed, and where should they be placed? The light MoT architecture (Figure 3) allows the generation branch to contain any number of layers and to align with any consecutive block of layers in the understanding branch. We study if alternating the number of generation layers and their placement affects model performance. First, we intialize `WM-VLM` with 1-layer generation branch and train it on Tetris-2D. The results are shown in Figure 6, where aligning generation layer with the top-most (23-27)layers in the understanding branch, the model achieves the best performance. Aligning with other layers shows lower performance, except layer-3, where there is a performance spike. Next, we scan the understanding branch with a 4-layer generation branch. As shown in Figure 8, when the number of generation layers increases, the performance gap between different placements becomes smaller. We attribute this to the increased model capacity that can mitigate the informational loss from different understanding branches. On the other hand, using the same number of generation layers with understanding layers (the original MoT setting) does not show significant performance gain.

Whhich one is the better reasoning medium? By default, we use the generated visual tokens, which is aligned with the vision encoder output, as the reasoning medium. But it is also possible to use the generated pixels to represent the intermediate reasoning state. We conduct an ablation study by simply adding an MLP on top of the generated visual tokens to reconstruct the pixels. We train the pixel-generation model with the pixel MSE loss, with the same optimizing steps as `WM-VLM`. Experiments result in Table 2 and 3 show that given the same compute budget, reasoning with the generated visual tokens yields better performance than the pixels.

Table 5: Relationship between visual-token retrieval quality and answer accuracy. Cosine denotes the average cosine similarity between the generated and ground truth visual embeddings. Retrieval top-1 denotes the percentage of times the correct ground truth embedding is retrieved. Accuracy-✓ is the accuracy when the retrieval is correct, while Accuracy-✗ is the accuracy when the retrieval is wrong. Δ denotes the absolute change relative to standard accuracy. The ϕ coefficient is computed between two binary indicators: whether the correct ground-truth embedding is retrieved and whether the final answer is correct.

Metric	Tetris-2D-ID	Tetris-2D-OOD	Tetris-3D-SC-ID	Tetris-3D-C-ID	Tetris-3D-OOD
Cosine	0.9347	0.7523	0.8511	0.7921	0.7746
Retrieval top-1	96.50	53.00	65.75	20.75	13.40
Standard accuracy	87.50	75.00	91.00	66.25	71.80
Accuracy-✓	89.38	79.25	99.24	92.77	86.57
Δ	+1.88	+4.25	+8.24	+26.52	+14.77
Accuracy-✗	35.71	70.21	75.18	59.31	69.52
Δ	-51.79	-4.79	-15.82	-6.94	-2.28
ϕ correlation	0.298	0.104	0.399	0.287	0.129



Figure 6: Layer-wise analysis of visual latent quality in the VLM.

5 Conclusion

To enable VLMs to reason about space through visual imagination, we introduce `WM-VLM`, a VLM equipped with an internal world model. We construct datasets with verifiable intermediate visual states and use them to study how world modeling affects interleaved visual-textual reasoning. Our experiments show that `WM-VLM` improves spatial reasoning performance and relies on its generated visual tokens to produce final answers. Ablations identify the number of visual tokens, the balance between cross-entropy and flow-matching losses, and two-stage training as important design choices. More broadly, internal world models may enable VLMs to reason jointly in language and visual space.

AI USE STATEMENT

In this work, we used generative AI tools in the following areas: "implement methods, support qualitative and thematic data analysis, interpret results". We have not used generative AI tools in the following areas: "generate synthetic data sets, help develop theoretical models or conceptual frameworks, design or provide feedback on research methodology or experiments, clean and reformat dataset", and "formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation" are not applicable to this work. Additionally, we used generative AI tools for polishing paper writing. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by human authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We will open-source the code used to generate the datasets. We'll also upload the datasets to a public repository. The code for training and evaluation will also be open-sourced. We have included necessary implementation details in the main text and appendix.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025. doi: 10.48550/ARXIV.2502.13923. URL https://doi.org/10.48550/arXiv.2502.13923.

Lei Chen, Xuanle Zhao, Zhixiong Zeng, Jing Huang, Yufeng Zhong, and Lin Ma. Chart-r1: Chain-of-thought supervision and reinforcement for advanced chart reasoner. *CoRR*, abs/2507.15509, 2025. doi: 10.48550/ARXIV.2507.15509. URL https://doi.org/10.48550/arXiv.2507.15509.

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.

Ethan Chern, Zhulin Hu, Steffi Chern, Siqi Kou, Jiadi Su, Yan Ma, Zhijie Deng, and Pengfei Liu. Thinking with generated images. *CoRR*, abs/2505.22525, 2025. doi: 10.48550/ARXIV.2505.22525. URL https://doi.org/10.48550/arXiv.2505.22525.

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

Jun Gao, Yongqi Li, Ziqiang Cao, and Wenjie Li. Interleaved-modal chain-of-thought. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19520–19529. IEEE, 2025.

Gracjan Góral, Alicja Ziarko, Michal Nauman, and Maciej Wolczyk. Seeing through their eyes: Evaluating visual perspective taking in vision language models. *CoRR*, abs/2409.12969, 2024. doi: 10.48550/ARXIV.2409.12969. URL https://doi.org/10.48550/arXiv.2409.12969.

Jiawei Gu, Yunzhuo Hao, Huichen Wang, Linjie Li, Michael Qizhe Shieh, Yejin Choi, Ranjay Krishna, and Yu Cheng. Thinkmorph: Emergent properties in multimodal interleaved chain-of-thought reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 141405–141447, 2026.

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. *Advances in Neural Information Processing Systems*, 37:139348–139379, 2024.

Zican Hu, Xuyang Hu, Yiming Liu, Zuwei Long, Wei Liu, Yunzhuo Hao, Jiawei Gu, Linjie Li, Yu Cheng, Zhenhong Sun, Weibo Gu, Xing Sun, and Zhi Wang. Bridging interleaved multimodal reasoning as a unified decision process. *CoRR*, abs/2607.03748, 2026. doi: 10.48550/ARXIV.2607.03748. URL https://doi.org/10.48550/arXiv.2607.03748.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, Zheyu Ye, Zhe Xu, Yao Hu, Shaohui Lin, et al. Vision-r1: Incentivizing reasoning capability in multimodal large language models. In *International Conference on Learning Representations*, volume 2026, pp. 63794–63812, 2026.

Emily Yue-Ting Jia, Weiduo Yuan, Tianheng Shi, Vitor Guizilini, Jiageng Mao, and Yue Wang. Dreamplan: Efficient reinforcement fine-tuning of vision-language planners via video world models. *arXiv preprint arXiv:2603.16860*, 2026.

Jiachun Jin, Zetong Zhou, Xiao Yang, Hao Zhang, Pengfei Liu, Jun Zhu, and Zhijie Deng. Latentum: Unleashing the potential of interleaved cross-modal reasoning via a latent-space unified model. *arXiv preprint arXiv:2604.02097*, 2026.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 9161–9175. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.568. URL https://doi.org/10.18653/v1/2023.emnlp-main.568.

Stephen M. Kosslyn, Thomas M. Ball, and Brian J. Reiser. Visual images preserve metric spatial information: Evidence from studies of image scanning. *Journal of Experimental Psychology: Human Perception and Performance*, 4(1):47–60, 1978. doi: 10.1037/0096-1523.4.1.47.

Jill H Larkin and Herbert A Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive science*, 11(1):65–100, 1987.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 148076–148090, 2026a.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulic, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/li25cz.html.

Kelvin Li, Chuyi Shang, Leonid Karlinsky, Rogerio Feris, Trevor Darrell, and Roei Herzig. Latent implicit visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33457–33466, 2026b.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Chengzhi Liu, Yuzhe Yang, Yue Fan, Qingyue Wei, Sheng Liu, and Xin Eric Wang. Reasoning within the mind: Dynamic multimodal interleaving in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9225–9236, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

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

Ahmed Masry, Abhay Puri, Masoud Hashemi, Juan A. Rodríguez, Megh Thakkar, Khyati Mahajan, Vikas Yadav, Sathwik Tejaswi Madhusudhan, Alexandre Piché, Dzmitry Bahdanau, Christopher Pal, David Vázquez, Enamul Hoque, Perouz Taslakian, Sai Rajeswar, and Spandana Gella. Bigcharts-r1: Enhanced chart reasoning with visual reinforcement finetuning. *CoRR*, abs/2508.09804, 2025. doi: 10.48550/ARXIV.2508.09804. URL https://doi.org/10.48550/arXiv.2508.09804.

Roger N. Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. doi: 10.1126/science.171.3972.701.

Ilias Stogiannidis, Steven McDonagh, and Sotirios A. Tsaftaris. Mind the gap: Benchmarking spatial reasoning in vision-language models. *CoRR*, abs/2503.19707, 2025. doi: 10.48550/ARXIV.2503.19707. URL https://doi.org/10.48550/arXiv.2503.19707.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. *Advances in Neural Information Processing Systems*, 38:8222–8251, 2026.

André G Viveiros, Nuno Gonçalves, André FT Martins, and Matthias Lindemann. What's holding back latent visual reasoning? *arXiv preprint arXiv:2605.18445*, 2026.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond image and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12030–12040, 2026.

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

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.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 10632–10643. Computer Vision Foundation / IEEE, 2025a. doi: 10.1109/CVPR52734.2025.00994. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_Thinking_in_Space_How_Multimodal_Large_Language_Models_See_Remember_CVPR_2025_paper.html.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, Bo Zhang, and Wei Chen. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 2376–2385. IEEE, 2025b. doi: 10.1109/ICCV51701.2025.00229. URL https://doi.org/10.1109/ICCV51701.2025.00229.

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. *Advances in Neural Information Processing Systems*, 38:109855–109885, 2026a.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33510–33520, 2026b.

Shoubin Yu, Yue Zhang, Zun Wang, Jaehong Yoon, Huaxiu Yao, Mingyu Ding, and Mohit Bansal. When and how much to imagine: Adaptive test-time scaling with world models for visual spatial reasoning. *arXiv preprint arXiv:2602.08236*, 2026.

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

Yuheng Zha, Kun Zhou, Yujia Wu, Yushu Wang, Jie Feng, Zhi Xu, Shibo Hao, Zhengzhong Liu, Eric P Xing, and Zhiting Hu. Vision-g1: Towards general reasoning vision-language models via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28131–28139, 2026.

Xin Zhang, Qiqi Tao, Jiawei Du, Moyun Liu, and Joey Tianyi Zhou. Visual latents know more than they say: Unsilencing latent reasoning in mllms. *arXiv preprint arXiv:2605.02735*, 2026a.

Yuyou Zhang, Radu Corcodel, Chiori Hori, Anoop Cherian, and Ding Zhao. Spinbench: Perspective and rotation as a lens on spatial reasoning in vlms. In *International Conference on Learning Representations*, volume 2026, pp. 70072–70141, 2026b.

Chenming Zhu, Jingli Lin, Yilin Long, Peizhou Cao, Tai Wang, Jiangmiao Pang, and Xihui Liu. Thinking with imagination: Agentic visual spatial reasoning with world simulators. *arXiv preprint arXiv:2606.06476*, 2026.

A Appendix

A.1 Implementation Details

A.1.1 VLM with an internal world model

In `WM-VLM`, each token's query, key, and value are computed as follows:

$$Q_i = x_i W_Q^{m_i}, \quad K_i = x_i W_K^{m_i}, \quad V_i = x_i W_V^{m_i}, \tag{7}$$

where $W_Q^{m_i}$, $W_K^{m_i}$, and $W_V^{m_i}$ are the learnable weight matrices for the query, key, and value projections of modality m_i, respectively.

The attention output a_i for token x_i in a transformer layer is:

$$a_i = \left[\mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} \right) V \right]_i W_O^{m_i}, \tag{8}$$

where $W_O^{m_i}$ is the learnable weight matrix for the output projection of modality m_i.

A.1.2 Rectified Flow Sampling

At inference time, we generate the estimated visual state $\hat{z}_j$ conditioned on $c_j = (H_j, T_j)$ by integrating the predicted velocity field from noise ($t = 0$) to the clean latent ($t = 1$). Specifically, we initialize $\tilde{z}_j^{(0)} \sim \mathcal{N}(0, I)$ and use K Euler steps with $t_k = k/K$:

$$\tilde{z}_j^{(t_{k+1})} = \tilde{z}_j^{(t_k)} + \frac{1}{K} v_\theta^{\mathrm{gen}}\left(\tilde{z}_j^{(t_k)}, t_k; c_j \right), \qquad k = 0, \ldots, K-1. \tag{9}$$

The final estimate is $\hat{z}_j = \tilde{z}_j^{(1)}$, which is appended to the reasoning history together with T_j to condition subsequent reasoning steps. Here, K denotes the number of Euler steps, while t denotes the continuous flow time.

A.1.3 Training Hyperparameters

We present the training hyperparameters in Table 8. All training tasks are conducted on 8 H200 GPUs.

A.2 Data Construction

A.2.1 Building Tetris-2D and Tetris-3D

We use programs to generate the interleaved visual-textual reasoning dataset. Detailed algorithm is shown in Algorithm 1. The 2D rotation dataset is constructed in a similar way, except that we use squares instead of cubes.

Algorithm 1 Construction of the Tetris-3D Dataset

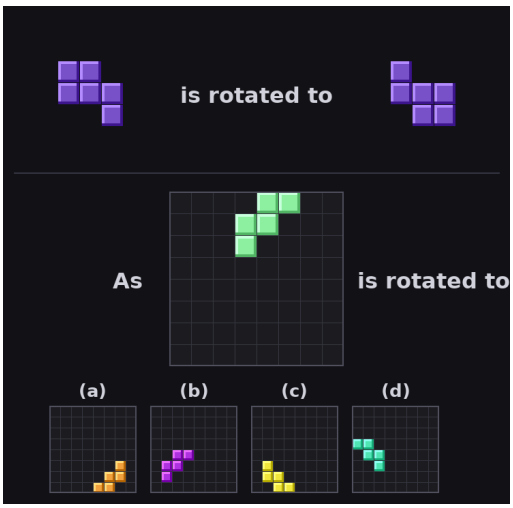
Require: Cube counts $\{4, 5, 6, 7\}$, rotation axes $\{X, Y, Z\}$, rotation angles $\{90°, 180°, 270°\}$
Ensure: Training set $\mathcal{D}_{\text{train}}$ and evaluation sets $\mathcal{D}_{\text{ID}}, \mathcal{D}_{\text{OOD}}$

$\mathcal{S} \leftarrow \emptyset$
for $n \in \{4, 5, 6, 7\}$ **do**
    Enumerate all connected shapes consisting of n cubes
    Remove shapes equivalent under proper 3D rotations
    Add the remaining canonical shapes to $\mathcal{S}$
end for
Split $\mathcal{S}$ into seen shapes $\mathcal{S}_{\text{seen}}$ and held-out shapes $\mathcal{S}_{\text{OOD}}$ ▷ Mirror-related shapes stay in the same split
$\mathcal{Q}_{\text{seen}} \leftarrow \emptyset$, $\mathcal{Q}_{\text{OOD}} \leftarrow \emptyset$
for each shape $S \in \mathcal{S}$ **do**
    for each distinct initial orientation P of S **do**
        for $a \in \{X, Y, Z\}$ and $\theta \in \{90°, 180°, 270°\}$ **do**
            Compute the rotated orientation $P' = R(a, \theta)P$
            if $P' \neq P$ **then**
                Add (S, P, a, θ, P') to the corresponding seen or OOD configuration pool
            end if
        end for
    end for
end for
Split the seen configuration pool into disjoint training and ID-evaluation pools
for each selected configuration $(S_C, P_C, a, \theta, P'_C)$ **do**
    Select a reference shape S_A with a different cube count
    Rotate S_A by the same (a, θ) to produce Reference B
    Use P'_C as the correct answer
    Sample three distinct rotated poses of S_C as distractors
    Randomize the position of the four answer options
    Render Reference A, Reference B, Query C, and the four options
end for
Construct $\mathcal{D}_{\text{train}}$ from training configurations
Construct $\mathcal{D}_{\text{ID}}$ from unseen configurations of seen shapes
Construct $\mathcal{D}_{\text{OOD}}$ from held-out shapes

A.2.2 Data Samples

Figure 7 shows representative training examples from Tetris-2D and Tetris-3D. Each example contains a question image, textual mental actions, visual states produced by those actions, and a final textual answer. Tetris-2D stores one visual state after the complete rotation sequence, whereas Tetris-3D stores a visual state after each atomic rotation.

(a) Tetris-2D



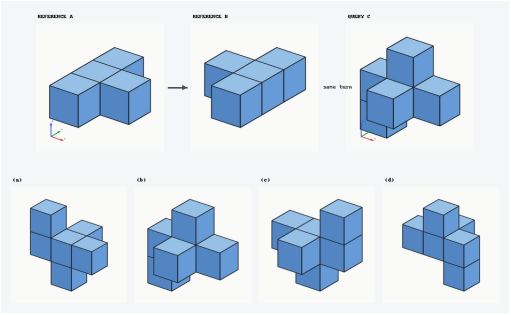
T_1, T_2**:** Rotate the query shape 90° clockwise twice.

↓



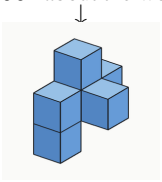
Final visual state
Compare with the options. **Answer: (a)**

(b) Tetris-3D



T_1**:** Rotate $+90^\circ$ about the world $+Z$ axis.

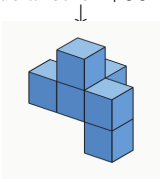
↓



I_1

T_2**:** Continue another $+90^\circ$ about $+Z$.

↓



I_2

Compare with the options. **Answer: (d)**

Figure 7: Representative samples from Tetris-2D and Tetris-3D. Text denotes mental rotation actions and images denote their resulting visual states. The 2D example stores the final state after two 90° rotations, while the 3D example explicitly interleaves each 90° rotation with its corresponding visual state.

A.2.3 Comparison between Different VLMs

We test different VLMs with different reasoning types on 2D mental rotation tasks. The results are shown in Table 6. We observe that stronger VLMs, i.e., with strong textual reasoning capabilities, can first convert the 2D or 3D shape into coordinates. They perform textual reasoning to rotate these coordinates and get the answer. We point out that this is not the general and scalable visual reasoning approach because language is lossy.

A.3 Additional Experiment Results

Initialization is not decisive. In our main experiments, generation branch is initialized with the same weights as the understanding branch. To weigh the factor of the intialization method, we conduct an ablation study by initializing the generation branch with random weights. As shown in Figure 11, in all three layer placement settings, neither initialization method shows a clear advantage over the other. This indicates that the initialization method is not decisive for the final performance.

Experiments on visual planning tasks. We further evaluate our model on visual planning tasks, which require predicting a sequence of actions to reach a goal state. We train on ThinkMorph-SN and evaluate on VSP-Nav. As shown in Table 7, our model, which generates continuous visual tokens as intermediate thoughts, shows competitive performance compared with baselines. BAGEL generates images as intermediate thoughts and achieves the best performance, likely because it was pre-trained on large-scale interleaved data. In contrast, on the Tetris tasks, where BAGEL appears not to benefit from similar pre-training data, it underperforms our model and even collapses during training.

Table 6: Evaluation results of VLMs on the test sets of Tetris-2D, Tetris-3D, and VSP-Nav under different reasoning-input settings. Think Mode is the model's built-in capability from their pre-training. Qwen2.5-VL-7B-Instruct and Qwen3-VL-8B-Instruct do not support think mode by default. Qwen3.5-9B, Qwen3.6-27B, and Qwen3.7-40B support think mode. We test these models in both modes. ✗ indicates the think mode is disabled; ✓ indicates the think mode is enabled. Q denotes the question input, RT denotes reasoning text, and I denotes intermediate reasoning images. When testing with the Q + RT setting, we replace the intermediate reasoning images with token "[IMAGINATION]".

Model	Think Mode	Eval Setting	Tetris-2D		Tetris-3D		VSP-Nav
			ID	OOD	ID	OOD	
Qwen2.5-VL-7B-Instruct	✗	Question only	22.00	23.80	23.00	21.40	6.50
		Q + RT	23.50	23.20	20.25	21.00	21.50
		Q + RT + I	26.25	26.20	60.50	59.60	24.83
Qwen3-VL-8B-Instruct	✗	Question only	23.00	21.80	9.00	9.20	1.33
		Q + RT	20.50	21.00	7.00	7.40	0.67
		Q + RT + I	33.75	38.40	53.00	46.60	0.50
Qwen3.5-9B	✗	Question only	29.25	24.20	18.50	19.20	0.17
		Q + RT	27.25	23.00	23.25	21.40	3.00
		Q + RT + I	30.50	29.80	58.50	61.80	42.50
	✓	Question only	42.50	41.80	22.25	23.80	61.00
		Q + RT	43.75	40.00	24.00	28.00	81.00
		Q + RT + I	44.50	41.20	40.50	46.80	75.67
Qwen3.6-27B	✗	Question only	43.50	45.60	18.25	21.60	0.00
		Q + RT	48.25	46.20	12.00	12.20	15.33
		Q + RT + I	28.50	31.20	66.00	68.40	2.83
	✓	Question only	56.00	47.80	25.25	26.60	71.17
		Q + RT	60.50	46.00	26.00	21.80	91.83
		Q + RT + I	55.50	51.60	63.75	66.00	95.33
Qwen3.8-27B	✗	Question only	67.00	64.20	29.25	26.40	16.00
		Q + RT	65.00	63.20	27.50	27.40	48.00
		Q + RT + I	61.25	60.60	53.75	55.00	95.67
	✓	Question only	93.50	85.40	39.00	32.20	88.00
		Q + RT	93.00	88.60	32.00	32.40	99.67
		Q + RT + I	87.50	84.00	85.75	87.80	99.83

Table 7: Maze-navigation results across different training datasets and grid sizes. Accuracy and per-grid-size results are reported in percent.

Training Data	Method	WM Objective	Accuracy	3×3	4×4	5×5	6×6	7×7	8×8
Proprietary interleaved data + ThinkMorph-SN	ThinkMorph (BAGEL)	✓	82.50	93	95	94	78	74	61
/	Qwen2.5-VL-7B-Inst.	✗	6.50	23	4	5	3	2	2
ThinkMorph-SN	Qwen2.5-VL-7B-Inst. SFT	✗	43.17	79	65	50	34	17	14
	LatentUM	✓	52.00	90	75	59	42	24	22
	Mirage	✓	42.50	79	58	56	29	21	12
	WM-VLM (Ours)	✓	50.00	98	83	63	31	18	7

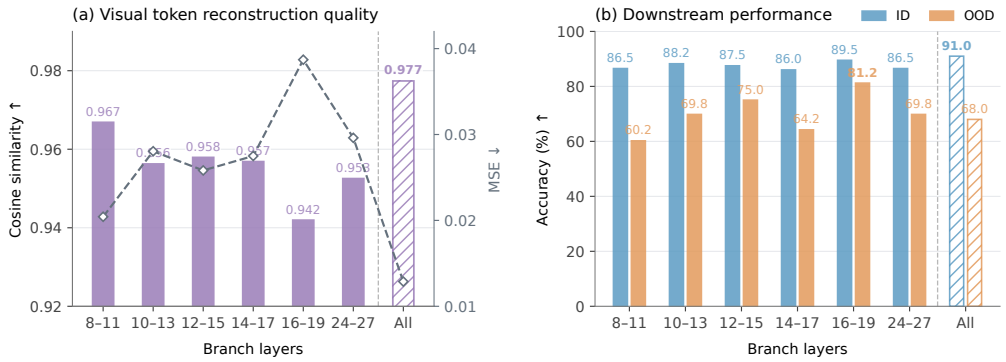


Figure 8: Effect of the layer-window selection on model performance. For example, ”8-11” means the generation branch is connected to the understanding branch from layer 8 to layer 11.

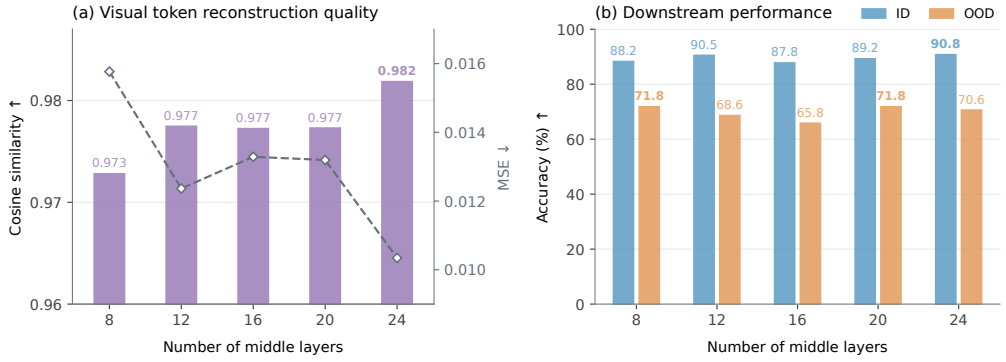


Figure 9: Effect of the number of middle layers on model performance.

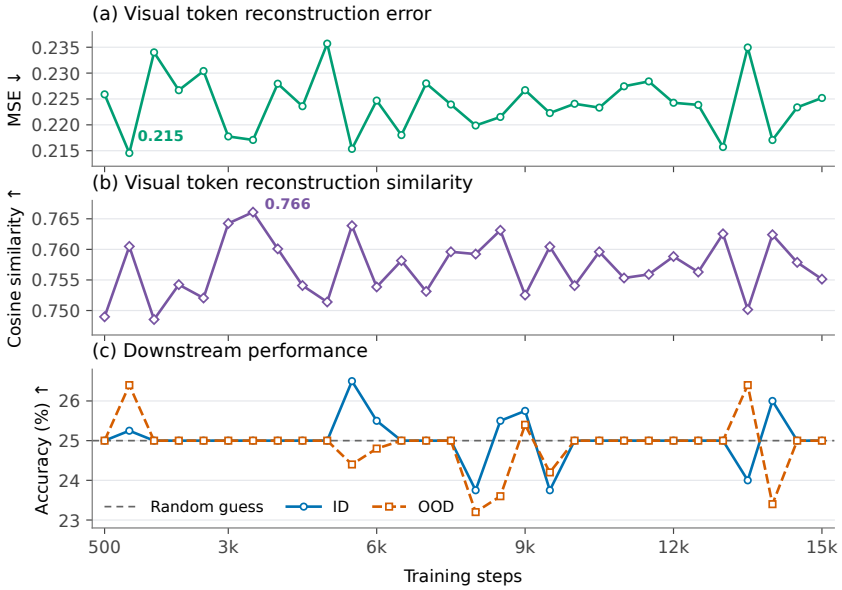


Figure 10: Training curves for joint training with $\alpha = 0.1$ and $\beta = 0.9$.

Table 8: Hyperparameters for the two-stage training procedure on Tetris-2D.

Hyperparameter	**Stage 1**	**Stage 2**
Initialization	Qwen2.5-VL-7B-Instruct	Stage 1 checkpoint
Learning rate	1e-4	1e-5
Optimizer	AdamW	AdamW
LR scheduler	Constant	Cosine
Warmup ratio	0	0.03
Weight decay	0	0.01
Per-device batch size	1	1
GPU Model	H200	H200
Number of GPUs	8	8
Gradient accumulation steps	1	1
Effective global batch size	8	8
Precision	BF16	BF16
Flow/MSE loss weight	1	0
Pixel loss weight	0	0
CE loss weight	0	1
Loss formula	$1.0 \times$ flow MSE	$1.0 \times$ CE
VLM backbone	Frozen	Trainable
Vision encoder	Frozen	Frozen
Latent-generation branch	Trainable	Trainable
Trainable parameters	0.959B	8.572B
Frozen parameters	8.289B	0.677B
Total parameters	9.248B	9.248B
Trainable ratio	10.37%	92.68%

Additional results on generation layers. We further vary the number of generation layers from 8 to 24. As shown in Figure 9, increasing the generation-branch depth generally improves visual-state reconstruction quality, but does not consistently improve downstream accuracy. This suggests that better reconstruction alone does not necessarily translate into better reasoning performance.

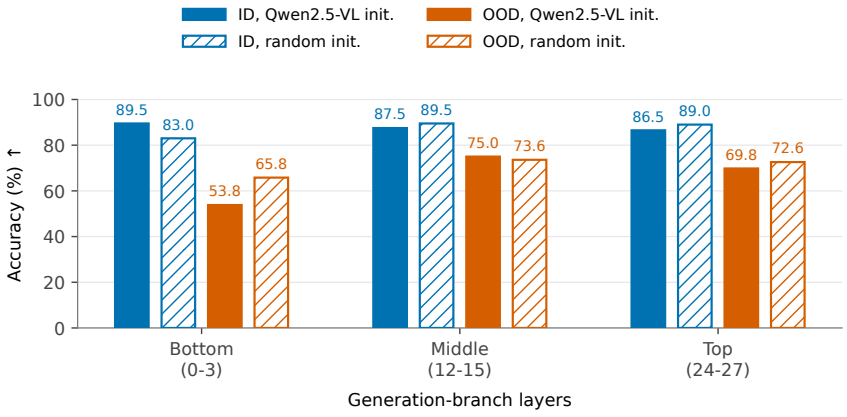


Figure 11: Effect of generation-branch initialization on downstream performance across different insertion depths.

A.4 Future Work

Our training paradigm actually support pre-training a VLM with an internal world model. The interleaved visual-textual reasoning dataset is required. However, current interleaved datasets are mainly

from static sources, e.g., from a webpage. It is worth pointing out that the ideal case is that we build a dataset with embodied experience. Each physical action in the dataset has a corresponding visual outcome. The full trajectory, including the original instruction and observation, the interleaved actions and observations, and the final outcome, can be collected from either real-world robots or simulated embodied agents. This VLM with world model can do mental imagination by nature and will significantly boost the performance for spatial reasoning and agentic planning. We leave this data and training paradigm for future work.