# EnvHint: Progressive Learning from Hinted Environments for VLM Agents

**Anonymous ACL submission**


## Abstract

Reinforcement learning (RL) has emerged as a promising approach for improving the reasoning and environment interaction abilities of vision-language models (VLMs). However, RL in sparse-reward VLM agent environments is often hindered by **exploration stagnation**: successful trajectories are rarely discovered early in training, and even successful rollouts provide only weak intermediate supervision. To address this problem, we introduce **EnvHint**, a progressive training framework for VLM agents. The framework constructs a **hinted environment** that generates paired hinted and base observation views online from the evolving environment state, enabling **hint-guided rollout exploration** while preserving the underlying task dynamics. We then apply **backward hint annealing** to gradually transfer training back to the original no-hint setting, so that behaviors discovered with hints are internalized by the policy rather than remaining dependent on privileged observations. Through controlled experiments and analysis, we show that EnvHint delivers substantial gains, and we further analyze how RL algorithm choice, hint intensity, and hint modality affect VLM agent learning.


## 1 Introduction

Reinforcement learning (RL) has shown strong potential for improving the complex reasoning abilities of vision-language models (VLMs), including tasks such as mathematical reasoning, spatial reasoning, and visual puzzle solving (Huang et al., 2025; Wang and Ling, 2025; Pan and Liu, 2025; Feng et al., 2025; Chen et al., 2025). Beyond static reasoning benchmarks, a growing body of work has also begun to explore how RL can improve VLM agents that learn through sustained interaction with multimodal environments and feedback signals (Wang et al., 2025a; Wei et al., 2025).

However, compared with traditional reasoning tasks, RL in VLM agent settings is more prone to

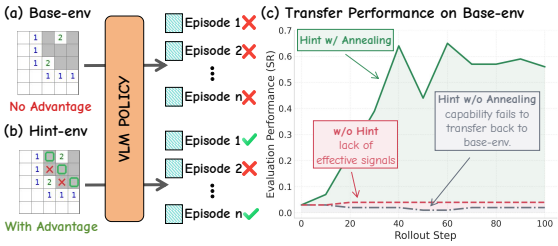



Figure 1: **EnvHint for rollout discovery and transfer.** Training with hinted interaction improves exploration, while backward hint annealing transfers the gains back to the original base environment.

**exploration stagnation** (Qiyuan et al., 2026) when rewards are sparse. Consider the Minesweeper base environment in Figure 1(a): the agent must explore over many possible action sequences, but successful trajectories are rarely sampled early in training. Even when a successful trajectory is sampled, rewards are typically observed only after completing the full interaction trajectory, while intermediate decisions receive little supervision. As a result, early rollouts are often dominated by uninformative episodes and training makes little progress.

Recent work in text-only reasoning has begun to explore the use of **hints**, *namely additional task-relevant guidance or privileged information*, to improve rollout discovery and mitigate exploration stagnation (Xia et al., 2026; Zhang et al., 2025c; Qiyuan et al., 2026; Zhang et al., 2025b). For example, SAGE samples a compact hint for each prompt to increase rollout diversity on hard questions (Liao et al., 2026). However, the use of hints in environment-interactive VLM tasks introduces two key challenges: **(1)** *How can hints be constructed for dynamic environments?* Unlike static reasoning problems, where a hint can often be prepared once in advance, multi-turn agent interaction requires step-specific hints that evolve with the environment state. This requires constructing a **hinted environment** that can *generate task-relevant guidance from the current environment*

*state as the interaction unfolds while remaining aligned with the original base environment*. In Minesweeper, for example, such a hinted environment could mark cells that are already inferable as safe or dangerous from existing evidence, corresponding to the green boxes and red crosses in Figure 1(b). **(2)** *How can hints be used for effective training and transfer back to the original no-hint environment?* This requires ensuring that behaviors discovered with hints are internalized by the policy rather than remaining dependent on privileged observations, since training and inference are performed under different observation conditions. Figure 1(c) further illustrates these challenges empirically: learning stagnates in the base environment, while gains obtained with hints alone fail to reliably transfer back to the original environment.

In this paper, we present **EnvHint**, a progressive learning framework for sparse-reward multi-turn VLM agents that addresses both challenges. Our method first **constructs hinted-environment observations** by modifying the environment implementation to generate paired hinted and base observations as the environment state evolves. The hinted views provide task-relevant guidance through text, vision, or both, while remaining aligned with the base views under the same task instance. It then uses these views to alleviate exploration stagnation and **improve rollout discovery** under sparse rewards, where the policy would otherwise struggle to discover useful behaviors early in training. Finally, it applies **backward hint annealing**, progressively replacing hinted observations with corresponding base observations to transfer training back to the original no-hint setting. This allows behaviors discovered with hints to be internalized in the base observation space rather than remaining tied to privileged observations.

We evaluate EnvHint on three Gym-V-based multimodal multi-turn environments with sparse rewards: Minesweeper, FrozenLake, and Hallway (Meng et al., 2026). Across these environments, EnvHint improves the overall average success rate by 0.27 over the backbone. We further show that the full pipeline substantially outperforms simpler variants such as *REINFORCE++ only*, *Hint-only training*, and *Immediate base transfer*. Additional analyses show that the REINFORCE++-based version of EnvHint yields earlier and more reliable gains than a GRPO-based alternative, while hint intensity and modality affect how well hint-discovered behaviors are internalized and transferred to the original no-hint setting.

Our contributions are threefold:

- We identify exploration stagnation in sparse-reward settings as a central bottleneck in VLM agent training, and address it through hinted environment construction and hint-guided rollout exploration to discover useful trajectories.
- To address the difficulty of internalizing behavior discovered under hints, we propose backward hint annealing to gradually transfer training back to the original no-hint setting.
- Through controlled experiments and analysis, we show that EnvHint improves the overall average success rate by 0.27 over the backbone, and we further discuss how RL algorithm, hint intensity, and hint modality shape VLM agent learning in EnvHint.

## 2 Related Work

### 2.1 Agentic RL for VLMs

Recent VLM post-training increasingly focuses on multi-turn agentic interaction (Zhang et al., 2026; Shi et al., 2026; Gu et al., 2025; Wu et al., 2025b; Tan et al., 2025; Gao et al., 2025). More broadly, multimodal agent research covers GUI control, embodied or real-world interaction, tool-based visual reasoning, and benchmarks for multi-step visual interaction (Xu et al., 2025; Li et al., 2025a; Meng et al., 2026; Chen et al., 2025; Lu et al., 2025; Wu et al., 2025a; Su et al., 2025; Wang et al., 2026; Xie et al., 2024; Cheng et al., 2025). These directions broaden the scope of multimodal agents, but their main challenges are often shaped by action interfaces, external tools, or real-world control complexity. Among the closest works to our setting, VAGEN (Wang et al., 2025a) and RAGEN (Wang et al., 2025b) focus on long-horizon environment interaction, delayed rewards, and policy improvement, while GTR (Wei et al., 2025) analyzes optimization failures in RL-based VLM agent training. Within this multi-turn environment-interaction regime, less attention has been paid to RL training under sparse rewards and the resulting exploration stagnation, which is the focus of our work.

### 2.2 Hint-Augmented Training

To mitigate exploration stagnation under sparse rewards, recent work explores the use of hints, guidance, or privileged information to improve learning.

Representative methods such as SAGE (Liao et al., 2026), HiPO (Qiyuan et al., 2026), and POPE (Qu et al., 2026) show that external guidance can effectively prevent signal collapse, recover informative updates, and expand the solvable task space (Zhang et al., 2025a; Li et al., 2025b). However, most of this literature is developed in text-centric reasoning or RLVR settings, where hints are primarily textual and training remains close to single-turn post-training (Huang et al., 2025; Yan et al., 2025; Hübotter et al., 2026). While prior work (Carta et al., 2020; Selvaraju et al., 2019) suggests that visual hints can provide useful supervision, hint-augmented training for sparse-reward multi-turn VLM agent learning remains underexplored.

## 3 Preliminary

### 3.1 Multimodal Agentic RL

Following prior multimodal agentic RL work (Wang et al., 2025a), multi-turn VLM tasks can be modeled as a partially observable Markov decision process (POMDP) (Åström, 1965). At each turn $t$, the agent conditions on a multimodal observation $o_t$ and produces a text action $a_t$. The environment then returns reward $r_t$ and the next observation $o_{t+1}$. In our sparse-reward settings, rewards are typically zero for most intermediate transitions and become non-zero only when the final task outcome is verified. The training objective is to learn a VLM policy $\pi_\theta$ that maximizes the expected cumulative discounted return over a trajectory, $\max_\theta \mathbb{E}_{\pi_\theta}\left[\sum_t \gamma^t r_t\right]$, where $\gamma \in [0, 1]$ is the discount factor.

### 3.2 Policy Optimization Algorithm

We build on critic-free return-based policy-gradient optimization and review three common formulations that differ in credit-assignment strategy. These methods can be expressed as the following weighted policy-gradient objective:

$$J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}\left[\sum_{t=0}^{T-1} w_t \log \pi_\theta(a_t \mid o_t)\right], \quad (1)$$

where $T$ is the trajectory length and $w_t$ is a scalar weight assigned to action $a_t$. REINFORCE (Williams, 1992) uses the Monte Carlo return as the action weight, with $w_t$ defined as $G_t$, where $G_t = \sum_{k=t}^{T-1} \gamma^{k-t} r_k$. Although simple and critic-free, raw returns often suffer from high variance, especially in long-horizon decision-making problems. GRPO (Shao et al., 2024) instead uses a group-relative trajectory-level estimator, with $w_t$ defined as $A_{\text{GRPO}}$, where $A_{\text{GRPO}} = \frac{R-\mu_R}{\sigma_R+\epsilon}$ and $R = \sum_{t=0}^{T-1} r_t$. For GRPO, $w_t$ is constant within each sampled trajectory and is computed by comparing trajectories sampled from the same initial instance. In multi-turn settings, this yields coarser credit assignment because action-specific returns are not distinguished. REINFORCE++ (Hu et al., 2025) replaces raw returns with batch-normalized Monte Carlo advantages, with $w_t$ defined as $A_t$, where $A_t = \frac{G_t-\mu_G}{\sigma_G+\epsilon}$. Here, $\mu_G$ and $\sigma_G$ are the mean and standard deviation of Monte Carlo returns in the current batch. This normalization reduces gradient-scale instability while retaining the critic-free Monte Carlo return structure.

In our setting, we use REINFORCE++ as a policy optimization algorithm, where the policy is conditioned on multimodal observations and optimized for multi-turn interaction. Given an on-policy trajectory $\tau$, we compute the normalized Monte Carlo return $A_t$, treat each transition as one training unit, and optimize the objective above accordingly.

## 4 Training VLM Agents with EnvHint

To address exploration stagnation under sparse rewards and enable behaviors discovered with hints to transfer back to the original base environment, we propose **EnvHint**, a training framework that consists of three components, as summarized in Figure 2: hinted environment construction, hint-guided rollout collection, and backward hint annealing. The following three subsections describe these components in turn.

### 4.1 Hinted Environment Construction

Training directly on the original base environments remains difficult in our setting because early in training, the policy often fails to discover correct trajectories, and sparse rewards provide only limited effective learning signal when exploration fails. To improve learnability, we modify the environment implementation to construct paired hinted and base observation views of the same underlying task instance online as the environment state evolves. Specifically, the base environment provides the original observation view, while the hinted environment augments it with extra task-relevant guidance generated from the current environment state. Let $o_{b,t} = (I_{b,t}, x_{b,t})$ denote the base observation at step $t$, where $I_{b,t}$ and $x_{b,t}$ denote its visual and tex-

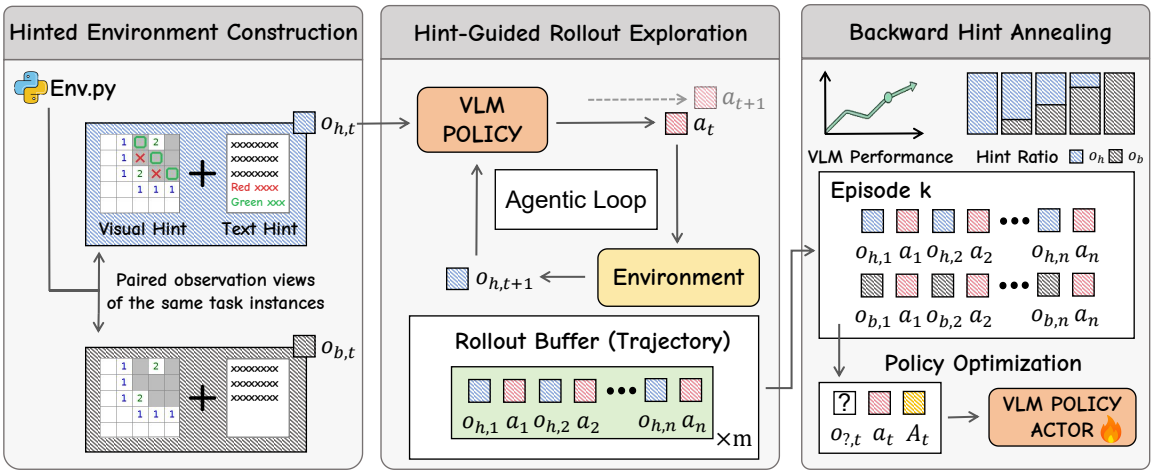



Figure 2: **Overview of EnvHint.** Hinted environment construction produces paired hinted and base observation views for the same task instance. Hint-guided rollout collection uses hinted interaction to improve rollout discovery. Backward hint annealing then progressively shifts training back to base observations before policy optimization.

tual components, respectively. We further construct the hinted observation $o_{h,t} = (I_{h,t}, x_{h,t})$, where hints may modify the textual channel, the visual channel, or both. As a result, the two views share the same action space, correct answer, environment feedback, and state-transition dynamics, differing only in their observations. This makes it possible for behaviors discovered with hints to transfer back to the corresponding base observations: in principle, a useful action sequence discovered with hints can remain valid after the extra observation scaffolds are removed.

### 4.2 Hint-Guided Rollout Exploration

Given the paired hint and base environments, the next question is how to use the hinted view to improve early rollout exploration and trajectory discovery before the policy can act reliably under the original base observations. Concretely, the policy interacts with the hint environment to collect multi-turn trajectories, which are then stored in a rollout buffer for subsequent training. This stage increases the probability of sampling useful behaviors early in training under an easier observation view of the same task. When informative, hints make it easier for the policy to reach useful states, obtain non-zero rewards, and discover successful action patterns and trajectories under sparse supervision. In this sense, hint-guided rollout exploration improves learnability by producing more informative learning signals during the early stage of training.

Hint-guided rollout exploration alone, however, is not sufficient. Training under the hinted observation distribution $p(a_t \mid o_{h,t})$ does not guarantee strong behavior under the corresponding base observation distribution $p(a_t \mid o_{b,t})$. Because the hint and base environments differ only in observation, the hinted view should be understood as training-time observation scaffolding rather than a shortcut to the task itself. At test time, however, the policy must act under the original base observations without access to hints. If training never returns to base observations, the policy may over-rely on hints rather than internalize the behavior in the original observation space, which directly motivates the backward annealing process introduced next.

### 4.3 Backward Hint Annealing

Hint-guided rollout exploration helps the policy discover useful behaviors under scaffolded observations, but success in the hinted view does not by itself guarantee strong behavior under the original base observations. Since the final target remains the base observation space, we therefore use a backward annealing curriculum to gradually transfer training from hinted observations back to their corresponding base observations.

This transfer is well defined in our setting because the hint and base environments are paired views of the same underlying task instance (Section 4.1). We instantiate this process at the environment level. Let $\mathcal{E}_{\text{hint}}$ denote the hint-environment distribution and $\mathcal{E}_{\text{base}}$ denote the corresponding base-environment distribution. At curriculum stage $k$, we maintain a base-environment ratio $\rho_k \in [0, 1]$ and define

$$\mathcal{E}^{(k)} = (1 - \rho_k)\, \mathcal{E}_{\text{hint}} + \rho_k\, \mathcal{E}_{\text{base}}. \qquad (2)$$

| Model | Minesweeper | | | | FrozenLake | | | | Hallway | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Base | Hard | Easy | Hint | Base | Hard | Easy | Hint | Base | Hint | |
| *Proprietary VLMs* | | | | | | | | | | | |
| GPT-5.4-mini | 0.13 | 0.03 | 0.18 | 1.00 | 0.27 | 0.03 | 0.40 | 0.86 | 0.47 | 0.60 | 0.40 |
| Gemini-3-Pro-Preview | 0.64 | 0.22 | 0.94 | 1.00 | 0.70 | 0.25 | 0.81 | 0.85 | 0.38 | 0.58 | 0.64 |
| Gemini-3-Flash-Preview | 0.56 | 0.22 | 0.87 | 1.00 | 0.56 | 0.09 | 0.74 | 1.00 | 0.55 | 0.60 | 0.62 |
| Gemini-2.5-Flash | 0.18 | 0.03 | 0.16 | 0.99 | 0.02 | 0.03 | 0.37 | 0.87 | 0.05 | 0.61 | 0.33 |
| *Open-source VLMs* | | | | | | | | | | | |
| Qwen2.5-VL-7B-Instruct | 0.06 | 0.00 | 0.00 | 0.87 | 0.00 | 0.00 | 0.00 | 0.65 | 0.04 | 0.57 | 0.22 |
| Qwen2.5-VL-32B-Instruct | 0.06 | 0.00 | 0.00 | 0.98 | 0.03 | 0.00 | 0.20 | 0.40 | 0.01 | 0.59 | 0.23 |
| Qwen2.5-VL-72B-Instruct | 0.10 | 0.01 | 0.14 | 1.00 | 0.13 | 0.03 | 0.30 | 0.56 | 0.38 | 0.61 | 0.33 |
| Qwen3-VL-32B-Instruct | 0.08 | 0.00 | 0.17 | 0.93 | 0.00 | 0.00 | 0.00 | 0.46 | 0.34 | 0.61 | 0.26 |
| Qwen3-VL-30B-A3B-Instruct | 0.07 | 0.00 | 0.15 | 0.92 | 0.00 | 0.00 | 0.00 | 0.45 | 0.01 | 0.61 | 0.22 |
| Qwen3-VL-235B-A22B-Instruct | 0.16 | 0.01 | 0.35 | 1.00 | 0.17 | 0.03 | 0.26 | 0.45 | 0.31 | 0.59 | 0.33 |
| *Our Models* | | | | | | | | | | | |
| Qwen3-VL-8B-Instruct (Backbone) | 0.03 | 0.01 | 0.31 | 0.99 | 0.12 | 0.00 | 0.23 | 0.69 | 0.06 | 0.60 | 0.30 |
| *w/* GRPO | 0.10 | 0.01 | 0.18 | 0.94 | 0.00 | 0.00 | 0.00 | 0.20 | 0.24 | 0.36 | $0.20_{\downarrow 0.10}$ |
| *w/* REINFORCE++ | 0.04 | 0.00 | 0.18 | 0.97 | 0.14 | 0.03 | 0.35 | 0.40 | 0.24 | 0.37 | $0.27_{\downarrow 0.03}$ |
| *w/* **EnvHint (Ours)** | 0.56 | 0.17 | 0.46 | 0.95 | 0.41 | 0.41 | 0.70 | 0.81 | 0.60 | 0.61 | $0.57_{\uparrow 0.27}$ |

Table 1: **Main results on agentic VLM environments.** Minesweeper and FrozenLake report task accuracy under four evaluation settings: the original base environment, hard split, easy split, and hint-augmented environment. Hallway reports task accuracy under the base environment and hint-augmented environment.

Training starts from $\rho_0 = 0$, so the policy first learns entirely under hinted observations. As training progresses, $\rho_k$ is increased, gradually shifting training toward base observations. The rollout buffer stores episodes collected under hinted observations. In our implementation, backward annealing operates on these buffered episodes before actor optimization by replacing the observations of a $\rho_k$ fraction of episodes with their corresponding base observations. After annealing, we compute the stepwise advantages $A_t$ defined in Section 3 for the episodes, and each transition-level sample is then passed to the actor for policy optimization.

The schedule is performance-triggered. Let $m_i$ denote the mean rollout reward at rollout round $i$. We maintain a moving window of length $W$ and compute

$$\bar{m}_i = \frac{1}{W} \sum_{j=i-W+1}^{i} m_j. \tag{3}$$

Whenever $\bar{m}_i \geq \eta$, we count one successful curriculum check; otherwise the counter is reset. After $P$ consecutive successful checks, we promote the curriculum by setting $\rho_{k+1} = \min(\rho_k + \Delta\rho, \rho_{\max})$, where $\rho_{\max}$ is the maximum base ratio, $\eta$ is the promotion threshold, $P$ is the patience parameter, and $\Delta\rho$ is the base-ratio increment. After each promotion, the running window is cleared and the process repeats at the new stage. In this way, transfer back to base observations is triggered only after performance under the current hint-to-base observation ratio has become sufficiently stable, rather than after a single noisy improvement.

Backward annealing serves as the transfer mechanism that converts hint-guided exploration gains into capability under the base observations. Early stages focus on alleviating exploration stagnation through privileged multimodal guidance, while later stages focus on robustness under the original base observations.

# 5 Experiment

## 5.1 Experiment Setting

**Benchmarks.** We evaluate on three Gym-V-based VLM agent environments: Minesweeper (MS), FrozenLake (FL), and Hallway (HW) (Meng et al., 2026). For each environment, we construct paired base and hinted variants. For MS and FL, we derive easy and hard evaluation splits from the base-environment parameterization; HW does not expose an analogous difficulty control, so it is evaluated only in the base and hint settings. Throughout training, we use only the paired base and hint environments, while the easy and hard splits are used only for evaluation. Training and evaluation are conducted on disjoint environment instances and initial states, and each reported success rate is computed over 100 evaluation seeds.

**Baselines.** We compare against a broad set of proprietary and open-source VLM baselines listed in Table 1. Proprietary baselines include GPT-5.4-mini, Gemini-3-Pro-Preview, Gemini-3-Flash-

Preview, and Gemini-2.5-Flash. Open-source baselines include Qwen-VL variants (Qwen2.5-VL and Qwen3-VL series). Within our controlled backbone family, we report the Qwen3-8B-VL together with GRPO, REINFORCE++, and EnvHint variants, so that gains can be attributed to training strategy rather than backbone replacement. Additional training details are provided in Appendix A.

### 5.2 Main Results

Table 1 summarizes the main results. We focus on the original no-hint base environments and the harder held-out no-hint splits, since these settings most directly test whether behaviors discovered under hinted training transfer back to the original base environments rather than remaining tied to privileged hinted observations. The easy, base, and hard splits also follow the expected difficulty ordering, with performance generally highest on easy variants, intermediate on base environments, and lowest on hard variants.

Within the controlled backbone family, EnvHint yields by far the strongest performance on the no-hint evaluations. Adding GRPO or REINFORCE++ alone produces only limited gains on these settings, whereas EnvHint substantially improves all three base environments and both hard splits, reaching an overall average of 0.57 versus 0.30 for the backbone, 0.20 for GRPO, and 0.27 for REINFORCE++ alone. The difference is especially clear on the harder no-hint evaluations: EnvHint reaches 0.17 on Minesweeper-hard and 0.41 on FrozenLake-hard, compared with 0.01 and 0.00 for GRPO, and 0.00 and 0.03 for REINFORCE++ alone. EnvHint is also competitive against a broad set of open-source and proprietary VLM baselines, achieving the strongest Hallway-base result in the table at 0.60 and performing strongly on both FrozenLake-base and Minesweeper-hard. Overall, these results show that the value of EnvHint lies not merely in making hinted training easier, but in turning hinted exploration into useful gains on the final no-hint base environments.

### 5.3 Ablation Studies

We ablate the main components of EnvHint in Table 2. **REINFORCE++ only** means training directly on the original base environments with the RL algorithm alone. **Hint-only training** means training remains in the hinted observations without transfer back to the original no-hint setting. **Immediate base transfer** means all hinted observations in buffered episodes are directly replaced with their base views before policy optimization, without gradual curriculum.

| Variant | Base Avg. | Hard Avg. |
|---|---|---|
| Backbone | 0.070 | 0.005 |
| REINFORCE++ only | $0.140_{\uparrow 0.070}$ | $0.015_{\uparrow 0.010}$ |
| Hint-only training | $0.067_{\downarrow 0.003}$ | $0.020_{\uparrow 0.015}$ |
| Immediate base transfer | $0.370_{\uparrow 0.300}$ | $0.175_{\uparrow 0.170}$ |
| EnvHint | $\mathbf{0.523}_{\uparrow 0.453}$ | $\mathbf{0.290}_{\uparrow 0.285}$ |

Table 2: **Ablation of the core components of EnvHint. Base Avg.** averages the base environments of MS, FL, and HW. **Hard Avg.** averages the harder held-out no-hint splits of MS and FL.

Table 2 shows that REINFORCE++ alone is not sufficient. Although *REINFORCE++ only* improves the Base Avg. from 0.070 to 0.140, its Hard Avg. remains extremely low at 0.015. This suggests that, under sparse rewards, simply optimizing the policy with REINFORCE++ is still not enough when the base model rarely samples useful trajectories early in training.

*Hint-only training (no base transfer)* reaches only 0.067 on Base Avg. and 0.020 on Hard Avg., both close to backbone-level performance. This suggests that hint-augmented training by itself is not enough: if the policy is never transferred back to the original no-hint setting, the learned behavior does not reliably carry over to either the original base environments or the harder held-out variants.

*Immediate base transfer (no curriculum)* still obtains 0.370 on Base Avg., but its Hard Avg. drops to 0.175, remaining far below the full method's 0.290. This indicates that immediately replacing the hint-augmented observations with the original base observations is less effective than gradually annealing from hint-augmented training back to the original no-hint setting. A likely reason is that curriculum does more than control difficulty: it also stabilizes distribution shift. Rather than forcing an abrupt replacement of the training observations from the beginning, the gradual schedule allows the policy to adapt to the original observation distribution in a smoother and more stable way, which in turn improves transfer to harder held-out variants.

Overall, the best results come from combining RL optimization with hint-guided exploration and a gradual curriculum that transfers training back to the no-hint base environment.

## 6 Discussion

### 6.1 RL Algorithm Choice

We first discuss how the choice of RL algorithm affects optimization dynamics in a representative Minesweeper setting. Figure 3 compares two EnvHint variants with different RL algorithms, denoted as EnvHint-GRPO and EnvHint-REINFORCE++. We align the curves by evaluation rounds, with one evaluation performed every 10 rollout steps for a consistent comparison.

EnvHint-REINFORCE++ produces useful gains on the no-hint base environment much earlier than EnvHint-GRPO. By round 20, it already reaches 0.23, while EnvHint-GRPO remains at 0.07; by rounds 30 and 40, the gap further widens to 0.39 vs. 0.10 and 0.65 vs. 0.05. Even at round 100, EnvHint-REINFORCE++ reaches 0.60, whereas EnvHint-GRPO is still at 0.09. The curves therefore show not only a final performance gap, but also a clear difference in how quickly scaffolded training progress turns into visible gains on the original no-hint environment.

At the same time, EnvHint-GRPO does not completely fail to learn: under a longer budget, it eventually reaches 0.32 around round 180. However, that progress emerges substantially later and remains well below the performance reached by EnvHint-REINFORCE++ much earlier in training. This difference is consistent with the RL algorithm design. REINFORCE++ preserves discounted return structure and transition-level credit assignment, whereas GRPO relies on shared trajectory-level relative signals that can become weak when reward variance is small. In this long-horizon sparse-reward setting, EnvHint-REINFORCE++ is therefore more effective at converting scaffolded training progress into stable improvement on the final no-hint base environment.

### 6.2 Hint Intensity Analysis

We next analyze how hint intensity shapes learning and transfer within EnvHint. Figure 4 summarizes this effect from three complementary views: evaluation on the original base environment, annealing progress measured by the base-observation replacement ratio, and mean rollout reward during training (with rollouts sampled from the hint environment).

Figure 4(a) shows that hint intensity directly affects how quickly performance emerges after transfer back to the original base environment. Strong Hint improves much earlier, whereas Weak Hint rises more slowly and reaches comparable performance only later. By contrast, No Hint remains near failure throughout training. The key difference is therefore not simply whether guidance is present, but whether it is informative enough to support useful behaviors that can later be retained under the original no-hint setting.

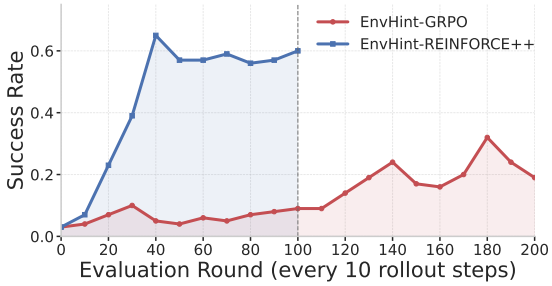



Figure 3: **Minesweeper-base evaluation curves** for EnvHint-GRPO and EnvHint-REINFORCE++.

Figure 4(b) shows that more informative hints also lead to faster annealing progress: once the policy can reliably succeed under the hinted view, the curriculum more quickly replaces hinted observations with their corresponding base-environment views. Weak Hint still supports annealing, but much later, indicating that weaker guidance slows the transition back to base observations rather than preventing transfer entirely.

Figure 4(c) helps explain these differences. More informative hints yield much higher rollout reward early in training, indicating that they provide a more useful scaffold in the hinted phase itself. By contrast, No Hint remains near zero throughout, showing that sparse-reward learning is difficult to even start without additional guidance. The weak hint still improves over the no-hint baseline, but provides a noticeably weaker training signal than Strong Hint, which helps explain why it also lags behind in annealing progress and base-environment performance. Taken together, these results suggest that the value of a hint lies not simply in making the task easier, but in providing sufficiently informative guidance for discovering behaviors that can later be transferred back to the original base observation space.

### 6.3 Hint Modality Analysis

We next analyze how hint modality affects transfer from hint-augmented training to the original no-hint setting and then to harder held-out variants. Figure 5 visualizes the gains of different hint modalities over the backbone on five no-hint evalu-

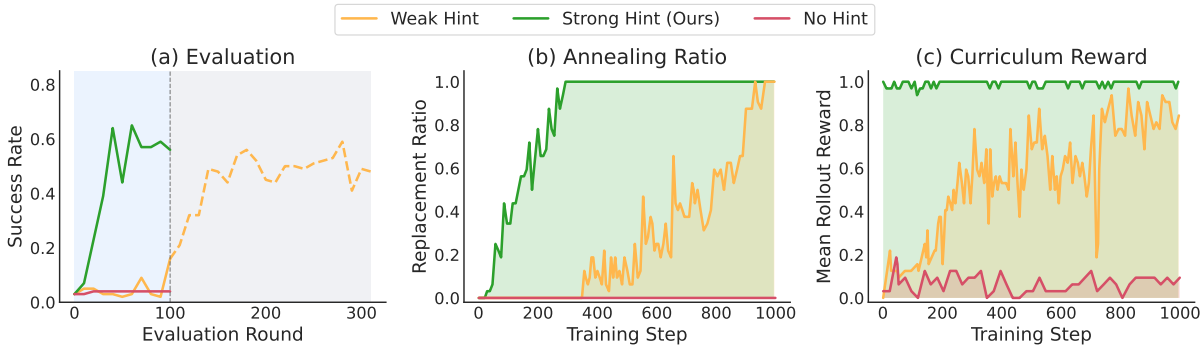



Figure 4: **Hint intensity analysis on Minesweeper.** Strong Hint (Ours) denotes our default hint setting, Weak Hint denotes a less informative hint variant, and No Hint denotes the REINFORCE++-only baseline without observation-level guidance.

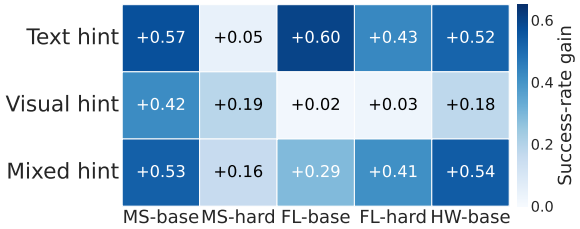



Figure 5: **Heatmap of hint-modality gains over the backbone** on no-hint evaluations. Values are absolute success-rate improvements over the backbone.

ations: MS-base, MS-hard, FL-base, FL-hard, and HW-base. These settings capture both performance on the original base environments and generalization to harder unseen no-hint variants.

Textual hints provide the strongest overall transfer back to the original base environments. They produce the largest average gain across these five settings, with especially strong improvements on FL-base and FL-hard. This suggests that, in rule-heavy multimodal agentic tasks, textual guidance may be easier for the policy to internalize and reuse after the scaffold is removed. Mixed hints are more balanced across environments. Although they do not always exceed text hints on every individual setting, they improve all five no-hint evaluations substantially and remain competitive on both the base environments and the harder held-out variants. This makes them a robust choice when training must cover heterogeneous environments with different sources of difficulty. By contrast, visual hints alone are less consistent in our current settings. They still improve over the backbone in several cases, but the gains are much smaller overall and are especially weak on FL. This suggests that visual-only hinting depends more strongly on environment structure and hint design, with its effectiveness largely determined by whether the injected visual cue is aligned with the task-relevant structure that actually drives decision making during interaction.

Taken together, the heatmap suggests a clear tradeoff between text and mixed hints: text hints are strongest for direct transfer back to the original no-hint setting, whereas mixed hints are more balanced across environments and harder held-out variants. At the same time, mixed hints do not uniformly outperform text hints, and visual hints show the largest variation across environments, suggesting that multimodal hinting does not automatically yield additive gains; combinations help only when the two modalities provide complementary rather than redundant guidance. Overall, these results show that hint modality affects not only the efficiency of early rollout exploration, but also how much learned behavior is retained after transfer back to the original no-hint setting.

## 7 Conclusion

We addressed exploration stagnation in sparse-reward VLM agent training with EnvHint, a progressive framework built on hinted environment construction, hint-guided rollout exploration, and backward hint annealing. EnvHint turns hinted interaction into a practical and effective training scaffold, substantially improving rollout discovery while gradually transferring learned behavior back to the original no-hint setting. Across Minesweeper, FrozenLake, and Hallway, this training process consistently improves performance on the original base environments and harder held-out variants. More broadly, these results suggest that hinted environments can provide a simple and effective path for training VLM agents under sparse rewards in challenging multi-turn environments.

## Limitations

Our current study is limited to sparse-reward multi-turn VLM agent environments in which task-relevant hints can be instantiated as paired observation views while preserving the underlying task dynamics, so whether the hinted-environment paradigm extends equally well to broader environment classes remains to be further validated. Moreover, although prior work has studied textual hints more extensively and our framework already supports textual, visual, and mixed hints, a more systematic understanding of multimodal hint design and its effects on agent learning is still lacking.

## References


Karl Johan Åström. 1965. Optimal control of markov processes with incomplete state information i. *Journal of mathematical analysis and applications*, 10:174–205.

Thomas Carta, Subhajit Chaudhury, Kartik Talamadupula, and Michiaki Tatsubori. 2020. Visual-hints: A visual-lingual environment for multimodal reinforcement learning. *CoRR*, abs/2010.13839.

Liang Chen, Hongcheng Gao, Tianyu Liu, Zhiqi Huang, Flood Sung, Xinyu Zhou, Yuxin Wu, and Baobao Chang. 2025. G1: bootstrapping perception and reasoning abilities of vision-language model via reinforcement learning. *CoRR*, abs/2505.13426.

Zhili Cheng, Yuge Tu, Ran Li, Shiqi Dai, Jinyi Hu, Shengding Hu, Jiahao Li, Yang Shi, Tianyu Yu, Weize Chen, Lei Shi, and Maosong Sun. 2025. Embodiedeval: Evaluate multimodal llms as embodied agents. *CoRR*, abs/2501.11858.

Yichen Feng, Zhangchen Xu, Fengqing Jiang, Yuetai Li, Bhaskar Ramasubramanian, Luyao Niu, Bill Yuchen Lin, and Radha Poovendran. 2025. Visualsphinx: Large-scale synthetic vision logic puzzles for RL. *CoRR*, abs/2505.23977.

Longxi Gao, Li Zhang, and Mengwei Xu. 2025. Uishift: Enhancing vlm-based GUI agents through self-supervised reinforcement learning. *CoRR*, abs/2505.12493.

Jihao Gu, Qihang Ai, Yingyao Wang, Pi Bu, Jingxuan Xing, Zekun Zhu, Wei Jiang, Ziming Wang, Yingxiu Zhao, Ming-Liang Zhang, Jun Song, Yuning Jiang, and Bo Zheng. 2025. Mobile-r1: Towards interactive reinforcement learning for vlm-based mobile agent via task-level rewards. *CoRR*, abs/2506.20332.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. 2025. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization. *arXiv preprint arXiv:2501.03262*.

Qihan Huang, Weilong Dai, Jinlong Liu, Wanggui He, Hao Jiang, Mingli Song, Jingyuan Chen, Chang Yao, and Jie Song. 2025. Boosting MLLM reasoning with text-debiased hint-grpo. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 4848–4857. IEEE.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. 2026. Reinforcement learning via self-distillation. *CoRR*, abs/2601.20802.

Pengxiang Li, Zechen Hu, Zirui Shang, Jingrong Wu, Yang Liu, Hui Liu, Zhi Gao, Chenrui Shi, Bofei Zhang, Zihao Zhang, Xiaochuan Shi, Zedong Yu, Yuwei Wu, Xinxiao Wu, Yunde Jia, Liuyu Xiang, Zhaofeng He, and Qing Li. 2025a. Efficient multi-turn RL for GUI agents via decoupled training and adaptive data curation. *CoRR*, abs/2509.23866.

Ziheng Li, Zexu Sun, Jinman Zhao, Erxue Min, Yongcheng Zeng, Hui Wu, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Xu Chen, and Zhi-Hong Deng. 2025b. Staying in the sweet spot: Responsive reasoning evolution via capability-adaptive hint scaffolding. *CoRR*, abs/2509.06923.

Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, and Jiang Bian. 2026. Self-hinting language models enhance reinforcement learning. *CoRR*, abs/2602.03143.

Meng Lu, Ran Xu, Yi Fang, Wenxuan Zhang, Yue Yu, Gaurav Srivastava, Yuchen Zhuang, Mohamed Elhoseiny, Charles Fleming, Carl Yang, Zhengzhong Tu, Yang Xie, Guanghua Xiao, Hanrui Wang, Di Jin, Wenqi Shi, and Xuan Wang. 2025. Scaling agentic reinforcement learning for tool-integrated reasoning in vlms. *CoRR*, abs/2511.19773.

Fanqing Meng, Lingxiao Du, Jiawei Gu, Jiaqi Liao, Linjie Li, Zijian Wu, Xiangyan Liu, Ziqi Zhao, Mengkang Hu, Yue Zhang, Zichen Liu, Jiaheng Zhang, and Michael Qizhe Shieh. 2026. Gym-v: A unified vision environment system for agentic vision research. *CoRR*, abs/2603.15432.

Zhenyu Pan and Han Liu. 2025. Metaspatial: Reinforcing 3d spatial reasoning in vlms for the metaverse. *CoRR*, abs/2503.18470.

Deng Qiyuan, Kehai Chen, Min Zhang, and Zhongwen Xu. 2026. Hipo: Self-hint policy optimization for rlvr. In *The Fourteenth International Conference on Learning Representations*.

Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. 2026. POPE: learning to reason on hard problems via privileged on-policy exploration. *CoRR*, abs/2601.18779.

Ramprasaath Ramasamy Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry P. Heck,

Dhruv Batra, and Devi Parikh. 2019. Taking a HINT: leveraging explanations to make vision and language models more grounded. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2591–2600. IEEE.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Chengshuai Shi, Wenzhe Li, Xinran Liang, Yizhou Lu, Wenjia Yang, Ruirong Feng, Seth Karten, Ziran Yang, Zihan Ding, Gabriel Sarch, and 1 others. 2026. Odysseus: Scaling vlms to 100+ turn decision-making in games via reinforcement learning. *arXiv preprint arXiv:2605.00347*.

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

Reuben Tan, Baolin Peng, Zhengyuan Yang, Hao Cheng, Oier Mees, Theodore Zhao, Andrea Tupini, Isar Meijier, Qianhui Wu, Yuncong Yang, Lars Liden, Yu Gu, Sheng Zhang, Xiaodong Liu, Lijuan Wang, Marc Pollefeys, Yong Jae Lee, and Jianfeng Gao. 2025. Multimodal reinforcement learning with agentic verifier for AI agents. *CoRR*, abs/2512.03438.

Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, Lijuan Wang, Ranjay Krishna, Jiajun Wu, Li Fei-Fei, Yejin Choi, and Manling Li. 2025a. VAGEN: reinforcing world model reasoning for multi-turn VLM agents. *CoRR*, abs/2510.16907.

Peiyao Wang and Haibin Ling. 2025. SVQA-R1: reinforcing spatial reasoning in mllms via view-consistent reward optimization. *CoRR*, abs/2506.01371.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. 2025b. RAGEN: understanding self-evolution in LLM agents via multi-turn reinforcement learning. *CoRR*, abs/2504.20073.

Zirui Wang, Junyi Zhang, Jiaxin Ge, Long Lian, Letian Fu, Lisa Dunlap, Ken Goldberg, Xudong Wang, Ion Stoica, David M. Chan, Sewon Min, and Joseph E. Gonzalez. 2026. Visgym: Diverse, customizable, scalable environments for multimodal agents. *CoRR*, abs/2601.16973.

Tong Wei, Yijun Yang, Junliang Xing, Yuanchun Shi, Zongqing Lu, and Deheng Ye. 2025. GTR: guided thought reinforcement prevents thought collapse in rl-based VLM agent training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 18855–18865. IEEE.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

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

Qingyuan Wu, Jianheng Liu, Jianye Hao, Jun Wang, and Kun Shao. 2025b. VSC-RL: advancing autonomous vision-language agents with variational subgoal-conditioned reinforcement learning. *CoRR*, abs/2502.07949.

Yu Xia, Canwen Xu, Zhewei Yao, Julian J. McAuley, and Yuxiong He. 2026. Learning to hint for reinforcement learning. *CoRR*, abs/2604.00698.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yifan Xu, Xiao Liu, Xinghan Liu, Jiaqi Fu, Hanchen Zhang, Bohao Jing, Shudan Zhang, Yuting Wang, Wenyi Zhao, and Yuxiao Dong. 2025. Mobilerl: Online agentic reinforcement learning for mobile GUI agents. *CoRR*, abs/2509.18119.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *CoRR*, abs/2504.14945.

Feng Zhang, Zezhong Tan, Xinhong Ma, Ziqiang Dong, Xi Leng, Jianfei Zhao, Xin Sun, and Yang Yang. 2025a. Adhint: Adaptive hints with difficulty priors for reinforcement learning. *CoRR*, abs/2512.13095.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. 2025b. Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. *CoRR*, abs/2507.02841.

Ruiyang Zhang, Qianguo Sun, Chao Song, Yiyan Qi, and Zhedong Zheng. 2026. Vsearcher: Long-horizon multimodal search agent via reinforcement learning. *CoRR*, abs/2603.02795.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. 2025c. Scaf-grpo: Scaffolded group relative policy optimization for enhancing llm reasoning. *arXiv preprint arXiv:2510.19807*.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. 2025. slime: An llm post-training framework for rl scaling. https://github.com/THUDM/slime. GitHub repository. Corresponding author: Xin Lv.

# A Experiment Details

## A.1 Training Algorithm

Algorithm 1 summarizes the full training procedure of EnvHint, including hinted rollout collection, actor-side backward hint annealing, and performance-triggered curriculum promotion.

**Algorithm 1** EnvHint Training Procedure

**Require:** Policy $\pi_\theta$, hint environment $\mathcal{E}_{\text{hint}}$, base environment $\mathcal{E}_{\text{base}}$, rollout batch size $N$, promotion threshold $\eta$, window size $W$, patience $P$, base-ratio increment $\Delta\rho$, maximum base ratio $\rho_{\max}$

Initialize base-environment ratio $\rho \leftarrow 0$, moving window $\mathcal{W} \leftarrow \emptyset$, and promotion counter $c \leftarrow 0$
**while** training budget not exhausted **do**
  Collect $N$ on-policy trajectories $\{\tau_{\text{hint}}^{(n)}\}_{n=1}^{N}$ in $\mathcal{E}_{\text{hint}}$ and store the hinted rollout episodes
  **for** each trajectory $\tau_{\text{hint}}^{(n)}$ **do**
    With probability $\rho$, route its actor-side observation stream through the paired base-observation view of the same task instance
    Build the annealed training episode $\tilde{\tau}^{(n)}$ from the resulting actor-side stream
  **end for**
  **for** each annealed episode $\tilde{\tau}^{(n)}$ **do**
    **for** each transition $t$ **do**
      Compute discounted return $G_t^{(n)} = \sum_{k=t}^{T(n)-1} \gamma^{k-t} r_k^{(n)}$
    **end for**
  **end for**
  Normalize returns over the current transition batch to obtain advantages $A_t^{(n)}$
  Update $\pi_\theta$ with the REINFORCE++ objective on the annealed transition-level samples
  Compute the mean rollout reward $m_i$ of the current rollout round and append it to $\mathcal{W}$
  Compute the moving-window reward $\bar{m}_i$ from the current window of up to $W$ rollout rounds
  **if** $\bar{m}_i \geq \eta$ **then**
    Increment promotion counter $c \leftarrow c + 1$
  **else**
    Reset promotion counter $c \leftarrow 0$
  **end if**
  **if** $c \geq P$ **then**
    Promote curriculum: $\rho \leftarrow \min(\rho + \Delta\rho, \rho_{\max})$
    Clear $\mathcal{W}$ and reset promotion counter $c \leftarrow 0$
  **end if**
**end while**

## A.2 Implementation Details

We build EnvHint on top of the REINFORCE++ implementation provided by `slime` (Zhu et al., 2025). All main experiments use the Megatron backend on 4 NVIDIA A100 GPUs.

## A.3 Hyperparameters

Table 3 summarizes the main hyperparameters used in our experiments.

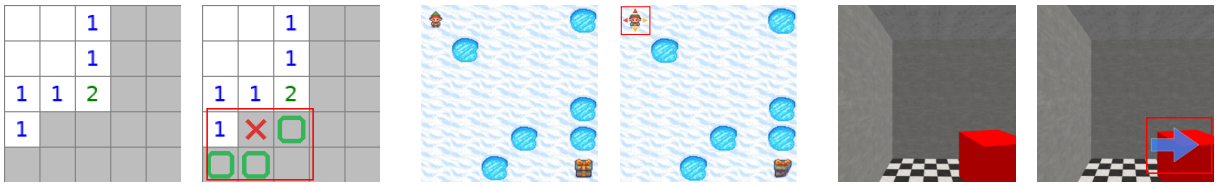
| Hyperparameter | Value |
|---|---|
| Training temperature | 1.0 |
| Evaluation temperature | 0 |
| Discount factor $\gamma$ | 0.8 |
| Initial base-environment ratio $\rho_0$ | 0.0 |
| Base-ratio increment $\Delta\rho$ | 0.1 |
| Maximum base ratio $\rho_{\max}$ | 1.0 |
| Promotion threshold $\eta$ | 0.60 |
| Window size $W$ | 5 |
| Patience $P$ | 3 |

Table 3: Main hyperparameters for EnvHint.

# B Hint Details

We further instantiate the hinted-environment formulation in Section 4.1 at the environment-implementation level. All hinted environments modify only the observation returned at the current state. In practice, the base and hinted variants share the same underlying environment and differ only through additional guidance injected into the rendered image or observation text. The implementation workflow follows a common pattern across environments: at each step, the environment first updates the current task state and produces the base observation, then applies a state-derived transformation that injects task-relevant guidance into the visual channel, the textual channel, or both.

Figure 6 illustrates base and hinted observations in the three training environments. In Minesweeper, the hinted observation is constructed from the currently revealed clue numbers and local unrevealed neighbors, with inferred safe-cell and mine-cell candidates rendered as green outlines and red Xs and mirrored in text. In FrozenLake, the hinted observation is constructed from the current player position, goal position, and hole layout, with directional arrows distinguishing unsafe moves, safe moves, and preferred safe moves that reduce distance to the goal. In Hallway, the hinted observation is constructed from the current first-person pose and target position, with the relative target angle converted into lightweight observation guidance: the visual channel adds an arrow for directional guidance, while the textual channel provides a cue derived from the same geometry.

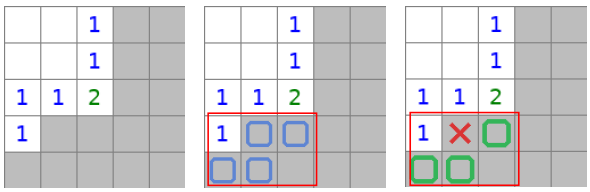


(a) Minesweeper (b) FrozenLake (c) Hallway

Figure 6: **Base and hinted observation examples across environments.** Within each pair, the left is the original base observation and the right is the corresponding hinted observation. In Minesweeper, red Xs denote inferred mine cells and green outlines denote candidate safe cells. In FrozenLake, gold/green/red arrows denote preferred safe / safe / unsafe moves. In Hallway, the translucent arrow provides guidance in the first-person view.

## C Weak Hint for Intensity Analysis

For Section 6.2, we study the effect of hint intensity by introducing a weak hint setting. This weak hint highlights only one small local frontier region with blue outlines and lists those cells in text. The highlighted region is selected from unrevealed cells adjacent to revealed numbered cells and serves only as an attention-directing cue for local reasoning, rather than directly providing inferred safe-cell or mine-cell information as in the strong hint used in the main experiments. Figure 7 shows a concrete comparison between base, weak-hint, and strong-hint observations on the same board state.



(a) Base (b) Weak Hint (c) Strong Hint

Figure 7: **Comparison of weak and strong hints in Minesweeper.** The weak frontier-focus hint highlights only a small blue frontier region, whereas the strong hint directly marks candidate safe cells and mine cells.

## D REINFORCE++-Only Training Curves

Figure 8 shows the training curves of REINFORCE++ only on the three base environments. Across Minesweeper, FrozenLake, and Hallway, the mean rollout reward remains low for a long portion of training and improves only slowly, reflecting the exploration stagnation problem described in the main text under sparse rewards.

## E Cross-Environment Comparison of RL Algorithm Choices

Figure 9 provides cross-environment support for the discussion in Section 6.1 by showing final environment-specific results for the **backbone**, **EnvHint-GRPO**, and **EnvHint-REINFORCE++** across the three environments. A consistent pattern is that EnvHint-REINFORCE++ yields stronger final performance than the backbone and usually also improves over EnvHint-GRPO, with the clearest gains appearing on the no-hint base and harder held-out evaluations. This provides a cross-environment view of the same trend observed: compared with GRPO, the REINFORCE++ variant converts hint-guided training into stronger final target-setting performance more reliably.

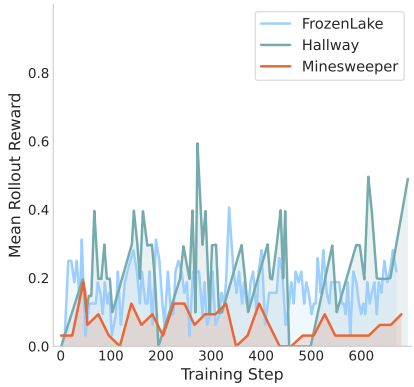
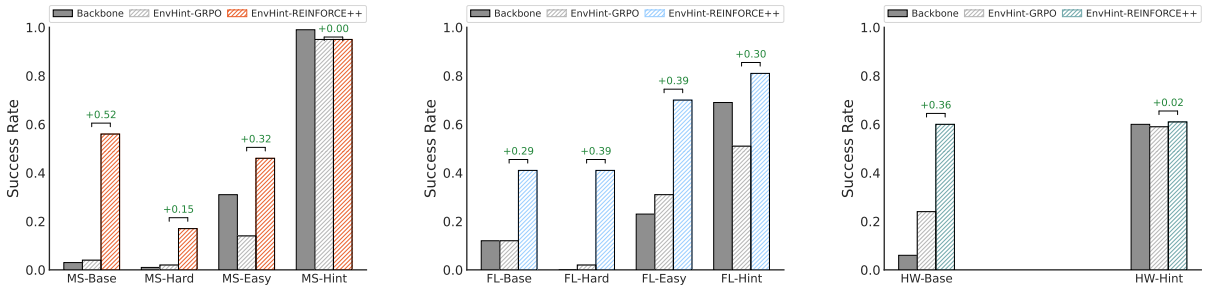



Figure 8: **REINFORCE++-only training curves on the base environments.** Mean rollout reward during training on Minesweeper, FrozenLake, and Hallway.

## F Case Study

Figures 10, 11, and 12 provide qualitative example trajectories across the three environments. For each environment, we compare four settings in the same figure: **Backbone-Base**, **EnvHint-Base**, **Backbone-Hint**, and **EnvHint-Hint**, which denote the backbone model or the EnvHint-trained model interacting under the base or hinted observation view, respectively. To keep multi-turn trajectories readable, we visualize a small set of key frames sampled from the interaction process while preserving the start and final states.




(a) Minesweeper (b) FrozenLake (c) Hallway

Figure 9: **Cross-environment comparison of RL algorithm choices.** Final performance on each evaluation setting.

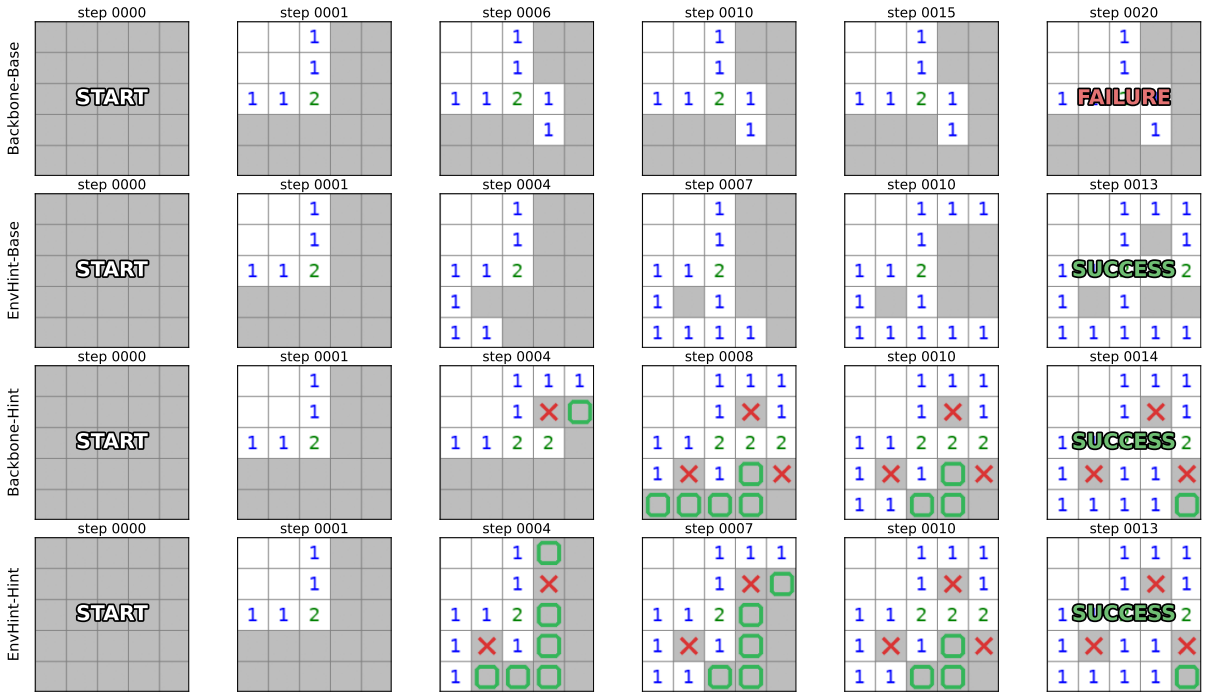



Figure 10: Minesweeper case study.

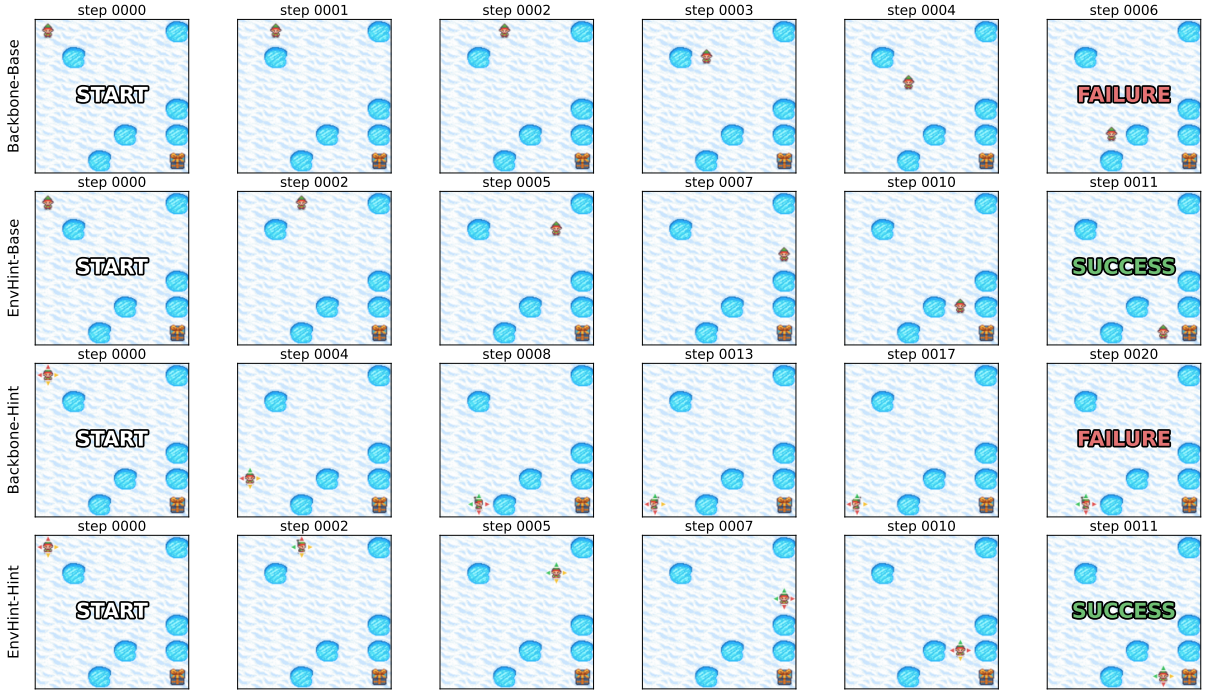



Figure 11: FrozenLake case study.

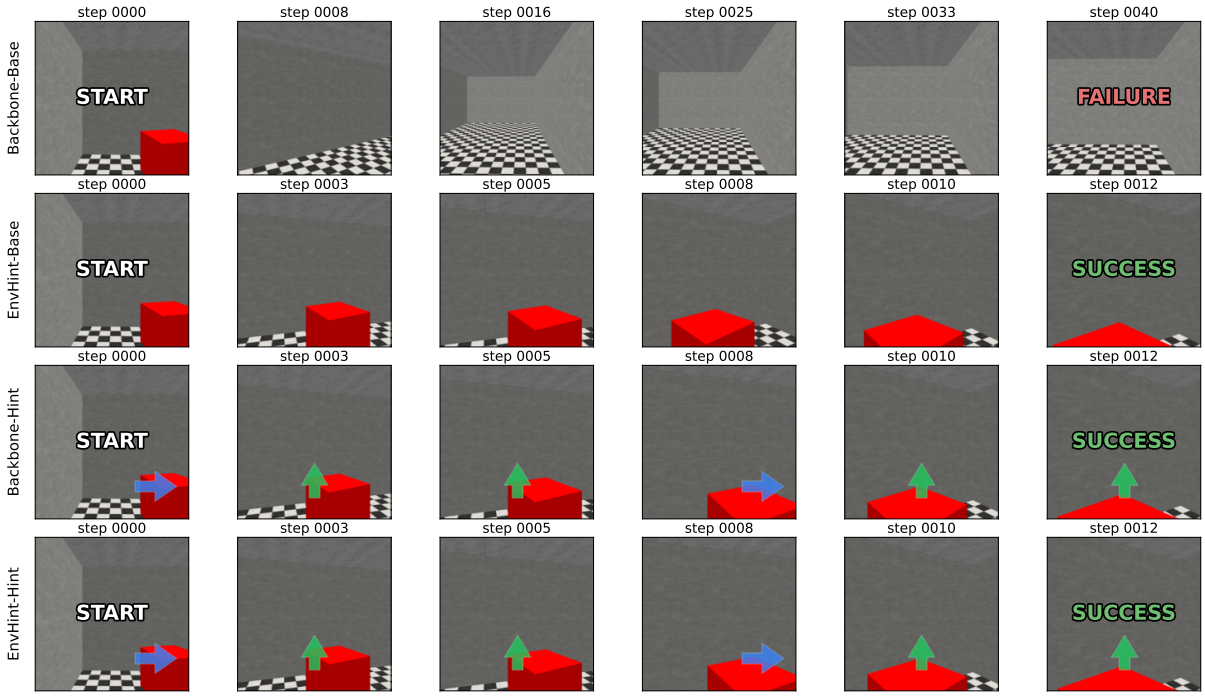



Figure 12: Hallway case study.