VisuoThink: Empowering LVLM Reasoning with Multimodal Tree Search

Yikun Wang[*1,2], **Siyin Wang**[*1,2], **Qinyuan Cheng**[1], **Zhaoye Fei**[1], **Liang Ding**[3], **Qipeng Guo**[2,4], **Dacheng Tao**[†5], **Xipeng Qiu**[†1,2]

[1] Fudan University [2] Shanghai Innovation Institute
[3] The University of Sydney [4] Shanghai AI Laboratory [5] Nanyang Technological University
yikunwang19@fudan.edu.cn

Abstract

Recent advancements in Large Vision-Language Models have showcased remarkable capabilities. However, they often falter when confronted with complex reasoning tasks that humans typically address through visual aids and deliberate, step-by-step thinking. While existing methods have explored text-based slow thinking or rudimentary visual assistance, they fall short of capturing the intricate, interleaved nature of human visual-verbal reasoning processes. To overcome these limitations and inspired by the mechanisms of slow thinking in human cognition, we introduce ***VisuoThink***, a novel framework that seamlessly integrates visuospatial and linguistic domains. *VisuoThink* facilitates multimodal slow thinking by enabling progressive visual-textual reasoning and incorporates test-time scaling through look-ahead tree search. Extensive experiments demonstrate that *VisuoThink* significantly enhances reasoning capabilities via inference-time scaling, even without fine-tuning, achieving state-of-the-art performance in tasks involving geometry and spatial reasoning. Our code has been open-sourced at https://github.com/ekonwang/VisuoThink.

1 Introduction

Recent advances in Large Vision-Language Models (LVLMs) (OpenAI, 2024a; Team, 2024) have shown remarkable progress across a variety of tasks. However, these models often struggle with complex reasoning challenges, such as geometric problem-solving (Qiao et al., 2024; Cherian et al., 2024) or spatial reasoning (Ramakrishnan et al., 2024; Wu et al., 2024), where human problem-solving approaches typically rely on visual aids. For example, when solving geometry problems, humans often iteratively sketch auxiliary lines or visualize intermediate steps, while exploring different reasoning paths - a form of "slow thinking" (Kahneman, 2011) that combines visual and verbal cognitive processes.

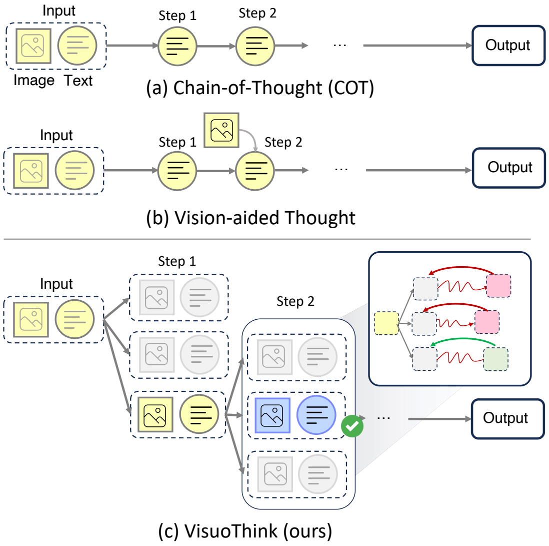

Figure 1: Illustration of Input-Output Prompting, *CoT*, Vision-aided Thought and our *VisuoThink*. Vision-aided Thought often relies on reasoning with one-step or unreliable multi-step visual cues (generated by LVLMs). While *VisuoThink* addresses this gap through tool-augmented visual hints, coupled with a predictive-rollout search mechanism to systematically optimize reasoning capability.

With the success of o1 series models (OpenAI, 2024b), researchers have explored language as a medium for implementing slow thinking, coupled with test-time scaling techniques (Zeng et al., 2024). Given the inherently multimodal nature of reality, early efforts (Xu et al., 2024; Thawakar et al., 2025; Yao et al., 2024; Du et al., 2025) have attempted to extend such deliberative thinking to multimodal reasoning. However, even augmented with search strategy, these methods treat visual information merely as static input, relying solely on textual reasoning chains during the reasoning process - creating a "visual blind spot", where the

[*] Yikun and Siyin contributed equally.

[†] Please correspond to dacheng.tao@gmail.com and xpqiu@fudan.edu.cn

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 21707–21719
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

potential for visual information throughout the reasoning process is largely ignored (Fig. 1a). On the other hand, while approaches like VisualSketchpad (Hu et al., 2024) and VoT (Wu et al., 2024) have recognized the importance of visual information by incorporating visual aids in reasoning (Fig. 1b), they mainly focus on single-step assistance or simplified visual hints (e.g., emojis). These methods lack the multi-step visual-textual interleaved reasoning process that characterizes human slow thinking, while failing to explore potential search strategies.

To address these limitations, we propose ***VisuoThink***, a multimodal tree search framework that systematically explores multiple reasoning paths with vision-text interleaved thinking at each step. Unlike previous approaches, Visuothink (Fig. 1c) enables multimodal slow thinking through two key innovations: (1) a step-by-step vision-text interleaved reasoning framework that dynamically utilizes multi-step visual aids from tool uses, and (2) a look-ahead tree search algorithm that explores multiple reasoning paths, enabling test-time scaling of the reasoning process. Specifically, our look-ahead tree search incorporates a predictive rollout mechanism that simulates the likely outcomes of different reasoning states. This allows the model to prioritize more promising paths and avoid less ones, guiding the reasoning process toward the optimal solution. Through this test-time scaling capability, the model can thoroughly explore and optimize reasoning paths dynamically during inference.

Our empirical evaluation demonstrates that Visuothink significantly outperforms existing methods across various reasoning tasks, particularly in geometry and spatial reasoning domains. On Geomeverse, Our methods achieves an accuracy@1 as high as *48.5%*, with an improvement of as high as *21.8%* over the state-of-the-art baseline, which particularly shows strong performance of VisuoThink on problems requiring multi-step visual reasoning. Through extensive ablation studies, we show that each component of our framework contributes meaningfully to its overall performance.

In summary, our contributions include:

- We propose a novel reasoning paradigm, multimodal tree search, for multimodal slow thinking that enables dynamic integration of visual and verbal reasoning paths throughout the problem-solving search process.
- We extend test-time scaling methods to the visual domain by proposing a predictive rollout mechanism that explores and optimizes visual reasoning paths by predicting future states.
- We demonstrate substantial empirical improvements across multiple reasoning tasks, particularly in geometry and spatial reasoning, with detailed analyses revealing key insights about our approach.

2 Related Work

2.1 Text-centric Reasoning in LVLMs

With the emergence of o1 models (OpenAI, 2024b), the importance of slow thinking has become increasingly evident (Zeng et al., 2024). Several works have attempted to extend this to LVLMs through methods like stage-wise reasoning (Xu et al., 2024), curriculum learning (Thawakar et al., 2025), tree search-based data generation (Yao et al., 2024), and LLM distillation (Du et al., 2025). However, these methods treat visual information as static input, relying only on textual data during reasoning, which limits their ability to fully leverage multimodal information for complex tasks.

2.2 Vision-aided Reasoning

Recent advancements in multimodal reasoning have demonstrated that incorporating visual information provides richer context and hints compared to text-only approaches. Early studies adopted a two-stage approach, where visual information is first transformed and grounded into text (Zhang et al., 2023), graph structures (e.g., scene graphs (Mitra et al., 2023) or knowledge graphs (Mondal et al., 2024)), or bounding boxes (Lei et al., 2024), followed by reasoning. Other works leverage existing vision models (e.g., segmentation, detection) to process input images into valuable cues for perception, enabling more precise image-understanding with fine-grained visual information (Yang et al., 2023; Zhou et al., 2024; Gao et al., 2024).

Another sequence of research focuses on intermediate visual representations to enhance reasoning. For instance, Visual Sketchpad (Hu et al., 2024) employs Python-based drawing tools to generate sketches as intermediate visual aids for geometric problems, while VoT (Wu et al., 2024) formalizes visual thinking by generating emoji-like textual representations. MVOT (Li et al., 2025) fine-tunes multimodal models to generate images

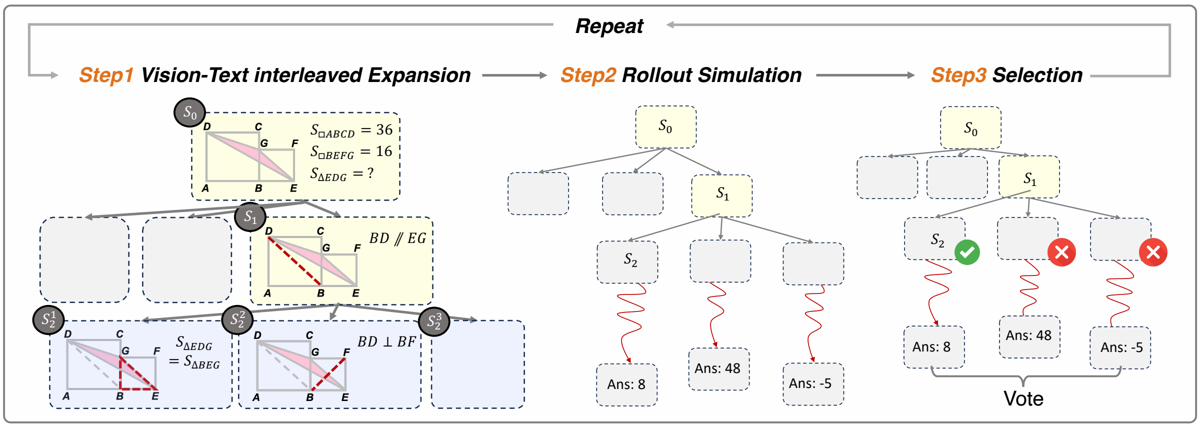

Figure 2: The illustration of our *VisuoThink* framework with three stages: (1) vision-text interleaved expansion: generates candidate paths through vision-text interleaved thinking; (2) rollout simulation: sample candidate reasoning nodes and then perform look-ahead search to better evaluate the value of current states; (3) selection: selects the most promising path via self-voting with results or states from rollout.

during reasoning, allowing the model to create visual aids dynamically. Despite these advancements, most existing methods rely on single-step or unreliable visual representations, lacking search mechanisms to test-time scaling through exploring multiple reasoning paths. In contrast, we develop a multimodal tree search framework that both leverages multi-step visual cues during reasoning and systematically explores reasoning paths through tree search.

2.3 Test-time Scaling with Tree Search

Scaling compute at test time has emerged as a powerful strategy to enhance LLMs' reasoning capabilities without increasing model parameters (Snell et al., 2024; Bi et al., 2025a,b). Various approaches including BoN (Gui et al., 2024; Sun et al., 2024; Amini et al., 2024), guided beam search (Xie et al., 2023; Yu et al., 2023), and Monte Carlo Tree Search (MCTS) (Feng et al., 2023; Liu et al., 2023; Chen et al., 2024) have been explored for text models, demonstrating improved performance through different search strategies. However, the exploration of test-time scaling in LVLMs remains limited. Prior work like AtomThink (Xiang et al., 2024) has only investigated basic methods such as beam search, with text-only reasoning chains. In contrast, our method introduces vision-text interleaved thinking with look-ahead search, extending test-time scaling to multimodal reasoning.

3 VisuoThink

We propose ***VisuoThink***, a novel framework for multimodal reasoning that dynamically integrates visual and textual information during the inference process. At its core, our framework implements multimodal slow thinking through a key mechanism: predictive rollout search that allows models to *think* ahead.

3.1 Vision-Text Interleaved Thinking

Our framework facilitates vision-text interleaved reasoning through an iterative cycle of **Thought**, **Action**, and **Observation** like existing work (Yao et al., 2023), which enables natural and dynamic interactions with external tools. (1) Thought phase: the model leverages visual information for textual reasoning (such as analyzing patterns based on previously added auxiliary lines) and determines the next step by planning what visual hints should be added to enhance understanding. (2) Action phase: the model executes the planned operations by calling external tools (like using `Python` code to draw auxiliary lines or highlight key features) to generate or modify visual information. (3) Observation phase: the model processes the visual feedback from the Action phase, incorporating these new visual hints into the next reasoning step.

The importance of visual information for LVLM reasoning is highlighted in *VisuoThink*, which utilize tool invocations to construct *reliable* visual hints step by step in a visual construction process. This tool-based design allows *VisuoThink* to flexibly adapt to various visual reasoning tasks. Moreover, unlike approaches (e.g. *VisualSketchpad*) that generate all visual aids at once, our step-by-step visual guidance naturally integrates with search techniques, enabling effective test-time scaling.

3.2 Predictive Rollout Search

Based on tree search methods and inspired by MCTS, we propose a predictive rollout search mechanism that interleaves visual-text thinking. By anticipating the outcomes of intermediate states, the model can make timely corrections, enabling more accurate and powerful reasoning. As shown in Figure 2, at each reasoning step, our framework first generates multiple candidate paths through vision-text interleaved thinking, then simulates these paths to predict their outcomes, and finally selects the most promising path through a self-voting mechanism.

Vision-Text Interleaved Expansion In the whole reasoning chain $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_t\}$, given the current node $\mathbf{a}_{t-1}$, the model samples k candidate nodes $\mathbf{S}_t = \{\mathbf{s}_t^1, \mathbf{s}_t^2, ..., \mathbf{s}_t^k\}$. Each candidate follows the vision-text interleaved thinking process described above, generating a sequence of Thought, Action, and Observation steps. This expansion creates a tree of possible reasoning paths, each representing a different problem-solving strategy.

Rollout Simulation Visual reasoning often requires multiple steps to reach a conclusion, making it crucial to evaluate the full potential of each path. For each candidate node $\mathbf{s}_t^i$, the model simulates the complete reasoning process to predict final outcomes $\mathbf{r}_t^i$, rather than relying solely on immediate state evaluation. Different from expansion, the simulation extends each candidate node with a single path of vision-text interleaved thinking until reaching a final result.

Selection The selection of the optimal path is performed through a self-voting mechanism. The model considers the task description, historical nodes, and the simulated path with predicted results for each candidate node. The selection process can be formalized as:

$$\mathbf{Select}(\mathbf{S}_t) = \arg\max_{\mathbf{s}_t^i \in \mathbf{S}_t} \mathbf{Vote}(\mathbf{A}_{t-1}, \mathbf{s}_t^i, \mathbf{r}_t^i) \quad (1)$$

where $\mathbf{A}_{t-1}$ represents the historical context, $\mathbf{s}_t^i$ for the candidate node, and $\mathbf{r}_t^i$ is the predicted result or final state. The **Select** is a heuristic function served by the LVLM model to guide the process. This selection ensures the model pursues the most promising reasoning strategy.

4 Solving Geometry with VisuoThink

The core of our methodology is rooted in multi-step visual information processing and search-based reasoning, enabling LVLMs to address strongly constrained mathematical problems (e.g., geometry challenges) and open-domain scenarios (such as visual navigation and visual tiling in section 5).

We formalize geometry problem-solving as a two-phase process integrating **visual construction** and **algebraic computation**. In Phase I, the model generates auxiliary lines defined by geometric constraints, such as connecting points (x_i, y_i) and (x_j, y_j), construct a perpendicular or parallel line to form line segments $\mathbf{L} = \{l_i\}$. This phase terminates with a AUX-END token, triggering Phase II, where geometric relationships are translated into solvable equations (e.g., $ax + b = 0$) through Python code execution.

Task Formulation LVLM should produce the reasoning trajectory consisting of reasoning steps $\mathbf{A} = \{\mathbf{a}_t\}$ that leads to the final result $\mathbf{r}$, given the original problem $\mathbf{Q}$ while taking into account the auxiliary lines $\mathbf{L}$. The framework operates under a constraint $\sum_{t=1}^{|A|} \|\mathbf{a}_t\| \leq \tau$, where $\mathbf{a}_t$ denotes visual-textual reasoning steps and τ is the maximum step limit:

$$\mathbf{A} \sim \mathcal{P}\left(\{\mathbf{a}_1, \ldots, \mathbf{a}_{|A|}, \mathbf{r}\} \mid \mathbf{Q}, \mathbf{L}\right) \text{ s.t. } \sum_{t=1}^{|\mathbf{A}|} \|\mathbf{a}_i\| \leq \tau \quad (2)$$

This formulation mirrors human problem-solving by decomposing proofs into executable visual-textual steps, validated via coordinate-based tools like matplotlib and equation solver.

Visual Construction We emphasize the criticality of incremental visual information for accurate solutions, where multi-step graphical representations originate from the progressive construction of auxiliary lines. This multi-stage approach facilitates search algorithm-enhanced refinement of auxiliary line generation, significantly improving LVLM capabilities in geometric reasoning. Consistent with Sketchpad methodology, we exclusively utilize common Python libraries (e.g., *matplotlib*) for diagram rendering.

Algebraic Computation Unlike general tasks, solving geometry problems cannot rely solely on visual construction or the model's inherent capabilities; instead, it necessitates the use of computational tools to achieve precise and accurate results. This requirement stems from the need for exact numerical solutions and the mitigation of potential errors in geometric reasoning. Through systematic

	Model	**GPT-4o**	**Qwen2-VL-72B-Instruct**	**Claude-3.5-sonnet**
Geomverse-109	CoT	11.1	5.6	14.4
	VisualSketchpad	8.9	6.7	16.7
	VisualSketchpad + *Equation Solver*	13.3	11.1	17.8
	VisuoThink w/o rollout search (*ours*)	24.4	19.0	26.7
	VisuoThink (*ours*)	**28.9**	**25.6**	**27.8**
Geometry3K (Lu et al., 2021)	CoT	20.8	18.8	37.5
	VisualSketchPad	22.9	17.0	39.6
	VisualSketchpad + *Equation Solver*	25.0	14.9	41.7
	VisuoThink w/o rollout search (*ours*)	27.1	20.8	37.5
	VisuoThink (*ours*)	**33.3**	**25.0**	**43.8**

Table 1: The 1-shot benchmark results (*Accuracy@1*) on Geometry including **Geomverse-109** and **Geometry3k** of SOTA large visual language models. For GPT-4o and Claude-3.5-sonnet, we employ newest cutoffs (*gpt-4o-2024-11-20* and *claude-3-5-sonnet-20241022*) separately. The gray part indicates results from VisuoThink and **bold** results represent the best performance.

Model	Dataset	**Visual Navigation**			**Visual Tiling**
	Subset (Num. Samples)	*level-3 (16)*	*level-4 (31)*	*level-5 (62)*	*level-2 (119)*
GPT-4o	CoT	18.8	3.2	0.0	0.8
	VoT	25.0	0.0	0.0	1.7
	VoT + *Executer*	62.5	9.7	4.8	12.6
	VisuoThink w/o rollout search (*ours*)	81.2	32.3	11.3	19.3
	VisuoThink (*ours*)	**93.8**	**61.3**	**19.4**	**51.2**
Qwen2-VL-72B-Instruct	CoT	6.7	3.2	-	0.0
	VoT	0.0	0.0	-	0.8
	VoT + *Executer*	25.0	3.2	-	6.7
	VisuoThink w/o rollout search (*ours*)	50.0	6.5	-	9.2
	VisuoThink (*ours*)	**81.3**	**12.9**	-	**20.2**
Claude-3.5-sonnet	CoT	37.5	3.2	0.0	0.8
	VoT	56.3	0.0	0.0	2.5
	VoT + *Executer*	68.8	22.6	16.1	10.1
	VisuoThink w/o rollout search (*ours*)	81.2	38.7	41.9	80.7
	VisuoThink (*ours*)	**93.8**	**61.3**	**53.2**	**84.0**

Table 2: The *Pass@1* performance comparison on spatial reasoning benchmarks including **Visual Navigation** and **Visual Tiling** across *SOTA* LVLMs. The gray part indicates results from VisuoThink and **bold** results represent the best performance. The results of Qwen2-VL-72B-Instruct on Visual Navigation (k = 5) are masked out due to its restrained performance on the subset. The results from *VoT* with *Executor* are also reported, where the models utilize the unreliable visual hints generated by themself rather than *executor*, consistent with the *VoT* framework.

integration, like VPD (Zhao et al., 2023), and VisualStechpad (Hu et al., 2024), phase II employs Python code execution for precise computation to mitigate LVLM hallucination risks. Furthermore, the model constructs single-variable algebraic equations based on identified geometric relationships, subsequently invoking equation solvers for numerical resolution.

4.1 Empirical Results

Setup We conduct comprehensive evaluations on the challenging Geometry3K and Geomverse-109 datasets to demonstrate the methodological superiority. Especially we detail the trajectory of Geomverse-109 dataset synthesis in appendix E. SOTA closed-source models including *gpt-4o-2024-11-20* and *claude-3-5-sonnet-20241022* are leveraged for inference. To ensure architectural diversity, open-source model (e.g., *Qwen2-VL-72B*) were incorporated; however, smaller-parameter open-source variants were excluded due to their capability constraints. And we detail the model and algorithm hyperparameters in appendix D.

Analysis Our empirical results reveal that, even without rollout search augmentation, our strategy substantially enhances LVLM reasoning capabilities compared to Chain-of-Thought (CoT) (Mitra et al., 2023) and Visual Sketchpad (Hu et al., 2024) baselines. Notably, on the Geomverse-109 (Kazemi et al., 2023) benchmark, **VisuoThink outperforms CoT and Visual Sketchpad by an average of *17.1%* and *16.7%* across all evaluated models, and predictive rollout search further**

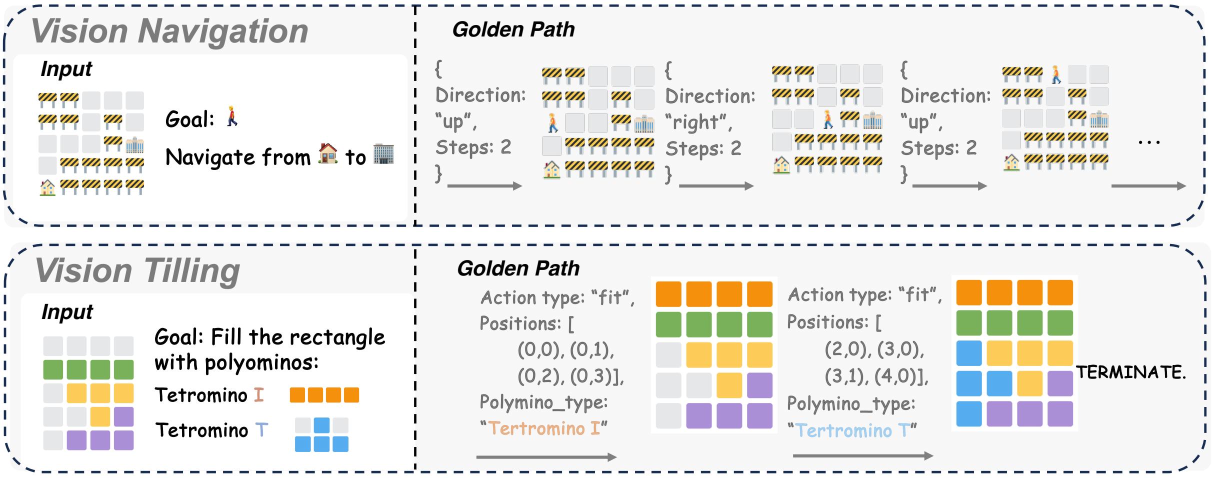
Vision Navigation
Input
Goal: 🚶
Navigate from 🏠 to 🏢
Golden Path
{ Direction: "up", Steps: 2 }
{ Direction: "right", Steps: 2 }
{ Direction: "up", Steps: 2 }
...
Vision Tilling
Input
Goal: Fill the rectangle with polyominos:
Tetromino I
Tetromino T
Golden Path
Action type: "fit", Positions: [(0,0), (0,1), (0,2), (0,3)], Polymino_type: "Tertromino I"
Action type: "fit", Positions: [(2,0), (3,0), (3,1), (4,0)], Polymino_type: "Tertromino T"
TERMINATE.

Figure 3: The illustration of spatial reasoning tasks derived from *VoT* (Wu et al., 2024), including Visual Navigation and Visual Tiling. LVLM is required to execute a sequence of actions to complete certain goals. Our experimental setting makes them much more challenging and closer to real-environment deployment.

enhances models' performance by an average of 4.1%. Also, the employment of *equation solver* on Visual Sketchpad also increases an average performance of *3.3%*. This performance gap likely stems from Geomverse's emphasis on geometric relationship construction, where our equation-solving framework help to accurately get intermediate answers and enables efficient resolution of structurally complex problems. The systematic integration of geometric analysis tools further mitigates error propagation inherent in conventional LVLM reasoning baselines.

5 Spatial Reasoning with VisuoThink

Spatial reasoning, defined as the cognitive capability to interpret spatial object relationships, motion dynamics, and environmental interactions, constitutes a foundational requirement for mission-critical applications such as robotic systems, autonomous navigation, and augmented reality. These domains demand robust integration of visual perception and precise manipulation of spatial-temporal constraints for optimal action planning.

Task Formulation Building upon the Visualization of Thought (*VoT*) (Wu et al., 2024) benchmarks, we design two challenging spatial reasoning benchmarks with enhanced complexity as shown in figure 3: Visual Navigation and Visual Tiling. We provide detailed materials of the differences between the original VoT benchmark setup and our experimental configuration in Appendix B and additionally provide the mathematical task formulation in appendix C.

Visual Construction via *Executor* During task execution, robots deployed in true environments typically receive environmental feedback following each action, which facilitates perception and subsequent decision-making processes. In our methodology, we leverage environmental interaction tools to enhance the model's spatial reasoning capabilities. In each action, we employ an *executor* to implement the corresponding action, and return textual execution feedback and visuospatial hint (*optional*) representing the map state. In the context of (1) Visual Navigation, the visual feedback corresponds to the map including agent's current position; while in (2) Visual Tiling scenarios, it represents the current state of rectangle occupation patterns.

5.1 Empirical Results

Setup We evaluate our framework on two spatial reasoning benchmarks: Visual Navigation and Visual Tiling. For Visual Navigation, we create three difficulty levels with increasing map complexity, where the level indicates the k for Visual Navigation as shown in table 2. For Visual Tiling, we focus on level-2 (i.e. $k = 2$) problems with 119 samples. We compare our method against Chain-of-Thought (*CoT*), Visualization of Thought (*VoT*) (Wu et al., 2024). As table 2 indicates, the results from *VoT* with tool interactions (i.e. *Executor*) are also reported, where textual feedbacks are employed but the visual hints are still generated by the model rather from *executor*, consistent with the *VoT* framework. The source of visual hints distinguishes it from our method. We employ the same

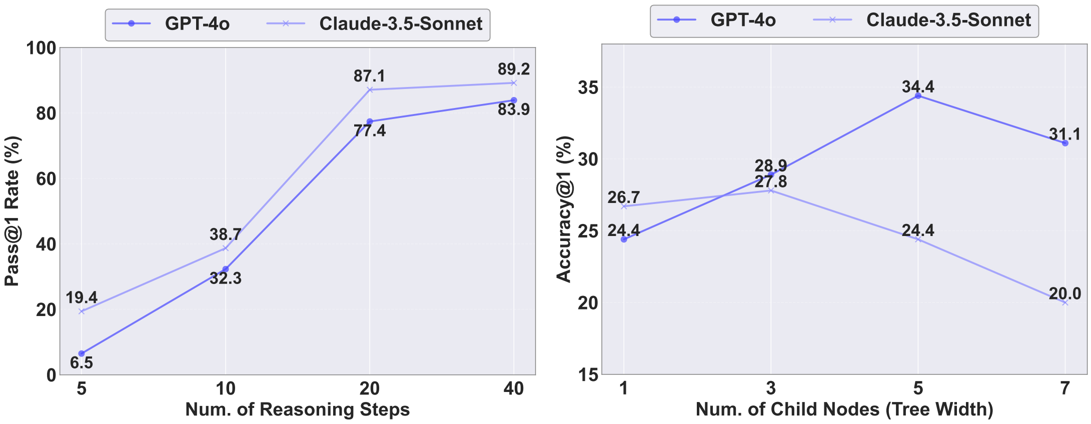

Figure 4: (**LEFT**) The trend of *Pass@1* rate on Visual Navigation as the number of reasoning steps increases. (**RIGHT**) The relationship between the *Accuracy@1* on geometry problems (Geomverse) and tree width for rollout search. **We observe that LVLMs significantly benefit from longer reasoning chains, although the effect plateaus rapidly beyond a certain threshold of reasoning steps. The relationship between performance and tree width exhibits a more complex pattern, demonstrating an inverted U-shaped trend with both *GPT-4o* and *Claude-3.5-Sonnet*.**

temperature and *VisuoThink* hyperparameters as section 4.1.

Analysis In spatial reasoning experiments, *VisuoThink* demonstrates significant performance improvements over baseline methods, particularly when augmented with predictive rollout search. As shown in Table 2, *VisuoThink* achieves the highest accuracy across all tasks, outperforming both *CoT* and *VoT* baselines. For instance, on the Visual Navigation task, *VisuoThink* on GPT-4o achieves a *93.8%* accuracy at level-3, compared to *62.5%* for VoT with an executor and *18.8%* for CoT. This trend is consistent across different model architectures, including *GPT-4o*, *Qwen2-VL-72B-Instruct*, and *Claude-3.5-sonnet*, highlighting the robustness of our approach.

Similar to the geometry experiments in Section 4, the integration of tool interactions and multi-step visual reasoning plays a critical role in enhancing performance. The executor's feedback mechanism, which provides visual updates after each action, mirrors the incremental visual refinement seen in geometry tasks, where auxiliary lines are progressively constructed. For instance, *VisuoThink* without rollout search demonstrates an average improvement of *34.7%* on Visual Tiling across diverse models. We observe that while VoT augmented with textual feedback achieves an average increase of *8.1%*, its performance gain is notably less pronounced compared to *VisuoThink* without rollout search. This underscores the critical role of reliable visual cues in enhancing reasoning capabilities. The dynamic interaction allows the model to iteratively refine its reasoning path, leading to more accurate solutions.

6 Discussion

In this section, we analyze key aspects of *VisuoThink*'s performance. We examine how the length of reasoning chain affects spatial reasoning, the impact of child node expansion in rollout search, and the influence of supervision levels in predictive rollouts across tasks. These insights highlight *VisuoThink*'s effectiveness and suggest future directions for multimodal reasoning frameworks.

6.1 Could Longer Reasoning Chains Assist LVLMs in Reasoning?

In practical applications of *LVLM*s for spatial reasoning tasks, each tool invocation can be seen as an agent attempting an action in the environment and receiving feedback. Although many attempts may be inaccurate, allowing the model more trial-and-error opportunities before achieving the final goal could potentially enhance its reasoning capabilities. By setting different upper limits on the number of reasoning steps in visual navigation tasks, **we observe a positive correlation between the number of reasoning steps and the model's task completion rate. This suggests that the model indeed benefits from more tool invocations and longer reasoning.**

However, as the number of reasoning steps increases, the completion rate gradually converges, making further significant improvements challenging. As shown in figure 4 (*left*), for instance, increasing reasoning steps from 10 to 20 resulted in substantial performance gains (*+54.1%* and *+48.4%*) across different LVLM architectures (GPT-4o and Claude-3.5-sonnet). However, when reasoning steps were increased from 20 to 40, the performance growth slowed dramatically, dropping to *+6.5%* and *+2.1%*, respectively. This phenomenon aligns with expectations, as merely increasing the number of tool invocations does not enable the model to better solve the most challenging samples. This underscores the necessity of techniques like rollout search within the broader context of test scaling.

6.2 Could Larger Tree Span Enhances *VisuoThink*'s Performance?

Predictive rollouts enhance the model's reasoning capabilities, which can be viewed as a tangible outcome of successfully expanding the model's reasoning search space. A natural question arises: Can we further improve the model's reasoning performance on benchmarks simply by increasing the number of candidate child nodes at each selection step, i.e., expanding the *tree width*, thereby enhancing model's reasoning capability? To investigate this, we conducted comparative experiments on geometry tasks using GPT-4o and Claude-3.5-sonnet, keeping the depth of the reasoning tree constant while varying the number of candidate child nodes.

As presented in figure 4 (*right*), we observed an inverted U-shaped trend in overall performance as the number of candidate tree nodes increased across different model architectures. Notably, when the number of candidate child nodes equals 1, the model follows a single reasoning path, effectively bypassing predictive rollout search. Contrary to expectations, the performance trend initially rises and then declines. This counterintuitive result can be attributed to the inherent errors in the model's evaluation of child nodes. **Simply and aggressively increasing the tree width leads to confusion in selecting child nodes, which in turn reduces overall reasoning efficiency.** Thus, an interesting conclusion emerges: we cannot expect to continuously improve model performance by merely increasing the number of child nodes in rollout search.

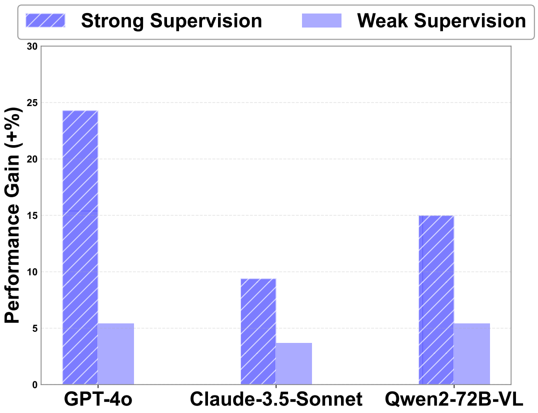

Figure 5: The performance gain (+%) on tasks through predictive rollout search. The performance gain is calculated via the performance gap between *VisuoThink (w/o rollout search)* and *VisuoThink*.

6.3 Strong v.s. Weak Supervision in Predictive Rollout Search

An intriguing observation is that the strength of guidance provided by predictive rollout results varies between geometry and spatial reasoning tasks. In geometry tasks, the model only receives the final numerical results of the problem, whereas in spatial reasoning tasks, the model has access to visual states of stronger supervision (e.g., *the agent's final position, the position of the destination, etc.*). In other word, predictive rollouts in geometry tasks offer weaker supervision, while those in spatial reasoning tasks provide stronger supervision.

This observation aligns with the findings of the Deepseek R1 report, which highlights that outcome-based supervision in RL can significantly enhance Deepseek-R1-Zero's reasoning capabilities (DeepSeek-AI, 2025). **The effectiveness of such supervision stems from its strong supervisory signal, and predictive rollouts with strong supervision are more effective in improving model reasoning performance.** This is further supported by our experimental results, as illustrated in figure 5, where predictive rollouts demonstrated more substantial performance gains in spatial reasoning tasks compared to geometry tasks, across both open-source and closed-source models. The detailed performance gain results are presented in appendix A.

7 Conclusion

We present ***VisuoThink***, a multimodal tree search framework enhancing LVLM reasoning through dynamic visual-textual interleaving and predictive rollout search. Our approach demonstrates significant improvements across geometry and spatial reasoning tasks without requiring model fine-tuning. Empirical results show substantial performance gains on geometry and spatial reasoning benchmarks. Our analysis reveals key insights about tool interaction benefits, search space optimization, and supervision strength in multimodal reasoning. These findings open new possibilities for advancing LVLM capabilities in complex reasoning tasks.

Acknowledgements

Thanks for the ARR reviewers for their diligent work. Dr Tao's research is partially supported by NTU RSR and Start Up Grants. This work was supported by the National Key Research and Development Program of China (No.2022ZD0160102).

Limitations

Despite its strong performance, *VisuoThink* has several limitations. First, the predictive rollout search process introduces significant computational overhead, making it potentially impractical for real-time applications. Second, our approach particularly relies on tool interactions for stronger capability, which may require more effort in some specific deployment environments. Third, the framework's effectiveness is constrained by the quality of the base VLM's reasoning capabilities - while it enhances performance, it cannot overcome fundamental model limitations. Finally, our evaluation focuses primarily on geometric and spatial reasoning tasks.

Ethics and Reproducibility Statements

Ethics We take ethical considerations very seriously and strictly adhere to the ACL Ethics Policy. This paper proposes a test-time slow-thinking framework to improve the multimodal reasoning ability of current LVLMs. All evaluation datasets used in this paper will be publicly available or have been widely adopted by researchers. Thus, we believe that this research will not pose ethical issues.

Reproducibility In this paper, we discuss the detailed experimental setup, such as hyper-parameters, implementation of algorithm, and statistic descriptions. ***More importantly, our code has been open-sourced at*** `https://github.com/ekonwang/VisuoThink` to help reproduce the experimental results of this paper.

References

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Variational best-of-n alignment. *ArXiv*, abs/2407.06057.

Jinhe Bi, Yifan Wang, Danqi Yan, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. 2025a. Prism: Self-pruning intrinsic selection method for training-free multimodal data selection. *Preprint*, arXiv:2502.12119.

Jinhe Bi, Yujun Wang, Haokun Chen, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. 2025b. Llava steering: Visual instruction tuning with 500x fewer parameters through modality linear representation-steering. *Preprint*, arXiv:2412.12359.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024. Alphamath almost zero: process supervision without process. *ArXiv*, abs/2405.03553.

Anoop Cherian, Kuan-Chuan Peng, Suhas Lohit, Joanna Matthiesen, Kevin Smith, and Joshua B Tenenbaum. 2024. Evaluating large vision-and-language models on children's mathematical olympiads. *arXiv preprint arXiv:2406.15736*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yifan Du, Zikang Liu, Yifan Li, Wayne Xin Zhao, Yuqi Huo, Bingning Wang, Weipeng Chen, Zheng Liu, Zhongyuan Wang, and Jiahui Wen. 2025. Virgo: A preliminary exploration on reproducing o1-like mllm.

Xidong Feng, Ziyu Wan, Muning Wen, Ying Wen, Weinan Zhang, and Jun Wang. 2023. Alphazero-like tree-search can guide large language model decoding and training. *ArXiv*, abs/2309.17179.

Timin Gao, Peixian Chen, Mengdan Zhang, Chaoyou Fu, Yunhang Shen, Yan Zhang, Shengchuan Zhang, Xiawu Zheng, Xing Sun, Liujuan Cao, and Rongrong Ji. 2024. Cantor: Inspiring multimodal chain-of-thought of mllm. *ArXiv*, abs/2404.16033.

Lin Gui, Cristina Garbacea, and Victor Veitch. 2024. Bonbon alignment for large language models and the sweetness of best-of-n sampling. *ArXiv*, abs/2406.00832.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke S. Zettlemoyer, Noah A. Smith, and Ranjay Krishna. 2024. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *ArXiv*, abs/2406.09403.

Daniel Kahneman. 2011. Thinking, fast and slow. *Farrar, Straus and Giroux*.

Mehran Kazemi, Hamidreza Alvari, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. 2023. Geomverse: A systematic evaluation of large models for geometric reasoning. *Preprint*, arXiv:2312.12241.

Xuanyu Lei, Zonghan Yang, Xinrui Chen, Peng Li, and Yang Liu. 2024. Scaffolding coordinates to promote vision-language coordination in large multi-modal models. In *International Conference on Computational Linguistics*.

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

Jiacheng Liu, Andrew Cohen, Ramakanth Pasunuru, Yejin Choi, Hannaneh Hajishirzi, and Asli Celikyilmaz. 2023. Don't throw away your value model! generating more preferable text with value-guided monte-carlo tree search decoding.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *Preprint*, arXiv:2105.04165.

Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. 2023. Compositional chain-of-thought prompting for large multimodal models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14420–14431.

Debjyoti Mondal, Suraj Modi, Subhadarshi Panda, Rituraj Singh, and Godawari Sudhakar Rao. 2024. Kam-cot: Knowledge augmented multimodal chain-of-thoughts reasoning. In *AAAI Conference on Artificial Intelligence*.

OpenAI. 2024a. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2024b. Learning to reason with llms.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, and 1 others. 2024. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*.

Santhosh Kumar Ramakrishnan, Erik Wijmans, Philipp Kraehenbuehl, and Vladlen Koltun. 2024. Does spatial cognition emerge in frontier models? *Preprint*, arXiv:2410.06468.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *ArXiv*, abs/2408.03314.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024. Fast best-of-n decoding via speculative rejection. *ArXiv*, abs/2410.20290.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, Hisham Cholakkal, Ivan Laptev, Mubarak Shah, Fahad Shahbaz Khan, and Salman H. Khan. 2025. Llamav-o1: Rethinking step-by-step visual reasoning in llms.

Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024. Mind's eye of llms: Visualization-of-thought elicits spatial reasoning in large language models.

Kun Xiang, Zhili Liu, Zihao Jiang, Yunshuang Nie, Runhui Huang, Haoxiang Fan, Hanhui Li, Weiran Huang, Yihan Zeng, Jianhua Han, Lanqing Hong, Hang Xu, and Xiaodan Liang. 2024. Atomthink: A slow thinking framework for multimodal mathematical reasoning. *Preprint*, arXiv:2411.11930.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, MingSung Kan, Junxian He, and Qizhe Xie. 2023. Self-evaluation guided beam search for reasoning. In *Neural Information Processing Systems*.

Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. 2024. Llava-cot: Let vision language models reason step-by-step. *ArXiv*, abs/2411.10440.

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

Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, and Dacheng Tao. 2024. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. *ArXiv*, abs/2412.18319.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Fei Yu, Anningzhe Gao, and Benyou Wang. 2023. Ovm, outcome-supervised value models for planning in mathematical reasoning. In *NAACL-HLT*.

Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Bo Wang, Shimin Li, Yunhua Zhou, Qipeng Guo, Xuanjing Huang, and Xipeng Qiu. 2024. Scaling of search and learning: A roadmap to reproduce o1 from reinforcement learning perspective. *Preprint*, arXiv:2412.14135.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alexander J. Smola. 2023. Multimodal chain-of-thought reasoning in language models. *Trans. Mach. Learn. Res.*, 2024.

Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. 2023. Unleashing text-to-image diffusion models for visual perception. *Preprint*, arXiv:2303.02153.

Qiji Zhou, Ruochen Zhou, Zike Hu, Panzhong Lu, Siyang Gao, and Yue Zhang. 2024. Image-of-thought prompting for visual reasoning refinement in multimodal large language models. *ArXiv*, abs/2405.13872.

A Performance Gain of *VisuoThink* Through Predictive Rollout Search

This appendix quantifies the performance improvements achieved by integrating predictive rollout search into the *VisuoThink* framework across geometry and spatial reasoning tasks. The performance gain through predictive rollout search is derived by subtracting the performance of *VisuoThink (w/o rollout search)* from those of the *VisuoThink* on models.

As shown in Table 3, tasks with strong supervision (e.g., **Visual Navigation** and **Visual Tiling**) exhibit significantly higher gains compared to weak supervision tasks (e.g., *Geometry3K* and *Geomverse-109*). For instance, under strong supervision, Claude-3.5-Sonnet achieves a *+25.1%* improvement in Visual Navigation, while GPT-4o attains *+16.6%* in Visual Tiling. In contrast, weak supervision tasks like **Geomverse-109** only show modest gains (e.g., *+5.4%* for GPT-4o).

B Spatial Reasoning Task Setting

Our formulation extends beyond *VoT*'s basic requirements by mandating LVLMs to generate comprehensive operational specifications - for instance, requiring explicit output of both movement directions and precise step counts at each decision node. This advancement creates more realistic and functionally grounded spatial reasoning evaluations (e.g., *robotic navigation emulation in real world*).

This appendix details the task formulation differences between *VisuoThink* and baseline methods (Table 4 and Table 5). For **Visual Navigation**, *VisuoThink* requires fine-grained, executable and explicit specification of both direction and step count in action sequences, whereas VoT focuses solely on direction navigation. This formulation mirrors real-world robotic navigation, where precise movement planning is critical. Similarly, in **Visual Tiling**, *VisuoThink* mandates detailed actions, including polyomino variant types, block positions, and action types (e.g., "fit" or "remove"), while *VoT* simplifies the task by omitting variant specifications.

C Task Formulation of Spatial Reasoning Tasks

Building upon *VoT* (Wu et al., 2024) framework, our challenging benchmarks comprise:

- **Visual Navigation** evaluates LVLMs in a simulated 2D grid environment, where agents must navigate from initial position $\mathbf{s}_0$ to destination $\mathbf{s}_k$ through obstacle-laden paths. The formal problem is defined by grid map $\mathbf{M}$ containing k interconnected edges $\mathbf{E} = \{\mathbf{e}(\mathbf{s}_0, \mathbf{s}_1), \mathbf{e}(\mathbf{s}_1, \mathbf{s}_2), \ldots, \mathbf{e}(\mathbf{s}_{k-1}, \mathbf{s}_k)\}$. The LVLM should generate a sequence of executable actions in `json` format $\mathbf{A} = \{(\mathbf{d}_0, \mathbf{l}_0), (\mathbf{d}_1, \mathbf{l}_1), \ldots, (\mathbf{d}_{|\mathbf{A}|-1}, \mathbf{l}_{|\mathbf{A}|-1})\}$, where each tuple specifies movement direction $\mathbf{d}_i$ and exact step count $\mathbf{l}_i$, governed by the policy:

$$\mathbf{a_t} \sim \mathcal{P}\left(\mathbf{d}_t, \mathbf{l}_t \mid \mathbf{A}_{t-1}, \mathbf{M}\right) \tag{3}$$

- **Visual Tiling** is a classic geometric reasoning challenge, this task assesses polyomino composition capabilities within confined rectangular regions $\mathbf{R}$ masked by k distinct polyominoes $\mathbf{MP} = \{\mathbf{mp}_1, \ldots, \mathbf{mp}_k\}$. The LVLM must output action sequences $\mathbf{a}_t = (\mathbf{p}_t, \{\mathbf{b}_1, \ldots, \mathbf{b}_{|B|}\}, \mathbf{at}_t)$, where $\mathbf{p}_t$ and $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_{|\mathbf{B}|}\}$ respectively indicate the selected polyomino type and the coordinates of the placement blocks. $\mathbf{at}_t \in \{\textit{fit}, \textit{remove}\}$ indicates the action type modifying rectangular state $\mathbf{R}_t$, thus formalized as:

$$\mathbf{a}_t \sim \mathcal{P}\left(\mathbf{p}_t, \mathbf{B}, \mathbf{at}_t \mid \mathbf{R}_{t-1}, \mathbf{MP}, \mathbf{A}_{t-1}\}\right) \tag{4}$$

Though the required actions are polyomino variant-aware as shown in table 5. As the polyomino variant type is implicitly expressed in the block positions, LVLM does not need to explicitly output it in actions anymore.

Supervision Type	Performance Gain	*GPT-4o*	*Qwen2-VL-72B*	*Claude-3.5-Sonnet*
Strong Supervision	Δ Visual Navigation (%)	+16.6	+18.9	+15.5
	Δ Visual Tiling (%)	+31.9	+11.0	+3.3
	Δ Average (%)	+24.3	+15.0	+9.4
Weak Supervision	Δ Geometry3K (%)	+4.5	+6.6	+1.1
	Δ Geomverse-109 (%)	+6.2	+4.2	+6.3
	Δ Average (%)	+5.4	+5.4	+3.7

Table 3: Detailed performance gain of ***VisuoThink*** through predictive rollout search on benchmarks from Geometry and Spatial Reasoning over variable *LVLM* models.

	Method	Direction	Steps	**Target**
Visual Navigation	*VoT*	✓	✗	Navigate from the starting position to the destination.
	VisuoThink	✓	✓	

Table 4: **Visual Navigation** task setting differences between *VoT* and ***VisuoThink***.

	Method	**Action**				**Target**
		Polyomino Type	Variant Type	Block Positions	Action Type	
Visual Tiling	*VoT*	✓	✓	✗	✗	To identify the correct variant for a polyomino in one action.
	VisuoThink	✓	✓	✓	✓	To fill the rectangle with feasible polyomino variants.

Table 5: **Visual Tiling** task setting differences between *VoT* and ***VisuoThink***.

D Model and *VisuoThink* Hyperparameters

We detail the model and *VisuoThink* Hyperparameters:

Model Hyperparameters To ensure experimental fairness, we uniformly constrained the number of reasoning steps (i.e., τ, *the depth of the reasoning tree*) to *10* across all experiments. During predictive rollout search, we set the number of sampled child nodes to *3*, and we discuss its impact in section 6.2.

***VisuoThink* Hyperparameters** While *VisuoThink* employed a temperature of *0.8* when sampling child nodes, all other model invocations, including the baselines (e.g. *CoT*, *VoT*, *VisualSketchpad*, *VisuoThink* w/o rollout search), were conducted with temperature set to *0* for frontier performance. During the voting phase, we similarly maintained a temperature of *0* and implemented single-vote sampling, which not only reduced computational overhead in terms of model calls but also achieved comparable performance.

E Geomverse-109 Problem Generation Trajectory

We establish a pipeline translating textual problems into problems with matplotlib-executable code. Beyond the **Geometry3K** (Lu et al., 2021) dataset (*48 problems*) utilized in Sketchpad, we incorporate the D2 subset of Geomverse (Kazemi et al., 2023) to construct an slightly bigger dataset **Geomverse-109** (*90 problems*). The original Geomverse dataset crucially includes annotated point coordinates essential for systematic problem synthesis. During the data synthesis phase, we first randomly choose 109 problems, then LVLMs generate corresponding high-quality Python code through LLM self-reflection (Shinn et al., 2023), then we filter out problems with poor diagram quality.