Visual Planning: Let's Think Only with Images

Yi Xu[2*] Chengzu Li[1*] Han Zhou[1*] Xingchen Wan[3] Caiqi Zhang[1]
Anna Korhonen[1] Ivan Vulić[1]
[1]University of Cambridge [2]University College London [3]Google
y.xu.23@ucl.ac.uk {cl917, hz416, cz391, alk23, iv250}@cam.ac.uk

Abstract

*Recent advancements in Large Language Models (LLMs) and their multimodal extensions (MLLMs) have substantially enhanced machine reasoning across diverse tasks. However, these models predominantly rely on pure text as the medium for both expressing and structuring reasoning, even when visual information is present. In this work, we argue that language may not always be the most natural or effective modality for reasoning, particularly in tasks involving spatial and geometrical information. Motivated by this, we propose a new paradigm, Visual Planning, which enables planning through purely visual representations, independent of text. In this paradigm, planning is executed via sequences of images that encode step-by-step inference in the visual domain, akin to how humans sketch or visualize future actions. We introduce a novel reinforcement learning framework, Visual Planning via Reinforcement Learning (*VPRL*), empowered by GRPO for post-training large vision models, leading to substantial improvements in planning in a selection of representative visual navigation tasks,* FROZENLAKE, MAZE, *and* MINIBEHAVIOR. *Our visual planning paradigm outperforms all other planning variants that conduct reasoning in the text-only space. Our results establish Visual Planning as a viable and promising alternative to language-based reasoning, opening new avenues for tasks that benefit from intuitive, image-based inference.*

1. Introduction

Large Language Models (LLMs) [2, 6, 38] have demonstrated strong capabilities in language understanding and generation, as well as growing competence in complex reasoning, enabled by their chain-of-thought reasoning abilities [50]. Building on these advances, recent work extends LLMs to support multiple modalities, yielding so-called Multimodal Large Language Models (MLLMs) [22, 42]: they incorporate visual embedded information at the input to tackle a broader spectrum of tasks, such as visual spatial reasoning [30, 33] and navigation [15, 29]. However, despite their multimodal inputs, these methods perform reasoning purely in the text format in inference, from captioning visual content [19] to generating verbal rationales [59].

Building on this observation, we argue that performing multimodal reasoning only in the text pathway may not always offer the most intuitive or effective strategy, particularly for tasks that depend heavily on visual information and/or are 'vision-first' by design. Indeed, recent results from multimodal benchmarks [7, 8, 30, 43] offer growing evidence that purely language-based reasoning falls short in certain domains, particularly those involving spatial, geometric, or physical dynamics [56]. Such reliance on grounding visual information into text before reasoning introduces a modality gap that hinders the model's ability to capture visual features and state transitions. This highlights a potential shortcoming of current MLLMs: while they process image inputs, they do not naturally "*think*" in images. For instance, tasks such as planning a route through a maze, designing the layout of a room, or predicting the next state of a mechanical system are often better served by visual representations, as verbal descriptions may struggle to accurately capture complex spatial reasoning relationships. These examples suggest a broader question, which we aim to tackle in this work: *can models plan in non-verbal modalities, such as images, without being mediated by text?*

Cognitive science also offers compelling motivation for this question [36]. Dual Coding Theory [39] proposes that human cognition operates through both verbal and nonverbal channels, each capable of independent representational and inferential processes. Recent work on MLLMs incorporates interleaved text and images as reasoning steps [21, 31]. However, they still remain fundamentally text-driven and rely on tool-based visualizations as auxiliary information for reasoning traces, with reasoning still mainly embedded in verbal traces. For instance, Visual Sketchpad [21] employs external tools to generate sketches as visual aids, and MVoT [31] generates per-step visualizations from

*Equal contribution.

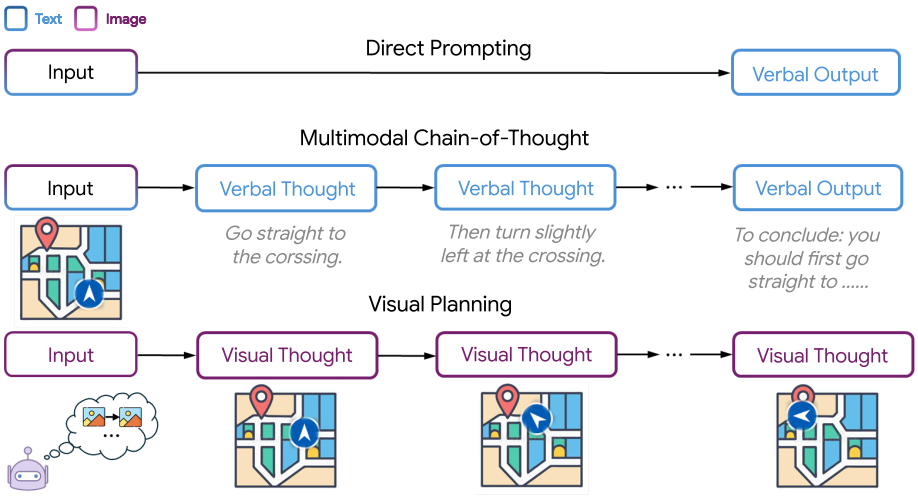

Figure 1. Comparison of reasoning paradigms. The traditional approaches (*top* and *middle* rows) generate verbose and inaccurate textual plan, while the Visual Planning paradigm (*bottom* row) predicts the next visual state directly, forming a pure image trajectory.

language-based actions but still reasons in text for decision-making. As such, a truly visual-only reasoning paradigm that avoids any text-based reasoning proxies remains under-explored.

In this work, we propose a new paradigm, *Visual Planning*, where reasoning is structured as a sequence of images, but without the mediation of language. To the best of our knowledge, this is the first attempt to investigate whether models can achieve planning purely through visual representations. Rather than generating textual rationales and answers, our approach produces step-by-step visualizations that encode planning or inference steps directly in images. As a pioneering exploration, it circumvents the modality mismatch that occurs when visual problems must be forced into explanations in verbal form, reinforces state transitions, and provides a new trackable interface for tasks like navigation [30], and visual problem-solving [19].

Specifically, we explore this paradigm using the Large Vision Model (LVM) [4] trained exclusively on images and video frames with **no** textual data. This design choice removes potential confounders introduced by language-based supervision and enables a clean investigation of whether models can reason purely within the visual modality. Motivated by the success of reinforcement learning in acquiring reasoning capabilities within the language modality [16] and its strong generalization performance [11], we propose Visual Planning via Reinforcement Learning (VPRL), a novel two-stage reinforcement learning framework empowered by GRPO [44] for visual planning. It involves a distinct initializing stage for encouraging the exploration of the policy model in the given environment, which is then followed by reinforcement learning with a progress reward function.

We validate the feasibility of our paradigms on grid-based navigation as a representative of spatial planning tasks, including MAZE [23], FROZENLAKE [53], and MINIBEHAVIOR [25], where one agent is requested to navigate to a target location successfully without violating environment constraints. Our experiments reveal that the visual planning paradigm substantially surpasses the traditional textual reasoning method by supervised fine-tuning (SFT), achieving more than 40% higher average exact-match rate. In addition to better performance, our novel method VPRL exhibits stronger generalization to out-of-distribution scenarios than the SFT method in the visual planning paradigm (VPFT). To the best of our knowledge, we are the first to apply RL to image generation in the context of planning; the main contributions comprise the following:

- We propose a new reasoning paradigm, *Visual Planning*, and validate the feasibility of visual reasoning without any use of text and language for reasoning.
- We introduce VPRL, a novel two-stage training framework that applies RL to achieve visual planning via sequential image generation.
- We demonstrate empirically that VPRL significantly outperforms the traditional textual reasoning paradigm and supervised baselines in visual spatial planning settings, achieving substantial gains in task performance and exhibiting improved generalization.

2. Visual Planning via Reinforcement Learning

2.1. The Visual Planning Paradigm

The majority of prior visual reasoning benchmarks [1, 14, 55] can be and is typically tackled by grounding the visual information in the textual domain [18, 40, 57], followed by a few steps of textual reasoning. However, once the visual content is mapped to text (e.g., object names, attributes, or relations), the problem gets reduced to a language reason-

ing task, where the reasoning is carried out by the language model, even without reflecting any information from the visual modality during the reasoning process.

Our visual planning paradigm is fundamentally different. It performs planning purely within the visual modality. We formally define visual planning as a process of generating a sequence of intermediate images $\mathcal{T} = (\hat{v}_1, \ldots, \hat{v}_n)$, where each $\hat{v}_i$ represents a visual state that together constitute a visual planing trajectory, given the input image v_0. Specifically, let π_θ denote a generative vision model parameterized by θ. The visual planning trajectory $\mathcal{T}$ is generated autoregressively, where each intermediate visual state $\hat{v}_i$ is sampled conditioned on the initial state and previously generated states:

$$\hat{v}_i \sim \pi_\theta(v_i | v_0, \hat{v}_1, ..., \hat{v}_{i-1}) \tag{1}$$

2.2. Reinforcement Learning for LVM

Reinforcement learning (RL) has shown notable advantages in improving the generalization of autoregressive models by optimizing with *sequence-level* rewards beyond token-level supervision signals [11]. In autoregressive image generation, an image is represented as a *sequence of visual tokens*. Inspired by the success of RL in language reasoning [16], we introduce an RL-based training framework for visual planning empowered by Group Relative Policy Optimization (GRPO) [44]. It leverages the transitions between visual states to compute the reward signals while verifying the constraints from the environments. To enforce the policy model that generates valid actions with diverse exploration during the RL process, we then propose a novel two-stage reinforcement learning framework for visual planning. In Stage 1, we first apply supervised learning to initialize the policy model with random walks. Model's visual planning is then optimized by the RL training in Stage 2.

Stage 1: Policy Initialization. In this stage, we initialize the model π_θ by training it on random trajectories obtained by random walks in the environment. The goal here is to generate valid sequences of visual states and retain exploration capability in a 'simulated' environment. For training, each trajectory consists of a sequence of visual states $(v_0, \ldots, v_n)$. From each trajectory, we extract $n-1$ image pairs of the form $(v_{\leq i}, v_{i+1})$, where $v_{\leq i}$ represents the prefix sequence $(v_0, \ldots, v_i)$. Subsequently, given an input prefix, the model is exposed to a set of plausible next states $\{v_{i+1}^{(j)}\}_{j=1}^{K}$, collected from K valid trajectories that share the same prefix. To prevent overfitting to the specific transition and encourage stochasticity, we randomly sample one candidate $v_{i+1}^{(\ell)}$ from this set at each training step as the supervision target by minimizing the following loss function of visual planning via fine-tuning (VPFT):

$$\mathcal{L}_{\text{VPFT}}(\theta) = -\mathbb{E}_{(v_{\leq i},\, v_{i+1}^{(\ell)})} \Big[\log \pi_\theta\big(v_{i+1}^{(\ell)} \,\big|\, v_{\leq i}\big) \Big]. \tag{2}$$

Overall, the first stage serves as a warm-up for subsequent optimization, focusing on producing visually coherent outputs and enhancing the generation quality.

Stage 2: Reinforcement Learning for Visual Planning. Building on Stage 1, where the model is initialized with random trajectories, it acquires the effective exploration capability. This property is essential for RL, as it ensures coverage over all possible transitions and prevents collapse to suboptimal behaviors. Stage 2 then leverages this ability to simulate the outcomes of potential actions by generating the next visual state and guiding the model to effectively do the planning. During this stage, the RL algorithm provides feedback and rewards based on the correctness of the simulated actions, gradually enabling the model to learn effective visual planning. Specifically, given an input prefix $v_{\leq i}$, the behavior model $\pi_{\theta_{\text{old}}}$ samples a group of G candidate responses $\{\hat{v}_{i+1}^{(1)}, \ldots, \hat{v}_{i+1}^{(G)}\}$. Each response represents a hypothetical visual state corresponding to a planned action $a_i^{(k)}$ at time step i. To interpret these transitions, we employ a rule-based parsing function that maps pairs of visual states $(v_i, \hat{v}_{i+1}^{(k)})$ to discrete actions. The candidate response is then scored using a composite reward function $r(v_i, \hat{v}_{i+1}^{(k)})$, which quantifies whether the generated visual state represents meaningful progress toward the goal state. The reward design is described in detail in the next paragraph.

Instead of relying on a learned critic to estimate value functions, which may introduce additional sources of uncertainty and complexity, GRPO provides more computationally efficient and interpretable training signals by computing relative advantages through comparisons within the group. In this case, the relative advantage of each candidate is $A^{(k)} = \frac{r^{(k)} - \text{mean}\{r^{(1)}, r^{(2)}, \ldots, r^{(G)}\}}{\text{std}\{r^{(1)}, r^{(2)}, \ldots, r^{(G)}\}}$.

To guide the model toward producing responses with higher advantages, we update the policy π_θ by maximizing the following objective:

$$\begin{aligned} \mathcal{J}_{\text{VPRL}}(\theta) = \mathbb{E}_{v_{\leq i} \sim \mathcal{D},\, \{\hat{v}_{i+1}^{(k)}\}_{k=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot | v_{\leq i})} \\ \Bigg[\frac{1}{G} \sum_{i=1}^{G} \min(\rho^{(k)} A^{(k)},\ \text{clip}(\rho^{(k)}, 1-\epsilon,\ 1+\epsilon) A^{(k)}) \\ - \beta\, D_{\text{KL}}(\pi_\theta \,||\, \pi_{\text{ref}}) \Bigg], \end{aligned} \tag{3}$$

where $\mathcal{D}$ is the prefix distribution and $\rho^{(k)} = \frac{\pi_\theta(\hat{v}_{i+1}^{(k)} | v_{\leq i})}{\pi_{\theta_{\text{old}}}(\hat{v}_{i+1}^{(k)} | v_{\leq i})}$ is the importance sampling ratio.

Reward Design. Unlike discrete actions or text tokens, visual outputs are sparse, high-dimensional, and not easily decomposable into interpretable units. In our visual planning framework, the challenge is even more specific: whether the generated visual state can correctly reflect the intended

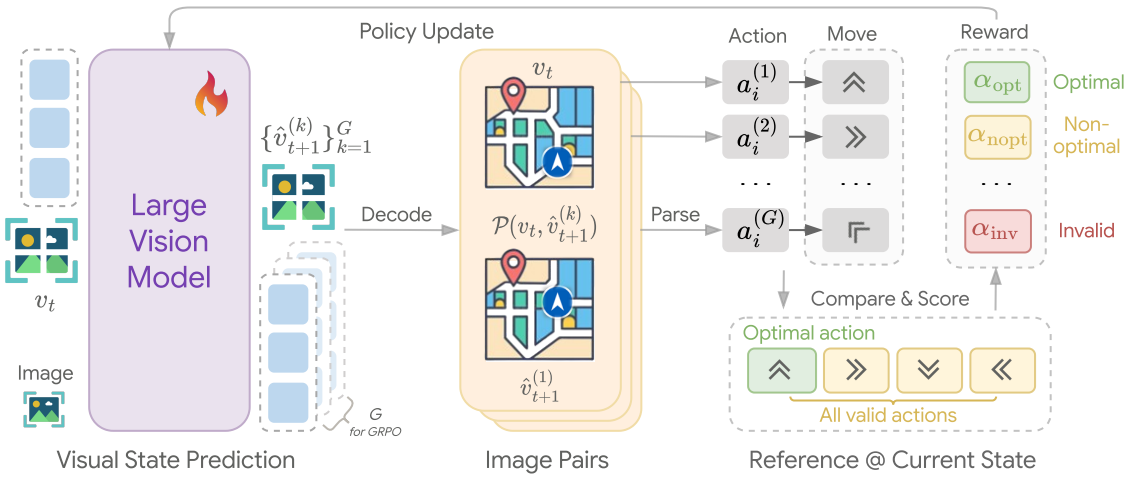

Figure 2. An overview of the proposed VPRL framework, illustrated with autoregressive large vision models for image generation in the context of a visual navigation task. We train the visual policy model with GRPO, using the *progress* reward that encourages progressing actions and penalizes invalid actions, yielding goal-aligned visual planning.

planning action. Therefore, the reward design focus on the progress toward the goal while validating the actions with constraints. To interpret the intended action that connects the current state v_i to a generated candidate state $\hat{v}_{i+t}^{(k)}$, we define a *state-action parsing function* $\mathcal{P} : \mathcal{V} \times \mathcal{V} \to \mathcal{A} \cup \mathcal{E}$, where $\mathcal{A}$ denotes the set of *valid* actions, and $\mathcal{E}$ is the set of *invalid* transitions, such as a violation of physical constraints of the environment. Formally,

$$\mathcal{P}(v_i, \hat{v}_{i+1}^{(k)}) = \begin{cases} a_i^{(k)}, & \text{if } a_i^{(k)} \in \mathcal{A}, \\ e_i^{(k)}, & \text{if } e_i^{(k)} \in \mathcal{E}. \end{cases} \tag{4}$$

It helps to interpret model behaviors from pixel data to intended action through either standalone segmentation components [41] or rule-based scripts. Once having the intended actions, to systematically evaluate action effectiveness, we introduce the *progress map* $D(v) \in \mathbb{N}$ that estimates the remaining steps or effort required to reach the goal from each visual state. By comparing the agent's current and resulting state against the progress map, we partition $\mathcal{A} \cup \mathcal{E}$ into three disjoint subsets:

$$\mathcal{A}_{\text{opt}} = \{a \in \mathcal{A} : D(\hat{v}_{i+1}^{(k)}) < D(v_i)\},$$
$$\mathcal{A}_{\text{nopt}} = \{a \in \mathcal{A} : D(\hat{v}_{i+1}^{(k)}) \geq D(v_i)\}, \quad \mathcal{E}_{\text{inv}} = \mathcal{E}.$$

We then propose the *progress* reward function $r(v_i, \hat{v}_{i+1}^{(k)})$ as:

$$\underbrace{\alpha_{\text{o}} \cdot \mathbb{I}[\mathcal{P}(v_i, \hat{v}_{i+1}^{(k)}) \in \mathcal{A}_{\text{opt}}]}_{\text{optimal}} + \underbrace{\alpha_{\text{n}} \cdot \mathbb{I}[\mathcal{P}(v_i, \hat{v}_{i+1}^{(k)}) \in \mathcal{A}_{\text{nopt}}]}_{\text{non-optimal}}$$
$$+ \underbrace{\alpha_{\text{i}} \cdot \mathbb{I}[\mathcal{P}(v_i, \hat{v}_{i+1}^{(k)}) \in \mathcal{E}_{\text{inv}}]}_{\text{invalid}}, \tag{5}$$

where $\alpha_{\text{o}}, \alpha_{\text{n}}, \alpha_{\text{i}}$ are reward coefficients. In our experiments, we set $\alpha_{\text{o}} = 1$, $\alpha_{\text{n}} = 0$, and $\alpha_{\text{i}} = -5$, thereby rewarding progressing actions, assigning zero to non-progressing actions, and heavily penalizing invalid transitions.

2.3. System Variants

In addition to VPRL, we include several training system variants as baselines that differ in supervision modalities (language vs. image) and optimization methods (SFT vs. RL), allowing us to compare language-based and vision-based planning while assessing the role of reinforcement learning.

Visual Planning via Fine-Tuning (VPFT). We propose Visual Planning via Fine-Tuning (VPFT) as a simplified variant of our framework, which shares the same training architecture as Stage 1 in Section 2.2, but replaces random trajectories with optimal planning trajectories. For each environment, we sample a distinct trajectory $(v_0^{\text{opt}}, v_1^{\text{opt}}, \ldots, v_n^{\text{opt}})$ representing the minimal-step path from the initial state $v_0^{\text{opt}} = v_0$ to the goal. At each step, the model is trained to predict the next state v_{i+1}^{opt} given the prefix $v_{\leq i}^{\text{opt}}$. The objective is identical to Equation 2, with supervision from the optimal trajectory.

Supervised Fine-Tuning (SFT) in Text. In this baseline, planning is formulated in the language modality. Instead of generating an intermediate visual consequence of an action, the model produces a textual description of the intended action sequence. Formally, given an visual input state v and a textual prompt p, which represents the task description, the model is trained to generate a verbalized action sequence $t = (t_1, \ldots, t_L)$, where each token $t_i \in \mathcal{V}_{\text{text}}$ represents an action. The input to the model is the concatenation of the

Model	Input	Output	FROZENLAKE		MAZE		MINIBEHAVIOR		AVG.	
			EM (%)	PR (%)	EM (%)	PR (%)	EM (%)	PR (%)	EM (%)	PR (%)
Closed-Source Model										
Gemini 2.0 Flash										
- Direct	A+🖼	A	21.2	47.6	8.3	31.4	0.7	29.8	10.1	36.3
- CoT	A+🖼	A	27.6	52.5	6.9	29.8	4.0	31.2	12.8	37.8
Gemini 2.5 Pro (*think*)	A+🖼	A	72.0	85.0	21.5	35.5	37.6	59.9	43.7	60.1
Open-Source Model										
Qwen 2.5-VL-Instruct-3B										
- Direct	A+🖼	A	0.9	14.4	0.5	13.6	0.0	10.0	0.5	12.7
- CoT	A+🖼	A	1.3	13.4	0.8	8.2	1.2	12.5	1.1	11.4
- SFT†	A+🖼	A	59.0	76.3	33.3	52.7	10.6	31.0	34.3	53.3
LVM-3B										
- VPFT† (ours)	🖼	🖼	75.4	79.5	59.0	64.0	33.8	52.2	56.1	65.2
- **VPRL**† (ours)	🖼	🖼	**91.6**	**93.2**	**74.5**	**77.6**	**75.8**	**83.8**	**80.6**	**84.9**

Table 1. Performance of the closed- and open-source models on FROZENLAKE, MAZE, and MINIBEHAVIOR. VPRL performs consistently the best (**bold**) across all tasks. † denotes the post-trained model. A represents texts and 🖼 represents images. The last column AVG. reports the average performance across three tasks.

prompt tokens and the visual tokens, and the target is the corresponding action sequence. Following prior work on supervised fine-tuning (SFT) [49] in autoregressive models, we minimize the cross-entropy loss for action prediction:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(v,t)}\left[\sum_{i=1}^{L} \log \pi_\theta(t_i \mid t_{<i}, v, p)\right]. \quad (6)$$

3. Experiments and Results

Tasks To evaluate our proposed visual planning paradigm, we select representative tasks where planning can be expressed and executed entirely in the visual modality. We focus on tasks where state transitions are visually observable, distinguishing them from language-centric tasks like code generation [27] or traditional visual question answering. This design allows us to analyze planning behavior without relying on textual rationales or symbolic outputs. To compare visual planning with language-based reasoning, we experiment with 3 visual navigation environments: FROZENLAKE [53], MAZE [23], and MINIBEHAVIOR [25]. All of them can be solved in both modalities, which enables a direct comparison between visual planning and language reasoning strategies.

- FROZENLAKE: It is initially proposed by Wu et al. [53] and implemented with Gym [5]. It simulates a grid-based frozen lake, where the agent is supposed to start from the designated position and find its way to the destination safely without falling into the 'holes'.
- MAZE: Given an initial image of the maze layout, the model is supposed to go through the maze from the starting point (green point) to the destination (red flag).
- MINIBEHAVIOR: The agent is required to reach the printer and pick it up. After that, the agent should go to the table and drop the printer. It consists of 2 additional actions, including 'pick' and 'drop'.

We construct synthetic datasets for the tasks with varying levels of complexity in patterns and environments. Details on data collection and implementation are in App. B.1.

Models To explore visual planning without any language influence as confounders and enables a clean investigation, we select models trained exclusively on visual data without any exposure to textual data during pretraining. For our methods (VPFT and VPRL), we use the Large Vision Model (LVM-3B) [4] as the backbone, which is only trained on image sequences and videos. For RL training, we design and provide the detailed implementation of rule-based state-action parsing function $\mathcal{P}$ and progress map $D(v)$ in App. B.3. We also include textual planning baselines for parallel comparison, where planning is formulated through language, typically as a textual sequence of actions. Specifically, we evaluate Qwen 2.5-VL-Instruct [3], matched in size to LVM-3B, on both inference-only (Direct[1] and CoT) and post-training settings (SFT) as baselines. We further evaluate closed-source models, including Gemini 2.0 Flash [26] (`gemini-2.0-flash-001`) and advanced thinking model Gemini 2.5 Pro [13] (`gemini-2.5-pro-preview-03-25`) as a reference from state-of-the-art multimodal reasoning. Full training details for all models are provided in App. B.4.

[1] Direct denotes answer prediction without being instructed to conduct intermediate reasoning.

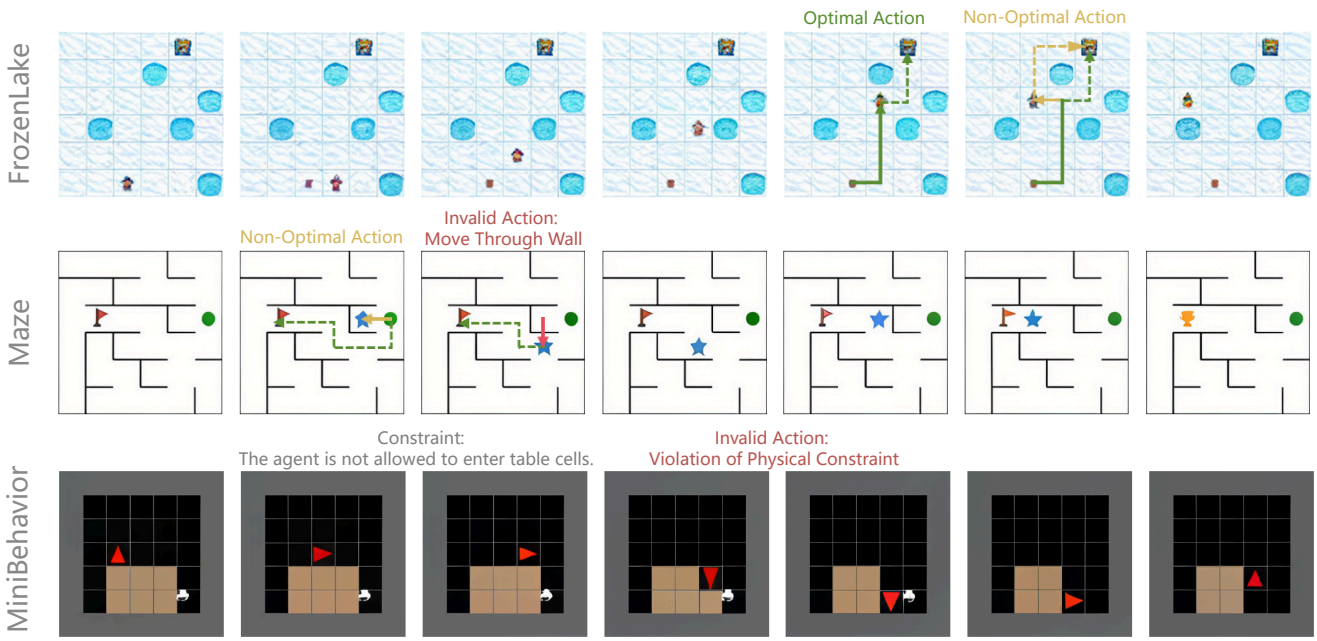

Figure 3. Illustration of each task with generated visual planning traces from LVM, covering different types of actions (optimal, non-optimal and invalid). More cases can be found in App. C.5.

Evaluation Metrics We adopt two complementary evaluation metrics for the selected tasks:

- **Exact Match (EM)** is defined as $\text{EM}_i = \prod_{j=1}^{n} \mathbb{I}(\hat{v}_j = v_j)$. This metric measures whether the model successfully generates the complete and correct planning trajectory that aligns with the shortest optimal valid path. One step of deviation from the optimal solution is considered incorrect.
- **Progress Rate (PR)** is defined as $\text{PR}_i = \frac{1}{n}\sum_{j=1}^{n}\left[\prod_{k=1}^{j}\mathbb{I}(\hat{v}_k = v_k)\right]$. PR measures the ratio of the number of consecutively correct steps (valid forward moves) from the start to the number of steps in the optimal path. This provides a softer signal than Exact Match, capturing the model's ability to make meaningful progress towards a full solution.

Visual Planning Surpasses Textual Planning. Table 1 shows that visual planners (VPFT and VPRL) achieve the highest scores on all tasks, outperforming all language-reasoning baselines. With identical supervised training method via fine-tuning, VPFT exceeds language-based SFT by an average of over 22% in Exact Match (EM), with VPRL further widening the gap. A similar trend is observed in Progress Rate (PR) as well. This highlights the advantages of the visual planning paradigm in visual-centric tasks, where language-driven approaches may be less aligned with task structure. Inference-only models, whether large closed-source systems or smaller open-source MLLMs, struggle with these planning tasks without task-specific tuning. Even the advanced thinking model Gemini 2.5 Pro achieves EM and PR almost below 50% on the more complex MAZE and MINIBEHAVIOR tasks, underscoring the challenges these tasks pose for current models despite being intuitive for humans.

Gains from Reinforcement Learning. The two-stage reinforcement learning approach (VPRL) yields the highest overall performance, surpassing all system variants. After Stage 2, the model achieves near-perfect planning on the simpler FROZENLAKE task (91.6% EM, 93.2% PR) and maintains strong performance on MAZE and MINIBEHAVIOR tasks. This marks a substantial improvement of more than 20% across all tasks over the supervised baseline VPFT. As expected, Stage 1 of our RL training, which enforces output format without teaching planning behavior, yields near-random performance (e.g., 11% EM on FROZENLAKE, see Table 8 in App. C.5). After the full Stage 2 optimization with our reward scheme, the planner achieve its best performance. This gain highlights a key advantage of RL over SFT. VPRL allows the model to freely explore diverse actions and learn from their outcome, while VPFT relies on imitation and tends to fit the training distribution. By encouraging exploitation through reward-driven updates, VPRL learns to capture underlying rules and patterns, leading to stronger planning performance.

Robustness with Scaling Complexity. The advantage of RL also holds when we study the performance of different methods with respect to task difficulties, where a larger grid usually relates to higher difficulties. In Figure 5, as the grid size increases from 3×3 to 6×6 in the FROZENLAKE en-

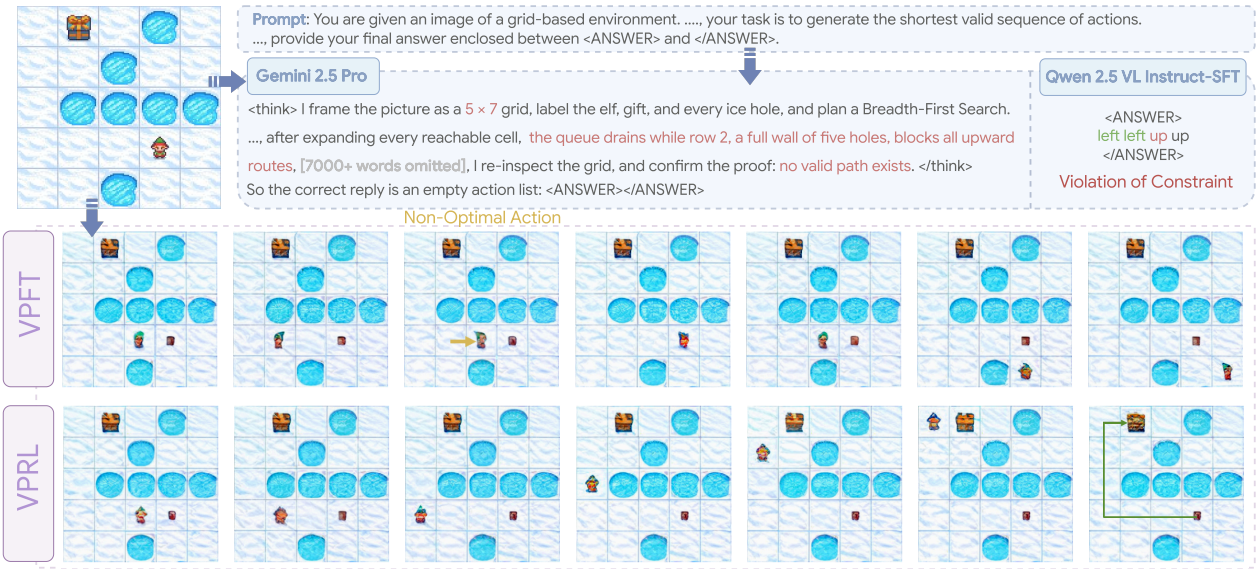

Figure 4. Visualization of a test example from FROZENLAKE comparing visual planning variants (VPFT and VPRL).

vironment, Gemini 2.5 Pro's EM score drops sharply from 98.0% to 38.8%. In comparison, our visual planners not only maintain higher accuracy at all grid sizes but also exhibit a much flatter performance curve. Similarly, VPRL demonstrates even greater stability than VPFT, with EM remaining at 97.6% on 3×3 grids and still achieving 82.4% on 6×6, indicating strong robustness. We observe similar trends in other tasks; see App. C.2 for other tasks.

4. Discussions and Analysis

Error Analysis and Case Study. Figure 3 presents visual planning traces generated by LVM across different tasks. As defined in Section 2.2, the model occasionally takes non-optimal actions that deviate from the shortest path, as seen in the FROZENLAKE. Invalid actions include violations of physical constraints (e.g., walking through walls in MAZE or entering the table in MINIBEHAVIOR), or executing multiple actions in a single step (see App. C.5 for examples).

Figure 4 compares visual planning with language-based reasoning systems. In FROZENLAKE, Gemini 2.5 Pro misinterprets the environment size at the first step, causing cascading errors that lead to concluding an incorrect final answer. Similarly, the language-based SFT baseline makes an invalid action at the third step, reflecting difficulty in tracking states during reasoning. In contrast, visual planning avoids such failures by reasoning directly in the visual modality while reflecting the visual states per action. VPRL demonstrates the ability to take detours to bypass obstacles while still progressing toward the goal, whereas VPFT, lacking this flexibility, gets stuck and fails to reach the destination. More examples are provided in App. C.5.

Random Policy Initialization Enables Exploration. A

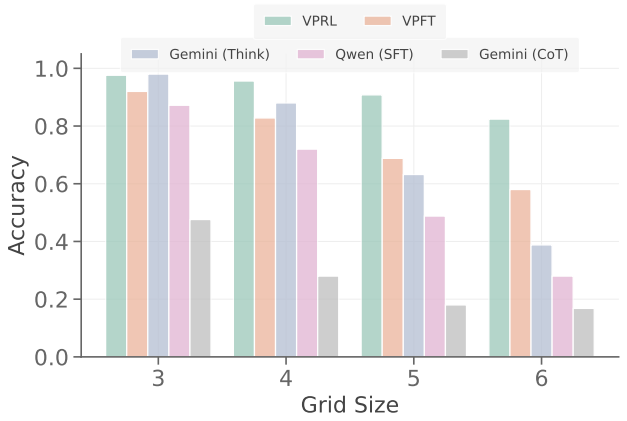

Figure 5. Evaluation of model performance on FROZENLAKE under varying levels of difficulty. As the environment complexity increases with larger grid sizes, language-based reasoning methods experience a sharp decline in performance, whereas visual planning methods exhibit a more gradual drop, demonstrating greater robustness.

natural follow-up question arises: could we directly use VPFT as the policy model for GRPO training rather than intentionally initialize a model with random trajectories? We hypothesize that VPFT, trained via teacher-forcing, inherently limits exploration by repeatedly generating similar actions, resulting in identical rewards. In this case, it yields zero advantage, preventing policy updates and hindering effective learning. We empirically validate this hypothesis by comparing the exploration capabilities of VPFT with VPRL Stage 1 (Figure 6). We observe that VPFT's entropy rapidly declines throughout training, eventually approaching zero, indicating severe exploration limitations. Although earlier VPFT checkpoints exhibit higher entropy,

they produce significantly more invalid actions. In contrast, VPRL Stage 1 demonstrates significantly higher entropy, closely approaching the entropy of the uniform random planner, while maintaining a much lower invalid action ratio. These results justify the necessity of random initialization in our reinforcement learning framework to ensure robust exploration.

VPRL Reduces Invalid Action Failure. Another important benefit of VPRL lies in its effectiveness in reducing invalid actions. To quantify this, we analyze all failed trajectories and compute the proportion that contains at least one invalid action, as opposed to failures caused by non-optimal but valid plans. We refer to this as the *invalid-failure* ratio. As shown in Table 5, VPFT exhibits high ratio ranging from 61% to 78% over three tasks, while VPRL reduces this ratio by at least 24% in all cases, demonstrating that VPRL not only improves success rates, but also encourage the model to stay within valid action spaces during planning.

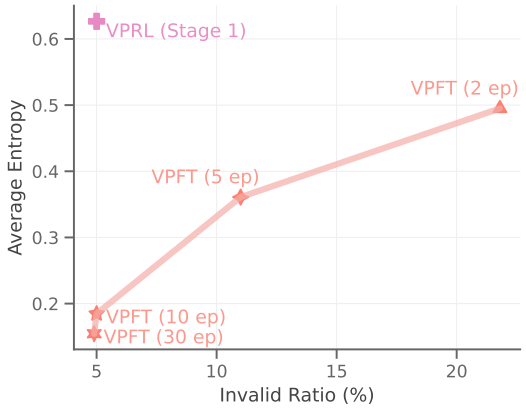

Figure 6. Comparison of exploration capabilities between VPFT and VPRL Stage 1 on FROZENLAKE. VPRL Stage 1 achieves significantly better exploration efficiency, balancing high entropy with a low invalid action ratio, whereas VPFT struggles with diminishing entropy and increased invalid actions over training.

5. Related Work

MLLM Reasoning Recent work has extended CoT prompting [51] to MLLMs through approaches such as grounding visual inputs into symbolic representations, such as graphs or bounding boxes [28, 59]. Other approaches integrate tools to generate visualizations during reasoning [21, 61]. For example, o3 model [37] incorporates visual rationales using tools such as cropping and zooming. MVoT [31] is also essentially a form of tool use: instead of relying on external modules, it invokes itself to generate visualizations of textual reasoning. These methods primarily conduct reasoning in language, with visual components merely illustrating the textual rationale rather than serving as the medium of reasoning. In this work, we take a step further to explore whether multi-step planning can emerge purely within visual representations, enabling reasoning without relying on language at all.

Reinforcement Learning for Visual Reasoning Reinforcement learning has been applied across a wide range of vision-related tasks, especially given the rise of GRPO as in DeepSeek-R1 [16]. Concurrently, in object detection, visual perception [54] is optimized though rewarding high Intersection-over-Union (IoU) scores between predicted and ground-truth bounding boxes [45]. For visual reasoning tasks such as Visual Question Answering (VQA), GRPO has been utilized to optimize the models for longer, more coherent, and logically grounded reasoning traces in textual responses [34, 46, 58, 60]. More recently, similar methods have also been applied to image generation tasks, where the model is guided to reflect on the generated images and refine them recursively based on the alignment with given textual instructions [17, 24, 48]. These approaches focus on pixel-level fidelity and semantic alignment with text, whereas our work leverages RL for goal-oriented visual planning, optimizing multi-step decision-making through visual state transitions without any reliance on language. While prior RL-based approaches ground reasoning traces in textual outputs despite multimodal inputs, the modality mismatch limits the effectiveness of RL in bridging perception and action. For the tasks that are 'vision-first' by design, our visual planning paradigm and two-stage training framework VPRL enable more natural and flexible policy exploration by operating entirely in the visual domain, outperforming all language-based training variants.

6. Conclusion

In this work, we present Visual Planning as a new paradigm for reasoning in visually oriented tasks, challenging the prevailing reliance on language as the primary medium for structured inference. By enabling models to operate entirely through visual state transitions without textual mediation, we demonstrate that purely visual representations can lead to more effective and intuitive planning, particularly in spatially grounded and dynamic tasks. Our proposed two-stage reinforcement learning framework, VPRL, empowered by GRPO, further enhances the planning capabilities of large vision models. It obtains significant gains across three visual navigation tasks, achieving over 40% improvements in task performance than language-based planning and showing stronger generalization on out-of-distribution scenarios. These findings underscore the promise of visual planning as a powerful alternative to text-based approaches. We believe our work opens up a rich new direction for multimodal research, offering a foundation for building more intuitive, flexible, and powerful reasoning systems across a wide range of domains.

References

[1] Arjun Akula, Soravit Changpinyo, Boqing Gong, Piyush Sharma, Song-Chun Zhu, and Radu Soricut. CrossVQA: Scalably generating benchmarks for systematically testing VQA generalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2148–2166, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. 2

[2] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. 1

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 5, 2

[4] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L. Yuille, Trevor Darrell, Jitendra Malik, and Alexei A. Efros. Sequential modeling enables scalable learning for large vision models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 22861–22872, 2024. 2, 5

[5] G Brockman. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016. 5

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1

[7] Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M^3CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8199–8221, Bangkok, Thailand, 2024. Association for Computational Linguistics. 1

[8] Zihui Cheng, Qiguang Chen, Jin Zhang, Hao Fei, Xiaocheng Feng, Wanxiang Che, Min Li, and Libo Qin. Comt: A novel benchmark for chain of multi-modal thought on large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23678–23686, 2025. 1

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

[10] Rohan Choudhury, Guanglei Zhu, Sihan Liu, Koichiro Niinuma, Kris Kitani, and László Jeni. Don't look twice: Faster video transformers with run-length tokenization. *Advances in Neural Information Processing Systems*, 37:28127–28149, 2024. 1

[11] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *The Second Conference on Parsimony and Learning (Recent Spotlight Track)*, 2025. 2, 3

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2

[13] Gemini. Gemini 2.5: Our most intelligent AI model. 2025. Accessed: 2025-05-09. 5, 1

[14] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[15] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7606–7623, Dublin, Ireland, 2022. Association for Computational Linguistics. 1

[16] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2, 3, 8

[17] Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. Can we generate images with cot? let's verify and reinforce image generation step by step. *arXiv preprint arXiv:2501.13926*, 2025. 8

[18] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018. 2

[19] Yunzhuo Hao, Jiawei Gu, Huichen Will Wang, Linjie Li, Zhengyuan Yang, Lijuan Wang, and Yu Cheng. Can mllms reason in multimodality? emma: An enhanced multimodal reasoning benchmark. *arXiv preprint arXiv:2501.05444*, 2025. 1, 2

[20] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 3

[21] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *arXiv preprint arXiv:2406.09403*, 2024. 1, 8

[22] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 1

[23] Michael Igorevich Ivanitskiy, Rusheb Shah, Alex F Spies, Tilman Räuker, Dan Valentine, Can Rager, Lucia Quirke, Chris Mathwin, Guillaume Corlouer, Cecilia Diniz Behn, et al. A configurable library for generating and manipulating maze datasets. *arXiv preprint arXiv:2309.10498*, 2023. 2, 5

[24] Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*, 2025. 8

[25] Emily Jin, Jiaheng Hu, Zhuoyi Huang, Ruohan Zhang, Jiajun Wu, Li Fei-Fei, and Roberto Martín-Martín. Mini-BEHAVIOR: A procedurally generated benchmark for long-horizon decision-making in embodied AI. In *NeurIPS 2023 Workshop on Generalization in Planning*, 2023. 2, 5

[26] Kat Kampf and Nicole Brichtova. Experiment with gemini 2.0 flash native image generation, 2025. Accessed: 2025-04-27. 5

[27] Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. Ds-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning*, pages 18319–18345. PMLR, 2023. 5

[28] Xuanyu Lei, Zonghan Yang, Xinrui Chen, Peng Li, and Yang Liu. Scaffolding coordinates to promote vision-language coordination in large multi-modal models. *arXiv preprint arXiv:2402.12058*, 2024. 8

[29] Chengzu Li, Chao Zhang, Simone Teufel, Rama Sanand Doddipatla, and Svetlana Stoyanchev. Semantic map-based generation of navigation instructions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14628–14640, Torino, Italia, 2024. ELRA and ICCL. 1

[30] Chengzu Li, Caiqi Zhang, Han Zhou, Nigel Collier, Anna Korhonen, and Ivan Vulić. TopViewRS: Vision-language models as top-view spatial reasoners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1786–1807, Miami, Florida, USA, 2024. Association for Computational Linguistics. 1, 2

[31] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*, 2025. 1, 8

[32] Drew Linsley*, Junkyung Kim*, Alekh Ashok, and Thomas Serre. Recurrent neural circuits for contour detection. In *International Conference on Learning Representations*, 2020. 1, 3

[33] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023. 1

[34] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025. 8

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 3

[36] Samuel T. Moulton and Stephen M. Kosslyn. Imagining predictions: mental imagery as mental emulation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364:1273 – 1280, 2009. 1

[37] OpenAI. Introducing OpenAI o3 and o4-mini: Our smartest and most capable models to date. 2025. Accessed: 2025-05-16. 8

[38] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 1

[39] Allan Paivio. Dual coding theory: Retrospect and current status. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 45(3):255, 1991. 1

[40] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, Qixiang Ye, and Furu Wei. Grounding multimodal large language models to the world. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[41] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 4, 1, 3

[42] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, and et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530, 2024. 1

[43] Jonathan Roberts, Mohammad Reza Taesiri, Ansh Sharma, Akash Gupta, Samuel Roberts, Ioana Croitoru, Simion-Vlad Bogolin, Jialu Tang, Florian Langer, Vyas Raina, et al. Zerobench: An impossible visual benchmark for contemporary large multimodal models. *arXiv preprint arXiv:2502.09696*, 2025. 1

[44] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv preprint*, abs/2402.03300, 2024. 2, 3

[45] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025. 8

[46] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025. 8

[47] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020. 3

[48] Junke Wang, Zhi Tian, Xun Wang, Xinyu Zhang, Weilin Huang, Zuxuan Wu, and Yu-Gang Jiang. Simplear: Pushing the frontier of autoregressive visual generation through pretraining, sft, and rl. *arXiv preprint arXiv:2504.11455*, 2025. 8

[49] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. 5, 3

[50] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. 1

[51] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, pages 24824–24837. Curran Associates, Inc., 2022. 8

[52] Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. Liquid: Language models are scalable multi-modal generators. *arXiv preprint arXiv:2412.04332*, 2024. 1

[53] 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, 2024. 2, 5

[54] En Yu, Kangheng Lin, Liang Zhao, Jisheng Yin, Yana Wei, Yuang Peng, Haoran Wei, Jianjian Sun, Chunrui Han, Zheng Ge, et al. Perception-r1: Pioneering perception policy with reinforcement learning. *arXiv preprint arXiv:2504.07954*, 2025. 8

[55] 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*, pages 9556–9567, 2024. 2

[56] Huanyu Zhang, Chengzu Li, Wenshan Wu, Shaoguang Mao, Ivan Vulić, Zhang Zhang, Liang Wang, Tieniu Tan, Furu Wei, et al. A call for new recipes to enhance spatial reasoning in mllms. *arXiv preprint arXiv:2504.15037*, 2025. 1

[57] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2

[58] Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *arXiv preprint arXiv:2503.12937*, 2025. 8

[59] Zhuosheng Zhang, Aston Zhang, Mu Li, hai zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*, 2024. 1, 8

[60] Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. R1-zero's" aha moment" in visual reasoning on a 2b non-sft model. *arXiv preprint arXiv:2503.05132*, 2025. 8

[61] Qiji Zhou, Ruochen Zhou, Zike Hu, Panzhong Lu, Siyang Gao, and Yue Zhang. Image-of-thought prompting for visual reasoning refinement in multimodal large language models, 2024. 8

Visual Planning: Let's Think Only with Images

Supplementary Material

A. Limitations and future work

In this work, we focus exclusively on Large Vision Model (LVM) to investigate visual planning capabilities by eliminating language as a confounding factor *for research purposes*. As such, this choice constrains the model size to 3B as the only available size of LVM, and excludes recently released native multimodal models capable of generating multimodal outputs [9, 52]. However, we argue that the visual planning paradigm can be extended to broader multimodal generation models for use in more diverse tasks, combined with more modalities as long as they support image generation.

Additionally, explicitly generating images introduces computational overhead during inference compared to a textual response. However, we argue that language-based reasoning, especially for thinking models [13], can be equally or more time-consuming. In our demonstration, Gemini generated over 7,000 thinking tokens yet failed to provide the correct answer in the end. The computation overhead introduced by image generation can be alleviated through more compact image representations using fewer tokens [10], which we advocate for future research.

Another limitation in this work lies in the implementation of state-action parsing function. For simplicity, we adopt the rule-based approach that compares pixel-wise features between the current state and the previous state (details in Appendix B.3). While effective in our controlled setup, this method limits generalizability to broader task settings. Nevertheless, we argue that the core idea is extensible and could be supported by well-established computer vision techniques such as segmentation [41], contour detection [32] and etc. We encourage future research to explore more robust and scalable designs for visual state-action parsing to advance visual planning systems.

Broader Impact This work introduces a novel paradigm of visual planning, where agents reason and act entirely within the visual modality without reliance on textual intermediaries. By demonstrating that models can plan through sequences of images, this research opens new possibilities for the way human and AI system interacts, particularly in domains like robotics, navigation, and assistive technologies, where perception and decision-making are tightly coupled. As the first step toward planning grounded purely in visual representations, our work lays the foundation for AI systems that integrate both verbal and non-verbal reasoning. We advocate for future research into more holistic multimodal thinking systems where interleaved text and image traces enable richer, more human-like reasoning, and emphasize the importance of strengthening the visual component in such traces for improved planning and cognition.

B. Implementation details

B.1. Dataset

Task Action Space. FROZENLAKE and MAZE both involve four primitive navigation actions: `up`, `down`, `left`, and `right`. MINIBEHAVIOR includes a more complex action space with two additional operations: `pick`, `drop`.

Dataset preparation. For both FROZENLAKE and MAZE, we construct environments of grid sizes ranging from 3×3 to 6×6. For each size, we sample 1250 environments, with 1000 used for training and 250 held out for testing (Table 2). Each environment here is guaranteed to have a unique layout, and the agent is randomly initialized at a grid from which the goal is reachable, forming the initial state v_0. Due to the relatively limited diversity of environments layout in MINIBEHAVIOR, where the complexity arises primarily from the action space, sampling unique environments in a small grid size becomes challenging. Therefore we focus only on grid sizes 7×7 and 8×8, allowing duplicates in layout but varying agent spawn positions to ensure sufficient data volume. To prevent data leakage, we split the dataset based on layout identity, ensuring no layout overlap between the training and test sets.

We next describe the dataset construction procedures corresponding to the training setups outlined in Section 3, with the number of samples per task summarized in Table 3.

- **SFT in Text** (Baseline): For each environment, we sample an optimal trajectory consisting of a sequence of visual states $(v_0, \ldots, v_n)$ as the ground truth. Each transition between states is determined by an action, enabling us to derive a corresponding verbal action sequence $(a_0, \ldots, a_{n-1})$. The input to the model is formulated by concatenating a textual prompt with an image representation of the initial state v_0, while the target output is the verbalized action sequence representing the optimal trajectory. The detailed prompt is provided in Appendix D.
- **VPFT**: We utilize the same set of optimal trajectories as the language-based reasoning baseline described above. In the visual scenario, each trajectory generates multiple input-target pairs by pairing the state at timestep t as the input with the subsequent state at timestep $t + 1$ as the target.
- **VPRL**:

  - Stage 1: This dataset serves solely for format control training of the visual backbone. For each environment, we enumerate all possible trajectories from the initial state as v_0 and generate corresponding input-target pairs. Duplicate pairs are filtered to maintain a balanced distribution.
  - Stage 2: To ensure fairness and comparability, this dataset uses the same input states as VPFT.
- **VPFT***: We conduct an ablation study (indicated with *) where VPFT is also trained in two stages, mirroring the structure of VPRL. Stage 1 follows the same procedure as VPRL Stage 1, focusing on format supervision using enumerated visual inputs. Stage 2 reuses the original VPFT training pipeline, learning from optimal trajectories. Experimental results and analysis see Appendix C.4.

Note: For both textual and visual planning setups, evaluation is performed using only the initial state v_0 of each test environment as input.

Dataset Statistics. We evaluate the performance of different system variants in in-distribution and out-of-distribution (OOD) settings. Table 2 show the training data distribution over different grid sizes across three tasks. The numbers of training and testing samples for different system variants are shown in Table 3. For OOD evaluation, the enlarged grid sizes are shown in Table 7. OOD evaluation data includes 250 samples for each task.

FROZENLAKE				
Grid Size	3	4	5	6
Train	1000	1000	1000	1000
Test	250	250	250	250
MAZE				
Grid Size	3	4	5	6
Train	1000	1000	1000	1000
Test	250	250	250	250
MINIBEHAVIOR				
Grid Size	7		8	
Train	796		801	
Test	204		199	

Table 2. Distribution of training dataset by grid sizes for each task. Value indicates the number of environments.

B.2. Models

Large Vision Model (LVM) [4] is an autoregressive models for image generation, which is only pretrained with image sequences with no exposure to language data. The model uses a tokenizer based on the VQGAN architecture [12], which extracts visual information from raw images and encodes it into 256 tokens from a fixed codebook. The image is generated in an auto-regressive manner with discrete tokens, which are then fed into the image detokenizer. Although LVM supports multiple model sizes, only the 3B-parameter version is publicly available; thus, we use this variant in our experiments. For a fair comparison, we use Qwen 2.5-VL-Instruct [3] with a matching parameter size as our language-based baseline.

B.3. Reward Implementation

We adopt rule-based state-action parsing function $\mathcal{P}$ and progress map $D(v)$ in VPRL. For the progress map, we apply the Breadth First Search (BFS) to search for the optimal trajectories and calculates the progress at each position in the grid for each task. The progress map are then used as a reward signal to guide VPRL training.

Specifically, for state-action parsing function, we parse the state and identify the difference between current state and previous state through pixel-wise feature extractor. We first convert both input and predicted states into a coordinate-based representation by dividing the image into a grid based on its size. Each region corresponds to a discrete coordinate in the environment. To reduce sensitivity to color and focus on structural differences, we convert all images to grayscale. We subsequently compute the Intersection-over-Union (IoU) between each coordinate in the predicted state and the coordinate in the input state that contains the player (input coordinate). The coordinate in the predicted state with the highest IoU is selected as the predicted agent position. The action is then inferred by comparing the start and predicted positions according to task-specific movement rules. For example, in the MAZE environment, movement across walls is not allowed and would be considered invalid.

Notably, to detect the invalid transitions, such as the disappearance of agents, we also calculate the pixel-wise mean squared error (MSE) between corresponding coordinates to measure local visual differences. If two coordinates exhibit significant MSE differences exceeding a predefined threshold, we treat them as the potential source and destination of a movement (agent disappears from one and appears in another). If only one such coordinate is found, we treat it as a disappearance event, indicating an invalid transition.

In MINIBEHAVIOR, we extend this logic to identify pick and drop actions. A pick is detected when the IoU between the printer's location in the input and predicted states falls below a threshold, indicating that the printer has been removed. A drop is inferred when a coordinate corresponding to the table region shows a large MSE increase, suggesting the printer has been placed there. Additional edge cases in these tasks are omitted for brevity.

For reward computation, if the predicted action is valid, we compare the progress values from the progress map $D(v)$ between the input and predicted states. A reward of 1 is given if the predicted state shows greater progress toward the goal than the input state; otherwise, the reward is

Task	Split	SFT in Text	VPFT	VPRL		VPFT*	
				Stage 1	Stage 2	Stage 1	SFT
FROZENLAKE	Train	4000	12806	170621	12806	170621	12806
	Test	1000	1000	N/A	1000	N/A	1000
MAZE	Train	4000	14459	156682	14459	156682	14459
	Test	1000	1000	N/A	1000	N/A	1000
MINIBEHAVIOR	Train	1597	9174	90808	9174	90808	9174
	Test	403	403	N/A	403	N/A	403

Table 3. Number of training and test samples for each task and method. For visual planning, the numbers here are represented in image pairs, which correspond to the same number of trajectories for SFT in Text.

0. Invalid actions are penalized with a reward of -5.

Our method and reward modeling approach are readily generalizable to other visual tasks. With reference to computer vision techniques such as segmentation [41] and contour detection [32], the pixel-level analysis used in our framework can be easily extended to a wide range of structured visual environments. Furthermore, our reward design is broadly applicable to planning tasks in general. Since actions in most planning settings can naturally be categorized into one of three types (valid and helpful, valid but non-progressing, or invalid), our simple reward structure remains intuitive and effective across tasks.

B.4. Training details

For all post-training experiments, we apply Low-Rank Adaptation (LoRA) [20] on both attention layers and feed-forward layers. The detailed hyper-parameters are shown in Table 4. Only the loss of the labels is calculated in an instruction-tuning manner [49] for SFT. The image tokenizer and detokenizer are frozen during training. We use the AdamW optimizer [35] for all training procedures.

When SFT for textual planning and visual planning, we train the model for a maximum of 30 epochs. For VPRL, we first do stage 1 on random trajectories for 10 epochs for the purpose of exploration. We then use GRPO to optimize the model for planning for another 10 epochs for stage 2. We sample a group of 10 candidate responses per prompt to compute the advantages accordingly. To encourage a balance between exploration and exploitation, we apply a KL divergence penalty with a coefficient $\beta = 0.001$. We use the TRL library for training [47]. We've conducted our experiments on the machine with 8×A100 GPUs.

B.5. Licenses

Model-wise, Large Vision Model and Qwen 2.5 VL are under the Apache-2.0 license. TRL is under the Apache-2.0 license. We collect the MAZE dataset with our own Python scripts. FROZENLAKE is collected from OpenAI Gym under the MIT License.

C. Results

C.1. Training

The reward curves with standard deviation for all tasks are shown in Figure 7. The shaded regions indicate the standard deviation across groups. For better visualization, we apply Gaussian smoothing to both the reward values and their corresponding standard deviations.

C.2. Performance with Scaling Difficulties

We evaluate the performance of different methods with respect to task difficulty in MINIBEHAVIOR and MAZE, as shown in Figure 8. Our visual planners consistently achieve higher accuracy across all grid sizes and exhibit notably flatter performance curves, indicating greater robustness to increasing environment complexity.

Interestingly, in MINIBEHAVIOR, we observe that the accuracy of visual planners increases with grid size, which is in contrast to the trend exhibited by textual planners. We hypothesize that this is due to the fixed layout components in this task, specifically, the presence of only a table and a printer. This maintains consistent layout complexity across different grid sizes and allows knowledge acquired in smaller grids to generalize effectively to larger grids. This suggests that visual planning better captures and transfers structural patterns in the environment.

C.3. Out-of-Distribution Performance

Figure 9 illustrates generated images from VPFT and VPRL on OOD scenarios across MAZE, FROZENLAKE, and MINIBEHAVIOR tasks. Notably, both models exhibit a certain level of visual generalization to unseen configurations, such as larger grids with finer step granularity, despite not encountering them during training.

We subsequently quantitatively test generalization by evaluating the model on OOD environments with larger grid

Hyper-Parameters	SFT in Text	VPFT	VPRL		VPFT*	
			Stage 1	Stage 2	Stage 1	SFT
Epochs	30	30	10	10	10	30
Learning Rate	1e-5	1.5e-4	1.5e-4	5e-5	1.5e-4	1.5e-4
Train Batch Size	16	8	8	1	8	8
Group Size	N/A	N/A	N/A	10	N/A	N/A
Grad Accumulation	2	1	1	1	1	1
GPUs	8	8	8	8	8	8

Table 4. Hyper-parameters of training both textual and visual planners.

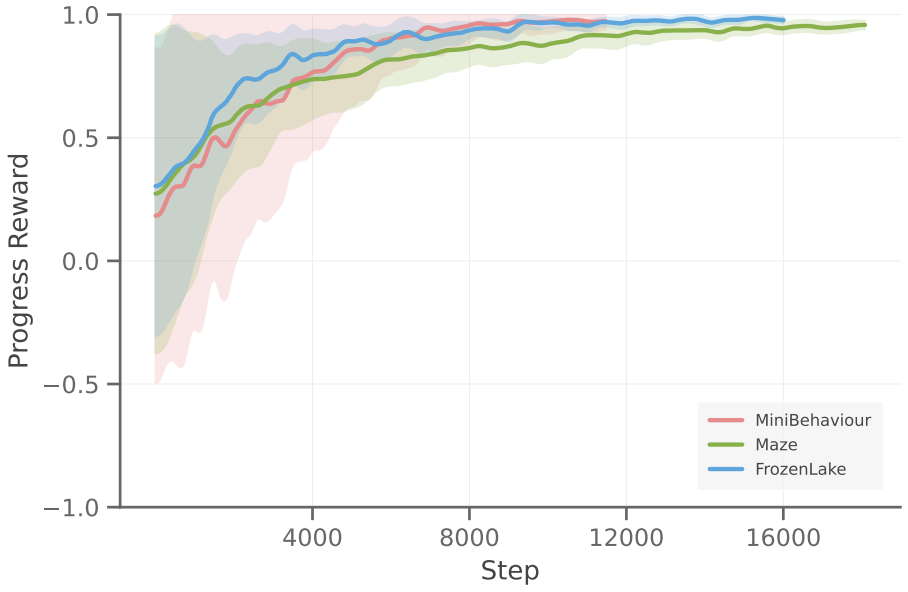

Figure 7. Reward curves with standard deviation for VPRL on FROZENLAKE, MAZE and MINIBEHAVIOR.

Task	Invalid-Failure Ratio (%)	
	VPRL	VPFT
FROZENLAKE	**36.9**	60.6
MAZE	**25.1**	73.7
MINIBEHAVIOR	**29.6**	78.3

Table 5. We compute the percentage of failed trajectories that are caused by at least one invalid action, rather than a suboptimal but valid action. Lower values indicate better action validity control.

	Exact Match (%)			
Model	3×3	4×4	5×5	6×6
VPFT*	86.4	73.6	50.0	33.2
VPFT	**92.0**	**82.8**	**68.8**	**58.0**

Table 6. Exact Match performance of VPFT and VPFT* across different grid sizes in FROZENLAKE.

sizes. We find that SFT models performs poorly, while VPRL still demonstrates a certain level of visual planning capability as shown in Table 7. VPRL consistently outperforms VPFT in both Exact Match and Progress Rate, suggesting that it, to some degree, captures underlying planning strategies rather than merely memorizing training patterns.

C.4. Ablation: The Role of Stage 1

To better understand the role of Stage 1 in our two-stage framework, we conduct an ablation study isolating its impact. The primary purpose of Stage 1 is not to improve planning performance directly, but rather to initialize a policy

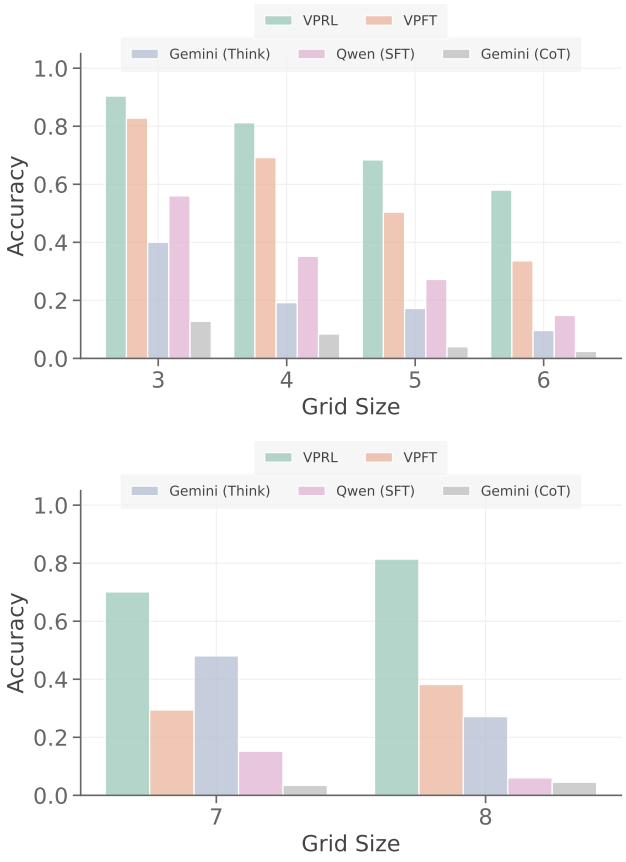

Figure 8. Performance across different grid sizes, reflecting task difficulty. **Left**: MAZE. **Right**: MINIBEHAVIOR. Visual planners consistently maintain higher accuracy and exhibit flatter performance curves, indicating robustness to increasing complexity.

Model	FROZENLAKE (7×7)		MAZE (7×7)		MINIBEHAVIOR (9×9)	
	EM (%)	PR (%)	EM (%)	PR (%)	EM (%)	PR (%)
VPFT	9.6	15.3	9.2	17.8	0.0	5.8
VPRL	**20.4**	**31.2**	**10.0**	**21.6**	**0.4**	**14.7**

Table 7. Out-of-distribution (OOD) performance on enlarged grids. Models are trained on smaller grids and evaluated on the sizes indicated in parentheses.

with strong exploration capacity and valid output formats. To verify this, we reuse the original VPFT training pipeline, i.e., learning from optimal trajectories, but start from the Stage 1 checkpoint as VPFT*. Surprisingly, this variant yields lower final performance on FROZENLAKE compared to standard VPFT. This result supports our hypothesis that Stage 1 does not contribute to planning ability itself, but instead provides an exploration-friendly initialization that facilitates effective reinforcement learning in Stage 2.

Model	FROZENLAKE		MAZE		MINIBEHAVIOR	
	EM (%)	PR (%)	EM (%)	PR (%)	EM (%)	PR (%)
VPRL Stage 1	11.1	27.2	9.6	22.7	0.5	14.2
VPRL Stage 2	**91.6**	**93.2**	**74.5**	**77.6**	**75.8**	**83.8**

Table 8. Performance comparison of VPRL Stage 1 and Stage 2 across all three tasks.

C.5. Visual Planning Results

VPRL Stage 1 and Stage 2 Table 8 presents results for each stage of VPRL. After Stage 1, the model learns to generate plausible images but lacks goal-directed behavior, resulting in near-random performance across tasks. In Stage 2, reinforcement learning instills purposeful planning, enabling the model to align generations with the goal and outperform VPFT across all benchmarks.

Generated Visual Planning Traces for Illustration Figure 10 shows the generated visual planning traces for FROZENLAKE, with Figure 11 for MAZE and Figure 12 for MINIBEHAVIOR. Each visual trajectory begins with the initial state as the input (the first frame), followed by a sequence of intermediate states generated by VPRL that form the predicted visual plan.

We include examples from three categories: (1) **Optimal cases**, where the model successfully generates the shortest valid path to the goal; (2) **Non-optimal cases**, where the agent fails to reach the goal within the optimal number of steps due to intermediate non-optimal actions; and (3) **Invalid cases**, in which the generated trajectory contains invalid actions that violate environment constraints, preventing task completion. Notably, as illustrated in Figure 3, we still observe occasional planning errors. While reinforcement learning significantly improves generalization compared to supervised fine-tuning, it does not fully eliminate such failure cases.

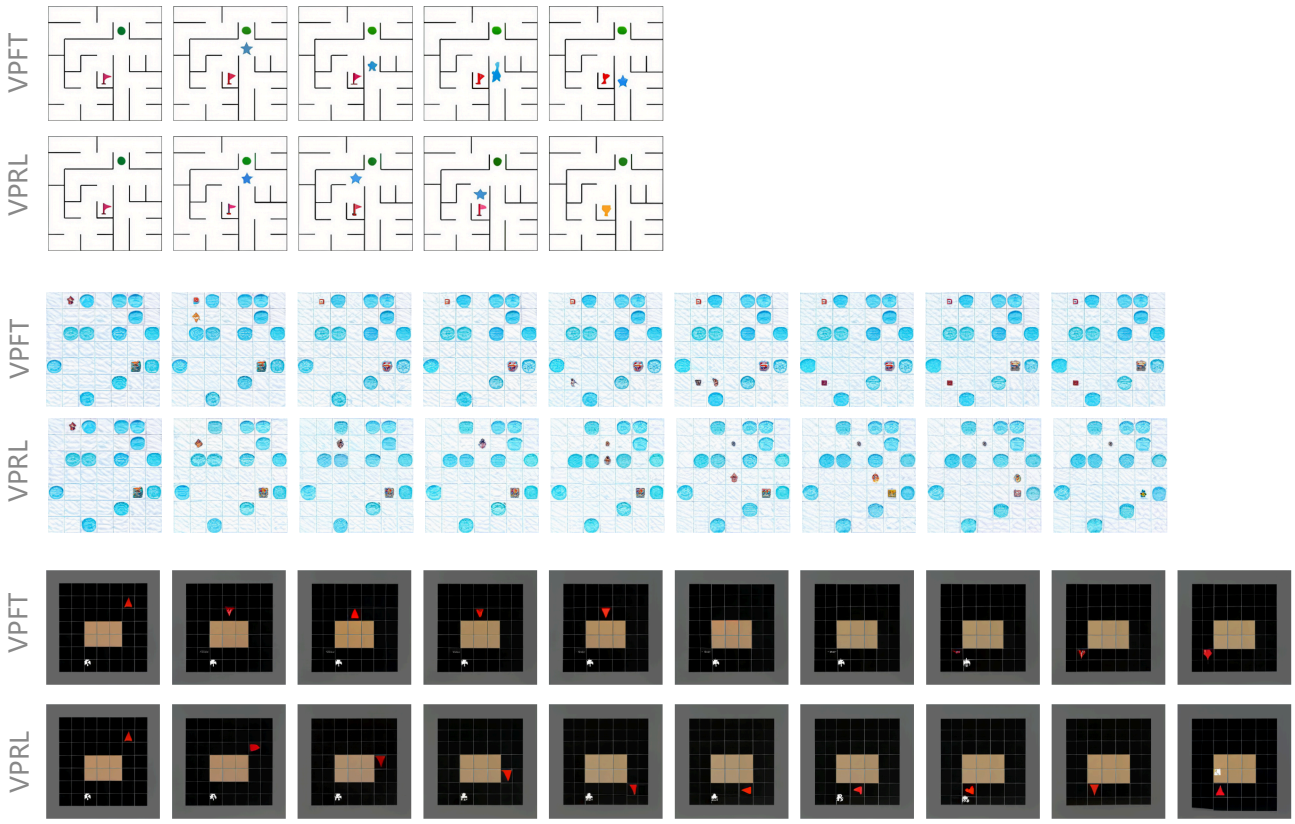

Figure 9. Qualitative comparison of visual planning outputs from VPFT (top) and VPRL (bottom) on out-of-distribution (OOD) scenarios with unseen larger grid size across MAZE, FROZENLAKE, and MINIBEHAVIOR. Each example shows a failure case from VPFT contrasted with a successful trajectory generated by VPRL under the same environment configuration.

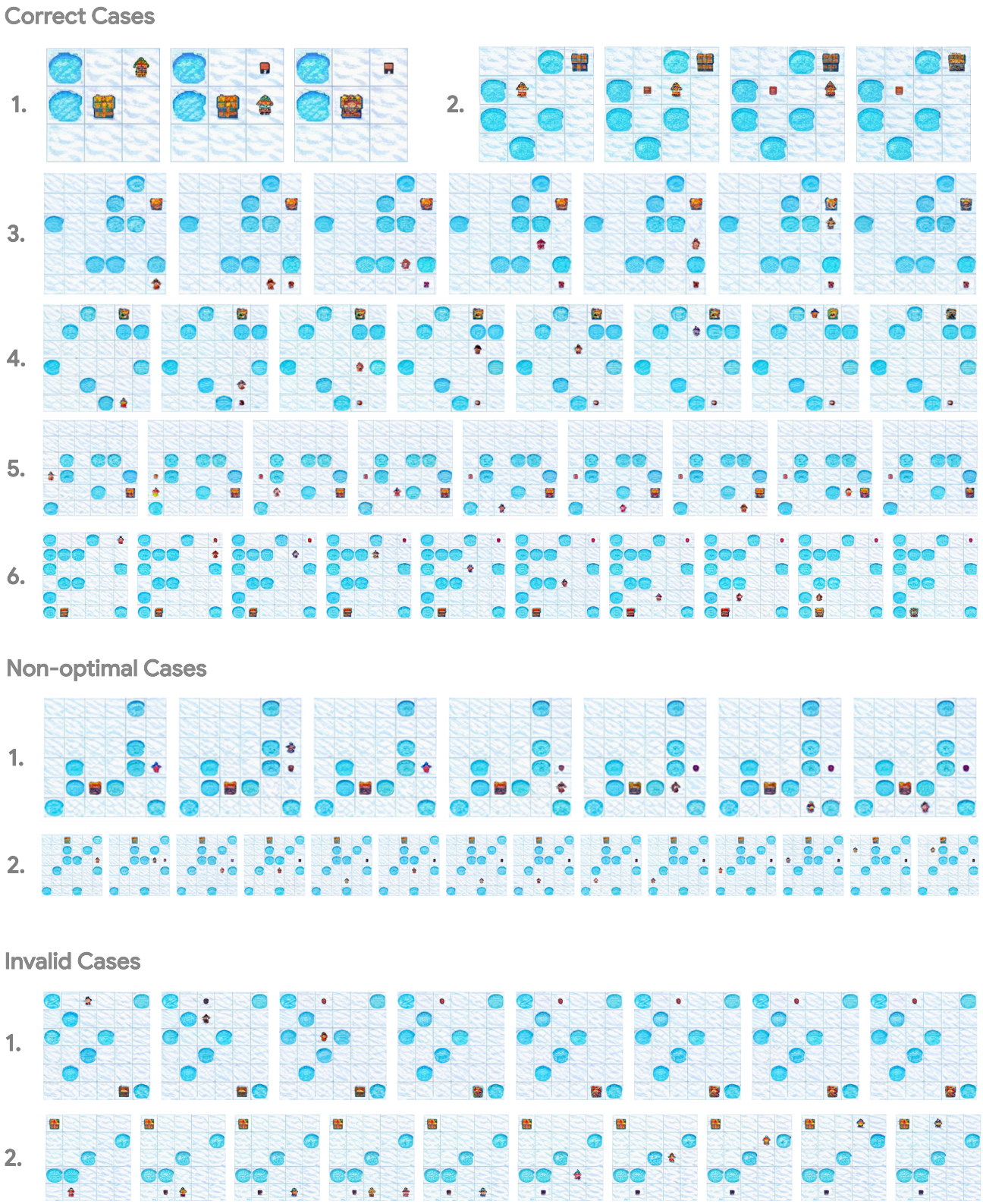

Figure 10. Generated visual planning trajectories from VPRL on the FROZENLAKE test set. We illustrate three representative categories: optimal, non-optimal, and invalid cases. In non-optimal examples, the model occasionally enters local loops but still has the chance to make progress toward the goal, see the first and third trajectories. In invalid cases, despite a significant reduction in failure rate, VPRL still exhibits errors such as disappearing agents, contradictory actions (e.g., simultaneous left and right), or unrealistic teleportation.

Figure 11. Generated visual planning trajectories from VPRL on the MAZE test set. We illustrate three representative categories: optimal, non-optimal, and invalid cases. In non-optimal examples, similar to FROZENLAKE, the model occasionally enters redundant loops but still progresses toward the goal. Invalid cases include maze-specific errors, such as the agent erroneously traversing through walls, violating the structural constraints of the environment. Notably, we observe that in the last invalid case, the agent is able to plan an optimal trajectory in subsequent steps.

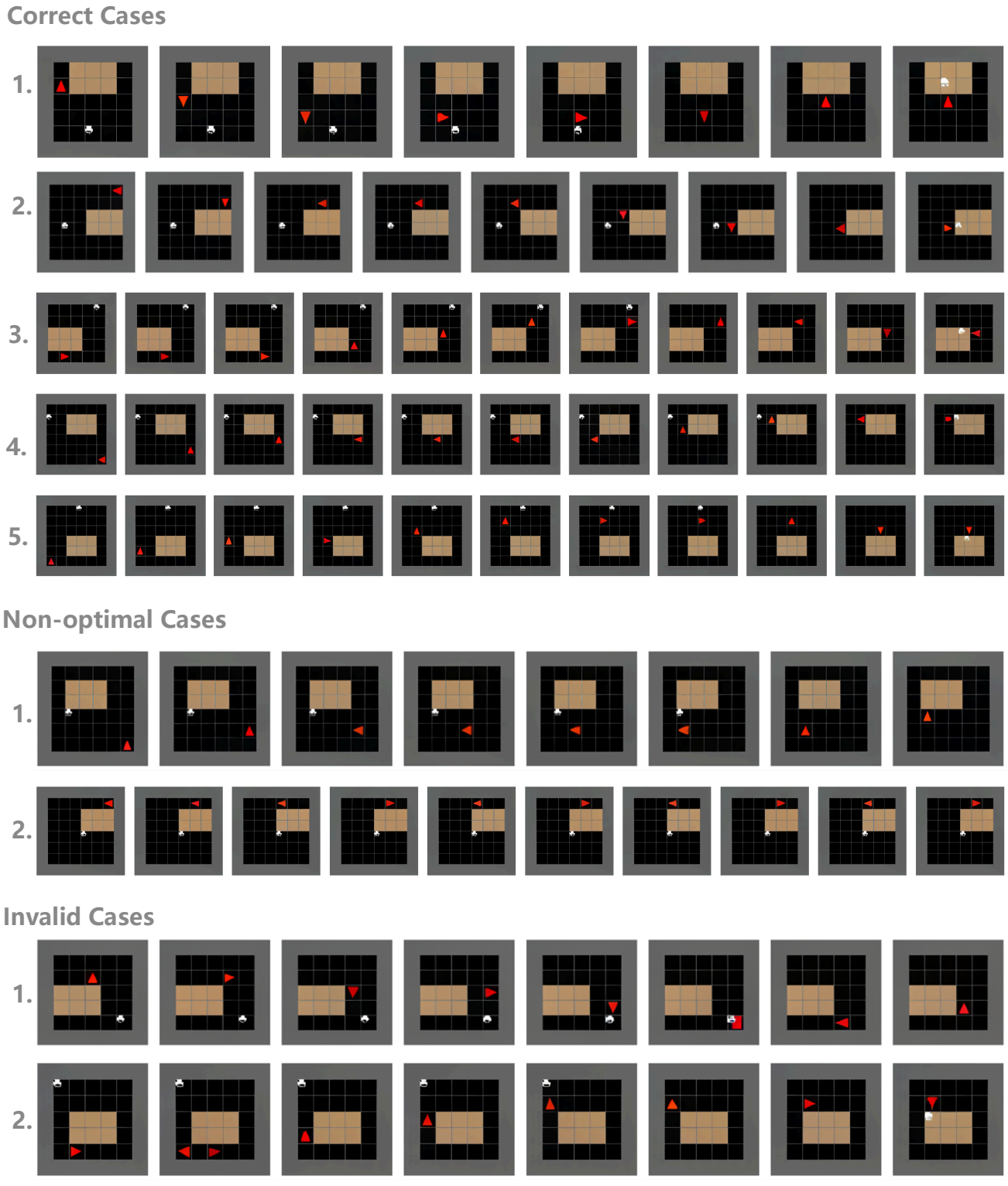

Figure 12. Generated visual planning trajectories from VPRL on the MINIBEHAVIOR test set.

D. Prompting Templates

FROZENLAKE

```
Task: Frozen Lake Shortest Path Planning

You are given an image of a grid-based environment. In this environment:
- An elf marks the starting position.
- A gift represents the goal.
- Some cells contain ice holes that are impassable for the elf.
- The elf can move in one of four directions only: "up", "down", "left", or "right". Each
    move transitions the elf by one cell in the corresponding absolute direction. Diagonal
    movement is not permitted.

Your task is to analyze the image and generate the shortest valid sequence of actions that
    moves the elf from the starting position to the goal without stepping into any ice holes
    .

Provide your final answer enclosed between <ANSWER> and </ANSWER>, for example: <ANSWER>
    right up up</ANSWER>.

<image>
```

MAZE

```
Task: Maze Shortest Path Planning

You are given an image of a maze environment. In this environment:
- A green circle marks the starting position of the agent.
- A red flag marks the goal.
- The agent can move in one of four cardinal directions only: "up", "down", "left", or "
    right". Each move shifts the agent by exactly one cell in that direction. Diagonal
    movement is not permitted.
- The black maze walls are impassable. The agent cannot pass through any wall segment.

Your task is to analyse the image and produce the shortest valid sequence of actions that
    moves the agent from its starting position to the goal without crossing any wall.

Provide your final answer enclosed between <ANSWER> and </ANSWER>, for example: <ANSWER>
    right up

<image>
```

MINIBEHAVIOR

```
Task: Mini-Behavior Installing the Printer

You are given an image of a grid-based environment. In this environment:
- The red triangle represents the agent.
- The white icon represents the printer, which must be picked up by the agent.
- The brown tiles represent the table, where the printer must be placed.

The agent can take the following actions:
- "up", "down", "left", "right": each action shifts the agent by exactly one cell in that
    direction. Diagonal movement is not permitted.
- "pick": pick up the printer if it is in one of the four adjacent cells surrounding the
    agent. This action is invalid if there is no adjacent printer.
- "drop": drop the printer onto the table if the agent is adjacent to a table cell. This
    action is invalid if there is no adjacent table.

Constraints:
```

```
- The agent cannot move through the table tiles.
- The agent cannot move through the printer until it has been picked up. After picking it up
    , the agent may move through the cell that previously contained the printer.

Your task is to analyse the image and produce the shortest valid sequence of actions that
    allows the agent to pick up the printer and then place it on the table.

Provide your final answer enclosed between <ANSWER> and </ANSWER>, for example: <ANSWER>
    right down right pick left drop</ANSWER>.

<image>
```