SELF-CORRECTING IMAGINATIVE PLANNING FOR VLM EMBODIED AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in Vision-Language Models (VLMs) have demonstrated their remarkable capability to serve as embodied agents in interactive environments. However, these agents often make plans (i.e., action sequences) that appear linguistically plausible but fail upon execution, without considering the visual consequences of each action. While prior works have explored enhancing VLM agents through language supervision or self-correction with text-based verification, enabling them to self-correct invalid plans against visual consequences before execution remains an open challenge. To tackle this, we propose **S**ELF-**C**ORRECTING **I**MAGINATIVE **P**LANNING (**SCIP**), a training framework that enables VLM agents to self-correct their plans against imagined visual consequences before execution. SCIP integrates an imagination-guided policy learning scheme that grounds the VLM agent's planning in visual consequences, and a verification-driven imagination learning scheme that calibrates the imagination process to faithfully reflect the actual environment dynamics. Extensive experiments show that SCIP outperforms existing embodied VLM agents across multiple embodied planning tasks. The code will be released upon acceptance.

1 INTRODUCTION

Recent advances in Vision-Language Models (VLMs) have demonstrated their remarkable capability to serve as embodied agents that perceive, reason, and act in interactive environments (Hurst et al., 2024; Anthropic, 2024; Gemini Team, Google, 2025). By leveraging the strong perception and language understanding inherited from large-scale pretraining, VLM-based agents are expected to make plans (i.e., action sequences) that can successfully achieve user goal across diverse embodied tasks, including high-level household planning (Shridhar et al., 2020; Yang et al., 2025) and low-level robotic manipulation (Zhang et al., 2025; Yang et al., 2025). However, despite the exhibited impressive language capabilities, these agents often make plans that appear linguistically plausible but fail to consider how each action would actually change the visual state of the environment (e.g., proposing to "find a sink, then pick up the plate" while no plate is actually placed near the sink). As a result, these linguistically plausible plans fail to achieve the user goal upon execution, severely limiting the capability of VLM agents on existing embodied benchmarks (Yang et al., 2025).

To enhance VLM agents in embodied tasks, recent efforts have explored several directions. Vlaser (Yang et al., 2026) and LoHo-Manip (Liu et al., 2026) build large-scale embodied reasoning datasets and hierarchical decomposition pipelines to bridge VLMs with downstream action policies. Reinforced Reasoner (Wu et al., 2025), VAGEN (Wang et al., 2025a), and Robot-R1 (Kim et al., 2025) further adopt reinforcement fine-tuning with textual reasoning rewards over environment states, action sequences, and next-state descriptions, respectively. While these methods (Yang et al., 2026; Liu et al., 2026; Wu et al., 2025; Kim et al., 2025; Wang et al., 2025a) improve the reasoning capabilities of VLM agents, they guide the agent solely through language supervision (e.g., state descriptions in text form), without exposing the agent to how each action would change the visual state of the environment. As a result, these methods are still tend to make plans based on their language prior and unable to encourage VLM agents to consider the resulting visual consequences when making plans, leading to the plans that fail to achieve the user goal upon execution.

Recent works on self-correction with verification mechanism (Jiang et al., 2025; Kumar et al., 2024; Shinn et al., 2023; Madaan et al., 2023; Huang et al., 2022; Li et al., 2024) train LLM/VLM agents

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

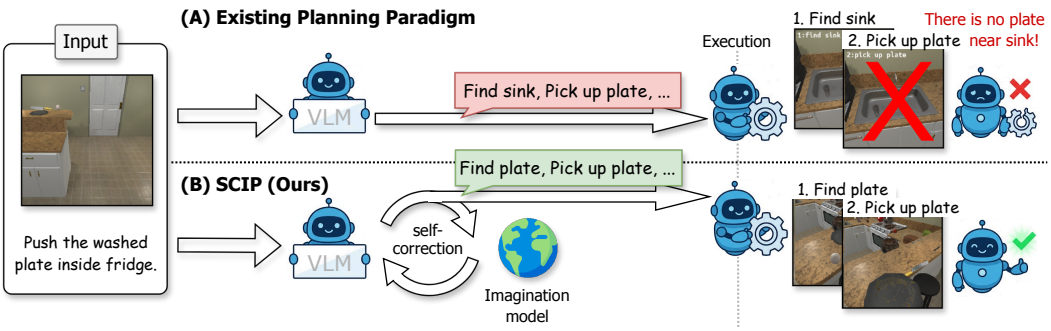


Figure 1: Comparison of (A) the existing planning paradigm, where VLM agents tend to make plans without considering how each action would change the visual state of the environment and fail when the plans do not match the actual environment dynamics, and (B) our SCIP, an imagination-guided self-correction paradigm where the VLM agent refines its plans against imagined visual consequences before execution.

to identify and revise their own mistakes through multi-turn interaction. However, these methods trigger correction only after a failed answer or action is observed, which is feasible for language-only tasks (e.g., mathematical reasoning) but not for embodied settings, where executing a failed action often leaves the failure unrecoverable. For example, PAG (Jiang et al., 2025) alternates LLMs between policy and verifier roles, but its verification is grounded in text alone, leaving the visual state outside the verification loop. Inner Monologue (Huang et al., 2022) brings verification into embodied settings, but its correction relies on environment feedback after each executed action, leaving no opportunity to prevent the failure before it occurs. Therefore, achieving self-correction grounded in the visual consequences induced by each action before execution remains challenging in embodied settings.

In this paper, we propose **S**ELF-**C**ORRECTING **I**MAGINATIVE **P**LANNING (**SCIP**), a training framework that enables VLM agent to self-correct their plans by refining them against imagined visual state changes before execution. As depicted in Fig. 1, SCIP is a imagination-guided self-correction paradigm that integrates *imagination-guided policy learning* and *verification-driven imagination learning*. More specifically, the former encourages the VLM agent to ground its planning in visual consequences by rewarding plans whose imagined visual states align with those of the ground-truth. The latter enables the imagined visual state changes faithfully reflect the actual environment dynamics by calibrating the imagination process with verification signals. With these two schemes, SCIP identifies whether a plan would achieve the user goal; if not, it explains the failure reason and revises the plan accordingly. By iteratively learning to plan with visual consequences and verify with faithful imagination, SCIP achieves self-correction grounded in visual consequences for embodied tasks.

We summarize our contributions as follows:

- We propose SCIP, a training framework that enables VLM agent to self-correct plans by refining them against imagined visual state changes before execution.
- We develop an imagination-guided policy learning scheme that encourages the VLM agent to ground its planning in visual consequences.
- We design a verification-driven imagination learning scheme that enables the imagination model to faithfully imagine visual states reflecting the actual environment dynamics, providing grounded visual evidence for verifying plans.

2 Related Work

Embodied Agents with Vision-Language Models. Vision-Language Models (VLMs) (Hurst et al., 2024; Anthropic, 2024; Gemini Team, Google, 2025; Bai et al., 2025) have shown strong capabilities as embodied agents in interactive environments. To enhance VLM agents in embodied tasks, research has explored two directions. Language-based approaches enrich VLMs with embodied data (Yang et al., 2026; Chen et al., 2025a;b) or reinforce internal reasoning through online reinforcement learning (Zhai et al., 2024; Wu et al., 2025; Wang et al., 2025a; Kim et al., 2025).

Although effective, these methods supervise the agent purely in language form, without exposing it to the visual consequences of each action. On the other hand, world models (Hafner et al., 2025; Assran et al., 2025; Bruce et al., 2024; Jang et al., 2025; Shang et al., 2025) predict future states in pixel or latent space, and recent embodied approaches leverage them for planning, either by unifying future-state prediction with action generation in a single transformer (Cen et al., 2025; Zhao et al., 2025) or by using world-model predictions as policy learning signals (Li et al., 2025; Xiao et al., 2025; Wang et al., 2025b). Despite their effectiveness, these approaches train the world model with task-agnostic objectives or use it as a fixed simulator, without calibrating it to produce imagined states that support correct plan verification. To address this, we propose SCIP to jointly train the policy and the world model with verification feedback, refining invalid plans against imagined visual consequences before execution.

Self-Correction in LLM / VLM Agents. Self-correction has emerged as a key capability for LLM-based agents to identify and revise their own mistakes. In the language domain, prior works have explored verbal reflection (Shinn et al., 2023; Madaan et al., 2023; Gou et al., 2024) and reinforcement learning (Kumar et al., 2024; Jiang et al., 2025; Ma et al., 2025; Lee et al., 2025; Liu et al., 2025) to train LLMs to alternate between policy and verifier roles, but their verification is grounded in text alone and cannot capture how an action would alter the visual state. Recent embodied extensions (Huang et al., 2022; Li et al., 2024) rely on environment feedback after execution to trigger plan revision, leaving the failure often unrecoverable. To address this, we propose SCIP to train the VLM agent as both planner and visual verifier, verifying and refining plans against imagined visual consequences before execution.

3 METHOD

3.1 PROBLEM FORMULATION

Given a high-level natural language instruction g and an initial visual observation o_0, a VLM-based embodied agent $\mathcal{M}_\theta$ aims to accomplish the instructed task in an interactive environment $\mathcal{E}$. At each planning step t, $\mathcal{M}_\theta$ observes the current state o_t and generates a plan $\hat{a}_{1:K} = \{\hat{a}_1, \ldots, \hat{a}_K\}$ consisting of K executable actions, which is then executed in $\mathcal{E}$, producing the next observation o_{t+1} for the next planning step until the task is accomplished. Each action $\hat{a}_k$ is expressed in natural language, either as a discrete skill (e.g., `pick up plate`) or as a low-level end-effector command (e.g., a 7-dim gripper pose with open/close state), and is handled identically by all components of SCIP. However, since $\mathcal{M}_\theta$ generates each plan purely in the language space, **it fails to consider the visual consequences induced by each action** $\hat{a}_k$, leading to plans that are linguistically plausible yet fail to achieve the user goal once executed.

To tackle this challenge, we propose **SCIP**, which enables $\mathcal{M}_\theta$ to refine its generated plans against the visual consequences imagined from an imagination model $\mathcal{W}_\phi$ before execution. As illustrated in Fig. 2, SCIP performs a plan-and-verify paradigm (Sec. 3.2) trained with *imagination-guided policy learning* (Sec. 3.3) and *verification-driven imagination learning* (Sec. 3.4). We summarize the training and inference procedures of SCIP in Sec. 3.5.

3.2 OVERVIEW OF SELF-CORRECTING IMAGINATIVE PLANNING

To enable a VLM agent to refine its proposed plans before execution, we design SCIP to integrate three core components: a VLM policy $\mathcal{M}^p_\theta$, an imagination model $\mathcal{W}_\phi$, and a VLM verifier $\mathcal{M}^v_\theta$ (sharing parameters with $\mathcal{M}^p_\theta$ but differing in input prompts). $\mathcal{W}_\phi$ is a latent world model built on a frozen V-JEPA2 (Assran et al., 2025) encoder; it maps the current observation and an action to the next latent state and rolls this forward one action at a time, so the world state is a latent embedding rather than pixels, and a diffusion decoder produces frames only for visualization (Appendix. D.4). As illustrated in Fig. 2 (A), $\mathcal{M}^p_\theta$ first proposes a candidate plan, $\mathcal{W}_\phi$ imagines the visual states after each action, and $\mathcal{M}^v_\theta$ verifies whether the plan would achieve the user goal based on the imagined visual states; if verification fails, the failed plan, imagined states, and explanation are fed back to $\mathcal{M}^p_\theta$ for refinement in the next turn. Formally, given the observation o_t at the t-th planning step, an instruction g, and the turn index ℓ, the plan-and-verify process is:

$$\hat{a}^\ell_{1:K} = \mathcal{M}^p_\theta(o_t, g, c^{\ell-1}), \quad \hat{z}^\ell_{1:K} = \mathcal{W}_\phi(o_t, \hat{a}^\ell_{1:K}), \quad (\hat{y}^\ell, \hat{e}^\ell) = \mathcal{M}^v_\theta(o_t, g, \hat{a}^\ell_{1:K}, \hat{z}^\ell_{1:K}), \quad (1)$$

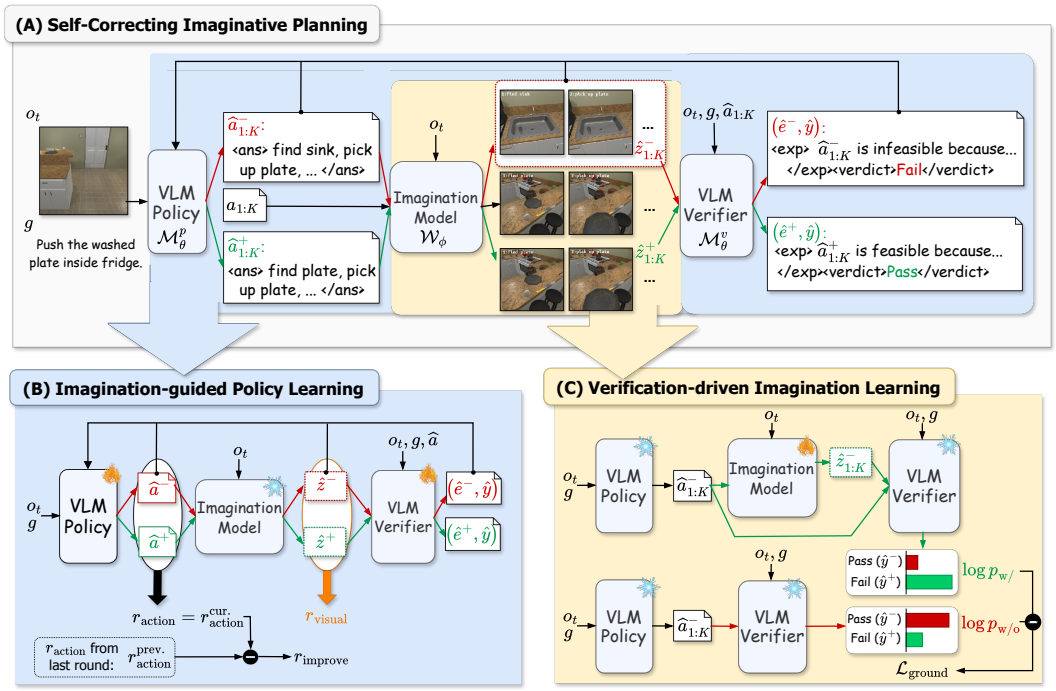


Figure 2: **Overview of SCIP.** (A) Self-Correcting Imaginative Planning enables the VLM agent to refine its plans before execution: $\mathcal{M}_\theta^p$ proposes a candidate plan, $\mathcal{W}_\phi$ imagines the visual states after executing each action, and $\mathcal{M}_\theta^v$ examines whether the plan would achieve the user goal and provides feedback to $\mathcal{M}_\theta^p$ for refining the plan upon failure. (B) Imagination-guided Policy Learning grounds $\mathcal{M}_\theta^p$'s plans in visual consequences by rewarding plans whose imagined visual consequences align with those of the ground-truth. (C) Verification-driven Imagination Learning calibrates $\mathcal{W}_\phi$ into a faithful imagination model that provides $\mathcal{M}_\theta^v$ (the same VLM as $\mathcal{M}_\theta^p$) with grounded visual evidence reflecting the environment dynamics.

where $c^{\ell-1} = (\hat{a}_{1:K}^{\ell-1}, \hat{z}_{1:K}^{\ell-1}, \hat{e}^{\ell-1})$ is the previous-turn feedback ($c^0 = \emptyset$), $\hat{y}^\ell \in \{\texttt{PASS}, \texttt{FAIL}\}$ is the verdict and $\hat{e}^\ell$ is the textual explanation; if $\hat{y}^\ell = \texttt{FAIL}$, c^ℓ is fed to $\mathcal{M}_\theta^p$ for the next turn, otherwise $\hat{a}_{1:K}^\ell$ is executed in $\mathcal{E}$, producing the next observation o_{t+1} for the subsequent planning step. To equip the above process with the capability of refining plans through visual imagination, we develop two complementary learning schemes (Sec. 3.3 & 3.4) that jointly train $\mathcal{M}_\theta^p$, $\mathcal{M}_\theta^v$, and $\mathcal{W}_\phi$.

3.3 Imagination-guided Policy Learning for Planning with Visual Consequences

To encourage the VLM policy ($\mathcal{M}_\theta^p$) to ground its plans in visual consequences, our key idea is to compute the reward in the visual space by leveraging the imagination model ($\mathcal{W}_\phi$). Since $\mathcal{W}_\phi$ predicts how the environment responds to each action, the visual consequences of an invalid action accumulate through subsequent imagined states, causing the final state to diverge from that of the ground-truth plan. For example, for the invalid plan "find sink, then pick up plate", $\mathcal{W}_\phi$'s final imagined state shows no held plate (as none exists at the sink), unlike the GT "find plate, then pick up plate", yielding a low imagination reward.

In contrast, computing rewards in the language space (Wang et al., 2025a; Bo et al., 2025) assigns high scores to plans sharing similar wordings with the ground-truth (GT), ignoring the actual environment state.

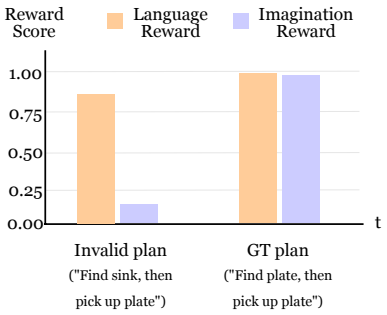


Figure 3: Reward comparison for an invalid plan vs. the GT. The Language Reward scores the invalid plan nearly as high as the GT, whereas the Imagination Reward sharply separates them, providing the learning signal for r_{visual}.

As shown in Fig. 3, our imagination reward correctly penalizes the invalid plan with a substantially larger gap between invalid and GT plans than the language reward.

Imagination Reward via Visual Consequence Similarity. To realize the above key idea, we design an imagination-guided reward r_{visual} that measures the cosine similarity between the final imagined states of the candidate and GT plans, with a threshold τ_{base} below which the reward is set to zero. Specifically, given an observation o_t, an instruction g, a candidate plan $\hat{a}_{1:K}$ proposed by $\mathcal{M}_\theta^p$, and the GT plan $a_{1:K'}$ (where K is determined by $\mathcal{M}_\theta^p$ and thus may differ from K'), r_{visual} is formulated as:

$$r_{\text{visual}} = \max\left(0,\, \cos\left(\hat{z}_K, z_{K'}\right) - \tau_{\text{base}}\right), \quad \text{where } \hat{z}_K = \left[\mathcal{W}_\phi(o_t, \hat{a}_{1:K})\right]_K \text{ and } z_{K'} = \left[\mathcal{W}_\phi(o_t, a_{1:K'})\right]_{K'}. \tag{2}$$

Since r_{visual} compares only the final imagined states, it does not require $\hat{a}_{1:K}$ to match $a_{1:K'}$ action by action, and plans reaching the same final state receive the same r_{visual}. By incorporating r_{visual} into reinforcement learning, $\mathcal{M}_\theta^p$ is steered toward plans grounded in visual consequences rather than mere linguistic patterns.

Role-specific Reward Shaping for Joint Policy-Verifier Optimization. While r_{visual} grounds the policy in visual consequences, we further design two *continuous* role-specific rewards, r_{policy} and r_{verify}, that respectively supervise $\mathcal{M}_\theta^p$ and $\mathcal{M}_\theta^v$ across the multi-turn rollout. Prior self-correction RL methods (Jiang et al., 2025; Kumar et al., 2024) employ binary rewards (i.e., 1 for correct and 0 for incorrect) to assess the correctness of the final answer in mathematical reasoning. However, directly applying these binary rewards to embodied planning yields sparse learning signals, since embodied plans are action sequences whose partial correctness cannot be captured by a single binary signal. Our designed r_{policy} and r_{verify} instead provide denser action-level signals.

More specifically, r_{policy} jointly grounds $\mathcal{M}_\theta^p$ in action-level correctness r_{action}, visual consequences r_{visual}, and turn-wise improvement r_{improve}, while r_{verify} promotes the confidence score of predicting verdicts from $\mathcal{M}_\theta^v$. Formally, the two role-specific rewards are formulated as:

$$r_{\text{policy}} = r_{\text{action}} + \lambda_{\text{visual}} \cdot r_{\text{visual}} + \lambda_{\text{improve}} \cdot r_{\text{improve}}, \quad r_{\text{verify}} = p_{\mathcal{M}_\theta^v}(y \mid o_t, g, \hat{a}_{1:K}, \hat{z}_{1:K}),$$
$$\text{where } r_{\text{action}} = \underbrace{\frac{|\{j : \hat{a}_j = a_j\}|}{K'}}_{\text{positional consistency}} \cdot \underbrace{\frac{K'}{\max(K', K)}}_{\text{length penalty}}, \quad r_{\text{improve}} = \max\left(0,\, r_{\text{action}}^{\text{cur.}} - r_{\text{action}}^{\text{prev.}}\right), \tag{3}$$

and $y = \texttt{PASS}$ if $r_{\text{action}} = 1$ and $\texttt{FAIL}$ otherwise. Here, r_{action} couples positional consistency between candidate $\hat{a}_{1:K}$ and ground-truth $a_{1:K'}$ plans with a length penalty against redundant actions, while r_{improve} rewards turn-wise gain in r_{action}. λ_{visual} and λ_{improve} are hyperparameters weighting their respective rewards. At the initial turn, r_{improve} is set to zero, reducing r_{policy} to $r_{\text{action}} + \lambda_{\text{visual}} \cdot r_{\text{visual}}$.

Building upon the rewards above, we update $\mathcal{M}_\theta^p$ and $\mathcal{M}_\theta^v$ jointly through the GRPO (Shao et al., 2024) objective:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{n}\sum_{i=1}^{n} A_i \log \mathcal{M}_\theta^p(\hat{\tau}_i), \quad A_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^n)}{\text{std}(\{r_j\}_{j=1}^n)}, \tag{4}$$

where $\hat{\tau}_i$ denotes the rollout token sequence of the i-th candidate. The reward r_i assigns r_{policy} to tokens from $\mathcal{M}_\theta^p$ and r_{verify} to tokens from $\mathcal{M}_\theta^v$, across all turns. We omit the standard clipping and KL terms for brevity (Shao et al., 2024).

By optimizing $\mathcal{L}_{\text{GRPO}}$ over multi-turn rollouts, $\mathcal{M}_\theta^p$ learns to ground its plans in visual consequences while $\mathcal{M}_\theta^v$ learns to verify them before execution.

3.4 Verification-driven Imagination Learning for Faithful State Predictions

To calibrate the imagination model ($\mathcal{W}_\phi$) to imagine future visual states that faithfully reflect the environment, we leverage the GT verdict y together with the verifier $\mathcal{M}_\theta^v$ as a calibration signal.

Our insight is that faithful imagined states enable $\mathcal{M}_\theta^v$ to assign higher likelihood to the correct verdict y, while hallucinated ones (e.g., a plate appearing on a sink where none exists) mislead it into incorrect verdicts. We thus calibrate $\mathcal{W}_\phi$ by steering $\mathcal{M}_\theta^v$ toward higher confidence on y when conditioned on the imagined states than without them.

Calibrating Imagination Model $\mathcal{W}_\phi$ with Verification Signals. To realize the above insight, we introduce a grounding loss $\mathcal{L}_{\text{ground}}$ that enforces $\mathcal{M}_\theta^v$'s ground-truth verdict likelihood with imagined visual states ($p_{\text{w/}}$) to exceed that without them ($p_{\text{w/o}}$). Formally, given a candidate plan $\hat{a}_{1:K}$ and its ground-truth verdict $y \in \{\texttt{PASS}, \texttt{FAIL}\}$, $\mathcal{L}_{\text{ground}}$ is formulated as:

$$\begin{aligned} \mathcal{L}_{\text{ground}} &= \max\left(0,\ \log p_{\text{w/o}} - \log p_{\text{w/}}\right), \\ \text{where } p_{\text{w/}} = p_{\mathcal{M}_\theta^v}(y \mid o_t, g, \hat{a}_{1:K}, \hat{z}_{1:K}),\ p_{\text{w/o}} &= p_{\mathcal{M}_\theta^v}(y \mid o_t, g, \hat{a}_{1:K}). \end{aligned} \tag{5}$$

Building upon $\mathcal{L}_{\text{ground}}$ that enforces correctness attributable to $\hat{z}_{1:K}$, we further introduce a verification loss $\mathcal{L}_{\text{verify}}$ that ensures $\mathcal{M}_\theta^v$'s verdict itself is accurate, and a regularization loss $\mathcal{L}_{\text{reg}}$ that preserves the interpretability of $\mathcal{W}_\phi$'s imagined visual latents, allowing them to be decoded back into images as during pre-training. Therefore, the total objective is formulated as:

$$\begin{aligned} \mathcal{L}_{\mathcal{W}_\phi} &= \mathcal{L}_{\text{ground}} + \lambda_{\text{verify}} \cdot \mathcal{L}_{\text{verify}} + \lambda_{\text{reg}} \cdot \mathcal{L}_{\text{reg}}, \\ \text{where } \mathcal{L}_{\text{verify}} = -\log p_{\text{w/}},\ \mathcal{L}_{\text{reg}} &= \left\| \mathcal{W}_\phi(o_t, \hat{a}_{1:K}) - \mathcal{W}_{\phi_{\text{ref}}}(o_t, \hat{a}_{1:K}) \right\|_2^2, \end{aligned} \tag{6}$$

where $\mathcal{W}_{\phi_{\text{ref}}}$ is the frozen pre-trained imagination model, and λ_{verify} and λ_{reg} are hyperparameters weighting $\mathcal{L}_{\text{verify}}$ and $\mathcal{L}_{\text{reg}}$, respectively.

By optimizing $\mathcal{L}_{\mathcal{W}_\phi}$, $\mathcal{W}_\phi$ is calibrated to supply $\mathcal{M}_\theta^v$ with grounded visual evidence, completing the alternating learning scheme that enables SCIP to self-correct plans before execution.

3.5 Summary of Training and Inference

Training. Starting from a pre-trained imagination model $\mathcal{W}_{\phi_{\text{ref}}}$ and a supervised fine-tuned VLM (see Appendix. D), SCIP is trained in two alternating rounds, where each round consists of imagination-guided policy learning (Sec. 3.3) followed by verification-driven imagination learning (Sec. 3.4). In policy learning rounds, $\mathcal{M}_\theta^p$ and $\mathcal{M}_\theta^v$ are jointly updated from the same VLM backbone with $\mathcal{W}_\phi$ frozen, where $\mathcal{W}_\phi$'s imagination of the ground-truth plan serves as the visual reference and the training set for r_{verify} is balanced between `PASS` and `FAIL` rollouts. In the imagination learning round, $\mathcal{W}_\phi$ is calibrated with θ frozen.

Inference. At inference time, given an observation o_t and an instruction g, SCIP performs the iterative plan-and-verify procedure of Eq. 1. As illustrated in Fig. 2, $\mathcal{M}_\theta^p$ first generates a candidate plan $\hat{a}_{1:K}$, which is then evaluated by $\mathcal{M}_\theta^v$ conditioned on the imagined visual states $\hat{z}_{1:K}$ produced by $\mathcal{W}_\phi$. If the verdict $\hat{y}$ is `PASS`, $\hat{a}_{1:K}$ is executed in $\mathcal{E}$; otherwise, the failed plan $\hat{a}_{1:K}$, the imagined visual states $\hat{z}_{1:K}$, and the explanation $\hat{e}$ are fed back to $\mathcal{M}_\theta^p$ to revise $\hat{a}_{1:K}$. The above procedure is iteratively performed up to T turns, where we set $T = 3$. If $\mathcal{M}_\theta^v$ does not return `PASS` within T turns, the final plan is executed; it is the most refined candidate, as r_{improve} rewards turn-wise gains, and has been verified at every turn.

4 Experiments

4.1 Experimental Setup

Benchmarks, Baselines, and Metrics. We primarily evaluate SCIP on the high-level planning benchmarks of EmbodiedBench (Yang et al., 2025) (EB-ALFRED, EB-Habitat), and further extend the evaluation to its low-level control tasks (EB-Manipulation, EB-Navigation). Each task spans five subsets — Base, Complex (Cpx.), Visual (Vis.), Common Sense (Com.), Spatial (Spat.); following (Chen et al., 2025b), we train on the seen split (Base, Cpx., Vis.) and evaluate out-of-distribution generalization on the fully held-out unseen split (Com., Spat.), reporting success rate (SR) per subset and the average. We compare SCIP against proprietary MLLMs (GPT-4o-mini (Hurst et al., 2024), Gemini-1.5-Flash (Team et al., 2024), Qwen-VL-Max (Bai et al., 2025)) and embodied agent baselines (VAGEN (Wang et al., 2025a), Reinforced Reasoner (Wu et al., 2025), Vlaser (Yang et al., 2026), VLA-RFT (Li et al., 2025), LoHo-Manip (Liu et al., 2026), RL4VLM (Zhai et al., 2024),

Table 1: Evaluation on two high-level planning benchmarks in Embodiedbench (Yang et al., 2025) (EB-ALFRED and EB-Habitat) benchmarks. **Bold**/underline: best/second-best per group; highlights: top-2 overall.

Method	EB-ALFRED						EB-Habitat					
	Avg	Base	Cpx.	Vis.	Com.	Spat.	Avg	Base	Cpx.	Vis.	Com.	Spat.
Frontier Proprietary MLLMs												
GPT-4o (Hurst et al., 2024)	56.8	64.0	68.0	46.0	54.0	52.0	58.0	86.0	56.0	68.0	44.0	36.0
Claude-3.7-Sonnet (Anthropic, 2025)	67.2	68.0	70.0	68.0	68.0	62.0	61.2	90.0	58.0	62.0	58.0	38.0
General Proprietary MLLMs												
GPT-4o-mini (Hurst et al., 2024)	28.8	34.0	36.0	24.0	28.0	22.0	36.4	74.0	32.0	22.0	22.0	**32.0**
GPT-4o-mini (Lang) (Hurst et al., 2024)	34.8	42.0	46.0	30.0	36.0	20.0	41.2	82.0	34.0	30.0	30.0	30.0
Gemini-1.5-Flash (Team et al., 2024)	41.6	44.0	56.0	42.0	40.0	26.0	44.8	76.0	48.0	36.0	32.0	**32.0**
Qwen-VL-Max (Bai et al., 2025)	43.2	44.0	44.0	42.0	48.0	38.0	47.2	74.0	50.0	42.0	**40.0**	30.0
Embodied VLM Agents												
VAGEN_{3B} (Wang et al., 2025a)	52.8	**70.0**	**70.0**	**58.0**	38.0	28.0	24.4	60.0	16.0	6.0	16.0	24.0
$\text{Reinforced Reasoner}_{7B}$ (Wu et al., 2025)	41.6	54.0	46.0	28.0	42.0	38.0	20.0	56.0	8.0	18.0	16.0	2.0
VLA-RFT_{3B} (Li et al., 2025)	18.0	24.0	24.0	6.0	24.0	12.0	20.0	32.0	22.0	18.0	12.0	16.0
Vlaser_{8B} (Yang et al., 2026)	50.0	62.0	60.0	42.0	**56.0**	**48.0**	42.8	74.0	42.0	40.0	32.0	26.0
LoHo-Manip_{4B} (Liu et al., 2026)	39.2	48.0	54.0	28.0	36.0	30.0	42.0	78.0	46.0	26.0	32.0	28.0
SCIP (Ours)	**54.0**	**70.0**	64.0	56.0	42.0	38.0	**68.0**	**92.0**	**96.0**	**82.0**	**40.0**	30.0

Table 2: Evaluation on EB-Manipulation, one of the low-level planning benchmarks in Embodiedbench (Yang et al., 2025). Notations follow Tab. 1.

Method	Avg	Base	Cpx.	Vis.	Com.	Spat.
Frontier Proprietary MLLMs						
GPT-4o (Hurst et al., 2024)	28.9	39.6	29.2	19.4	29.2	25.0
Claude-3.7-Sonnet (Anthropic, 2025)	28.5	31.3	43.8	25.0	20.8	20.8
General Proprietary MLLMs						
GPT-4o-mini (Hurst et al., 2024)	4.8	4.2	2.1	0.0	6.3	10.4
GPT-4o-mini (Lang) (Hurst et al., 2024)	6.6	12.5	2.1	2.8	0.0	14.6
Embodied VLM Agents						
RL4VLM_{3B} (Zhai et al., 2024)	21.9	**33.3**	**29.2**	**30.6**	8.3	8.3
Robot-R1_{7B} (Kim et al., 2025)	11.7	12.5	6.3	16.7	8.3	14.6
SCIP (Ours)	**22.8**	18.8	25.0	22.2	**20.8**	27.1

Table 3: Evaluation on EB-Navigation, one of the low-level planning benchmarks in Embodiedbench (Yang et al., 2025). Notations follow Tab. 1.

Method	Avg	Base	Cpx.	Vis.	Com.
Frontier Proprietary MLLMs					
GPT-4o (Hurst et al., 2024)	58.3	55.0	58.3	60.0	60.0
Claude-3.7-Sonnet (Anthropic, 2025)	49.6	50.0	50.0	36.7	61.7
General Proprietary MLLMs					
GPT-4o-mini (Hurst et al., 2024)	32.1	31.7	35.0	28.3	**33.3**
GPT-4o-mini (Lang) (Hurst et al., 2024)	10.4	3.3	10.0	15.0	13.3
Embodied VLM Agents					
VAGEN_{3B} (Wang et al., 2025a)	32.9	33.3	33.3	31.7	**33.3**
SCIP (Ours)	**41.7**	**43.3**	**45.0**	**50.0**	28.3

Robot-R1 (Kim et al., 2025)). GPT-4o (Hurst et al., 2024) and Claude-3.7-Sonnet (Anthropic, 2025) are additionally reported as references that indicate the very strong performance of frontier proprietary MLLMs, whose hundreds of billions of parameters are orders of magnitude more than the 2B backbone of SCIP.

Implementation Details. SCIP uses Qwen3-VL-2B-Instruct (Bai et al., 2025) as a shared VLM backbone for the policy $\mathcal{M}_\theta^p$ and verifier $\mathcal{M}_\theta^v$, and V-JEPA2 (ViT-g/16) (Assran et al., 2025) as the imagination model $\mathcal{W}_\phi$, with $T = 3$ self-correction turns at inference. Further implementation details and hyperparameters are provided in Appendix. D.

4.2 Quantitative Comparison

High-Level Planning. As shown in Tab. 1, SCIP achieves the best average SR among open-source embodied VLM agents on both EB-ALFRED (54.0%, $+1.2\%$ over VAGEN) and EB-Habitat (68.0%, $+25.2\%$ over Vlaser), as it corrects invalid plans before execution. For long-horizon instructions, linguistically plausible plans often violate environment dynamics, and SCIP detects these failures by verifying visual outcomes before execution. On EB-Habitat, SCIP even outperforms Claude-3.7-Sonnet by $+6.8\%$ on average, with the largest gain on the Cpx. subset ($+38.0\%$ over Claude-3.7-Sonnet), since long-horizon instructions benefit most from verifying imagined visual consequences.

Low-Level Control. As shown in Tab. 2 and Tab. 3, SCIP remains effective at low-level control. On EB-Manipulation, SCIP achieves the best average SR (22.8%) among open-source embodied agents, with the strongest gain on the unseen split ($+12.5\%$ on Com. and $+18.8\%$ on Spat. over

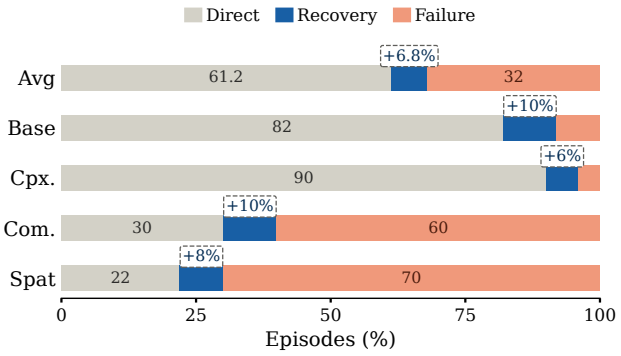


Figure 4: Analysis of failure recovery rate of SCIP on EB-Habitat across different subsets.

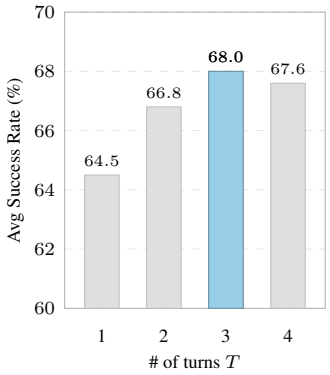


Figure 5: Ablation of SCIP on the maximum number of plan-and-verify iterations (T) on EB-Habitat.

Table 4: Ablation of SCIP on EB-Habitat (all rows SFT-initialized). ✓/✗ indicates whether the learning strategy is applied. **Bold** marks the best result per column.

Learning Strategy		EB-Habitat					
Sec. 3.3	Sec. 3.4	Avg	Base	Cpx.	Vis.	Com.	Spat.
✗	✗	57.6	90.0	86.0	60.0	34.0	18.0
✓	✗	64.8	90.0	90.0	78.0	40.0	26.0
✓	✓	**68.0**	**92.0**	**96.0**	**82.0**	**40.0**	**30.0**

Table 5: Analysis of the impact of imagined visual evidence on verification on EB-Habitat. **Bold** marks the best result per column.

Method	Avg	Base	Cpx.	Vis.	Com.	Spat.
SCIP w/o $\mathcal{W}_\phi$	63.6	**92.0**	86.0	**82.0**	32.0	26.0
SCIP	**68.0**	**92.0**	**96.0**	**82.0**	**40.0**	**30.0**

RL4VLM), indicating improved generalization to world-knowledge and spatial-reasoning instructions. On EB-Navigation, SCIP achieves 41.7% average SR ($+8.8\%$ over VAGEN).

4.3 Ablation Studies and Analysis

Ablation of SCIP's Learning Schemes. As shown in Tab. 4, starting from the SFT VLM (57.6% average SR), imagination-guided policy learning (Sec. 3.3) improves it to 64.8%, and verification-driven imagination learning (Sec. 3.4) further to 68.0%. Each scheme contributes on top of the previous one, with the largest gains on Vis. ($+22.0\%$) and Spat. ($+12.0\%$).

Analysis of the Impact of Imagined Visual Evidence on Verification. We compare SCIP against SCIP w/o $\mathcal{W}_\phi$ in Tab. 5, which removes the imagined states $\hat{z}_{1:K}$ from the verifier input at inference only, while training remains guided by $\mathcal{W}_\phi$ through r_{visual}. Removing $\mathcal{W}_\phi$ drops the average SR from 68.0% to 63.6%, with the drop concentrated on Cpx. (-10.0%), Com. (-8.0%) and Spat. (-4.0%) while Base and Vis. stay unchanged, suggesting that imagined visual consequences provide the grounded evidence the verifier relies on, with the largest gains on complex instructions.

Analysis of Failure Recovery via Self-Correction. To verify that the verifier's verdicts reflect plan quality, we decompose EB-Habitat test episodes in Fig. 4 into Direct, Recovery, and Failure. SCIP recovers $+6.8\%$ of episodes on average, with the largest gains on Base and Com. (both $+10\%$), suggesting that $\mathcal{M}_\theta^v$ reliably identifies invalid plans before execution and $\mathcal{M}_\theta^p$ effectively revises them given the imagined visual evidence from $\mathcal{W}_\phi$.

Analysis of the Number of Self-Correction Turns. As shown in Fig. 5, the average SR monotonically improves from 64.5% to 68.0% as T increases from 1 to 3, then marginally drops at $T = 4$. Three turns are sufficient for refining invalid plans, while additional turns introduce redundant deliberation; we thus set $T = 3$ as default.

Analysis of Goal Achievement beyond Ground-Truth Plans. Since r_{visual} rewards any plan reaching the same final state as $a_{1:K'}$ (Eq. 2), we examine whether $\mathcal{M}_\theta^p$ achieves the goal through plans other than $a_{1:K'}$ by counting the successful test episodes whose executed plan differs from

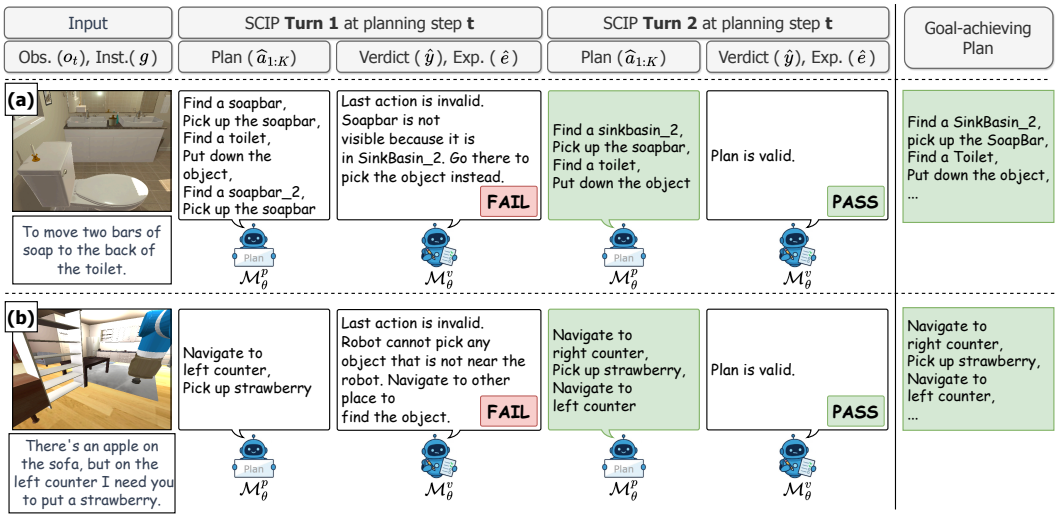


Figure 6: Qualitative examples of self-correction by SCIP on (a) EB-ALFRED and (b) EB-Habitat. Each row visualizes the plan-and-verify procedure at planning step t: $\mathcal{M}_\theta^p$ proposes a plan $\hat{a}_{1:K}$, and $\mathcal{M}_\theta^v$ produces the verdict $\hat{y}$ with explanation $\hat{e}$ conditioned on the imagined visual states from $\mathcal{W}_\phi$. Upon `FAIL` in Turn 1, $\mathcal{M}_\theta^p$ refines the plan in Turn 2 until $\mathcal{M}_\theta^v$ returns `PASS`. The rightmost column shows the leading actions of the goal-achieving plan, which match the plan verified by $\mathcal{M}_\theta^v$ in Turn 2, suggesting that $\mathcal{M}_\theta^v$'s `PASS` verdict aligns with plans that successfully achieve the user goal.

$a_{1:K'}$. SCIP achieves 68.5% of its successes on EB-Habitat and 50.4% on EB-ALFRED with such plans, suggesting that $\mathcal{M}_\theta^p$ learns to reach the goal through multiple valid plans rather than a single one.

Analysis of Computational Cost. As reported in Appendix. B.1, the full $T = 3$ loop of SCIP adds only 19% inference time over a single pass of Vlaser-8B (Yang et al., 2026) (103.8s vs. 86.7s per sample on EB-ALFRED), while surpassing it by $+25.2\%$ on EB-Habitat and $+4.0\%$ on EB-ALFRED with a 2B backbone. The gain from self-correction thus far outweighs its overhead.

4.4 Qualitative Results

Self-Correction Examples. Fig. 6 shows how $\mathcal{M}_\theta^v$ rejects invalid plans by detecting object mismatches in the imagined visual states. In (a), $\mathcal{M}_\theta^v$ identifies that the soapbar is located in `SinkBasin_2`; in (b), $\mathcal{M}_\theta^v$ detects that the strawberry is not near the robot. Both cases lead $\mathcal{M}_\theta^p$ to a revised plan that is correctly verified as achieving the user goal.

Visualization of Imagined Visual Latents. To verify that $\mathcal{W}_\phi$ faithfully reflects the environment, we decode and visualize $\hat{z}_{1:K}$ on EB-ALFRED and EB-Navigation, where the imagined states match the ground truth even for no-effect actions such as picking up an already-held object, suggesting that $\mathcal{W}_\phi$ supplies $\mathcal{M}_\theta^v$ with grounded visual evidence (see Fig. 7 and 8 in Appendix. C.2).

5 Conclusion

In this paper, we introduce Self-Correcting Imaginative Planning (SCIP), a training framework that enables VLM agents to self-correct their plans by refining them against imagined visual consequences before execution. To ground the VLM agent's planning in visual consequences, we develop an imagination-guided policy learning scheme that rewards plans whose imagined visual states align with those of the ground-truth. To produce imagined visual states that faithfully reflect the actual environment dynamics, we design a verification-driven imagination learning scheme that calibrates the imagination model with verification signals from the VLM verifier. Extensive experiments demonstrate that SCIP substantially improves embodied planning capability over existing approaches, ranking first in average success rate among open-source embodied VLM agents on all four EmbodiedBench tasks, spanning both high-level planning and low-level control.

AI USE STATEMENT

We used generative AI tools to suggest how to present existing results, to draft and edit parts of the manuscript, and to assist in writing code; all results were produced by the authors, and all AI-assisted text and code were reviewed and verified by the authors. We did not use them to generate data, formulate or prove claims, or design experiments; other required-disclosure tasks are not applicable. We take responsibility for the final content of this work.

ETHICS STATEMENT

This work uses the public EmbodiedBench benchmark (Yang et al., 2025) and publicly released pre-trained models, and involves no human subjects or personal data. We have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

The method and its training and inference procedures are described in Sec. 3. Appendix. D provides all hyperparameters, the SFT data construction, and the pretraining of the imagination model; the evaluation protocol is specified in Sec. 4.1. All experiments use public benchmarks and pre-trained models. The code will be released upon acceptance.

REFERENCES

Anthropic. Claude 3.5 sonnet. `https://www.anthropic.com/news/claude-3-5-sonnet`, 2024. Model announcement.

Anthropic. Claude 3.7 sonnet. `https://www.anthropic.com/news/claude-3-7-sonnet`, 2025. Model announcement.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Zitong Bo, Yue Hu, Jinming Ma, Mingliang Zhou, Junhui Yin, Yachen Kang, Yuqi Liu, Tong Wu, Diyun Xiang, and Hao Chen. Reinforced embodied planning with verifiable reward for real-world robotic manipulation. *arXiv preprint arXiv:2509.25852*, 2025.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Delong Chen, Theo Moutakanni, Willy Chung, Yejin Bang, Ziwei Ji, Allen Bolourchi, and Pascale Fung. Planning with reasoning using vision language world model. *arXiv preprint arXiv:2509.02722*, 2025a.

Hanyang Chen, Mark Zhao, Rui Yang, Qinwei Ma, Ke Yang, Jiarui Yao, Kangrui Wang, Hao Bai, Zhenhailong Wang, Rui Pan, et al. Era: Transforming vlms into embodied agents via embodied prior learning and online reinforcement learning. *arXiv preprint arXiv:2510.12693*, 2025b.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Gemini Team, Google. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Sx038qxjek.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pp. 1–7, 2025.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

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.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

Yuhua Jiang, Yuwen Xiong, Yufeng Yuan, Chao Xin, Wenyuan Xu, Yu Yue, Qianchuan Zhao, and Lin Yan. Pag: Multi-turn reinforced llm self-correction with policy as generative verifier. *arXiv preprint arXiv:2506.10406*, 2025.

Dongyoung Kim, Sumin Park, Huiwon Jang, Jinwoo Shin, Jaehyung Kim, and Younggyo Seo. Robot-r1: Reinforcement learning for enhanced embodied reasoning in robotics. *arXiv preprint arXiv:2506.00070*, 2025.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

Hyunseok Lee, Seunghyuk Oh, Jaehyung Kim, Jinwoo Shin, and Jihoon Tack. Revise: Learning to refine at test-time via intrinsic self-verification. *arXiv preprint arXiv:2502.14565*, 2025.

Chenxuan Li, Jiaming Liu, Guanqun Wang, Xiaoqi Li, Sixiang Chen, Liang Heng, Chuyan Xiong, Jiaxin Ge, Renrui Zhang, Kaichen Zhou, et al. A self-correcting vision-language-action model for fast and slow system manipulation. *arXiv preprint arXiv:2405.17418*, 2024.

Hengtao Li, Pengxiang Ding, Runze Suo, Yihao Wang, Zirui Ge, Dongyuan Zang, Kexian Yu, Mingyang Sun, Hongyin Zhang, Donglin Wang, et al. Vla-rft: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators. *arXiv preprint arXiv:2510.00406*, 2025.

Isabella Liu, An-Chieh Cheng, Rui Yan, Geng Chen, Ri-Zhao Qiu, Xueyan Zou, Sha Yi, Hongxu Yin, Xiaolong Wang, and Sifei Liu. Long-horizon manipulation via trace-conditioned vla planning. *arXiv preprint arXiv:2604.21924*, 2026.

Xiaoyuan Liu, Tian Liang, Zhiwei He, Jiahao Xu, Wenxuan Wang, Pinjia He, Zhaopeng Tu, Haitao Mi, and Dong Yu. Trust, but verify: A self-verification approach to reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2505.13445*, 2025.

Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. S2r: Teaching llms to self-verify and self-correct via reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22632–22654, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Yu Shang, Xin Zhang, Yinzhou Tang, Lei Jin, Chen Gao, Wei Wu, and Yong Li. Roboscape: Physics-informed embodied world model. *arXiv preprint arXiv:2506.23135*, 2025.

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10740–10749, 2020.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Yiping Lu, Zhengyuan Yang, Lijuan Wang, Ranjay Krishna, Jiajun Wu, Li Fei-Fei, Yejin Choi, and Manling Li. VAGEN: Reinforcing world model reasoning for multi-turn VLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=xpjWEgf8zi.

Siyin Wang, Zhaoye Fei, Qinyuan Cheng, Shiduo Zhang, Panpan Cai, Jinlan Fu, and Xipeng Qiu. World modeling makes a better planner: Dual preference optimization for embodied task planning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21518–21537, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1044. URL https://aclanthology.org/2025.acl-long.1044/.

Di Wu, Jiaxin Fan, Junzhe Zang, Guanbo Wang, Wei Yin, Wenhao Li, and Bo Jin. Reinforced reasoning for embodied planning. *arXiv preprint arXiv:2505.22050*, 2025.

Junjin Xiao, Yandan Yang, Xinyuan Chang, Ronghan Chen, Feng Xiong, Mu Xu, Wei-Shi Zheng, and Qing Zhang. World-env: Leveraging world model as a virtual environment for vla post-training. *arXiv preprint arXiv:2509.24948*, 2025.

Ganlin Yang, Tianyi Zhang, Haoran Hao, Weiyun Wang, Yibin Liu, Dehui Wang, Guanzhou Chen, Zijian Cai, Junting Chen, Weijie Su, Wengang Zhou, Yu Qiao, Jifeng Dai, Jiangmiao Pang, Gen Luo, Wenhai Wang, Yao Mu, and Zhi Hou. Vlaser: Vision-language-action model with synergistic embodied reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8xTDnj39Ti.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DgGF2LEBPS.

Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *Advances in neural information processing systems*, 37:110935–110971, 2024.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, et al. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11142–11152, 2025.

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

Self-Correcting Imaginative Planning for VLM Embodied Agents

Appendix

A Limitations and Future Work

SCIP is evaluated in simulation on EmbodiedBench (Yang et al., 2025), following the embodied agents we compare against. The only component tied to simulation is the trajectory source used to pre-train $\mathcal{W}_{\phi_{\text{ref}}}$; the V-JEPA2 (Assran et al., 2025) encoder is pre-trained on large-scale real video, and the plan-and-verify procedure is agnostic to where the observations come from, so extending SCIP to real robots only requires collecting real-world trajectories for $\mathcal{W}_{\phi_{\text{ref}}}$ and leaves the rest unchanged. SCIP also inherits the limitations of its pre-trained imagination model, whose predictions on out-of-distribution dynamics may mislead $\mathcal{M}_\theta^v$; verification-driven imagination learning mitigates this, and stronger imagination backbones or uncertainty-aware verification are directions for future work.

B Additional Analysis

B.1 Inference Latency

Tab. 6 reports the average per-sample inference time on EB-ALFRED (Yang et al., 2025), measured on two NVIDIA H100 80GB GPUs. The full $T = 3$ loop of SCIP adds only 19% inference time over a single pass of $\text{Vlaser}_{8\text{B}}$ (Yang et al., 2026), while SCIP with a 2B backbone outperforms $\text{Vlaser}_{8\text{B}}$ on all high-level planning tasks, with average success rate improvements of 25.2% on EB-Habitat and 4.0% on EB-ALFRED. The gain from self-correction thus far outweighs its overhead.

Table 6: **Per-sample inference latency** comparison on EB-ALFRED (Yang et al., 2025) benchmark. Results are reported in seconds.

Models	Avg	Base	Cpx.	Vis.	Com.	Spat.
$\text{Qwen3-VL}_{3\text{B}}$ (Wang et al., 2025a)	56.1	53.1	51.3	56.8	59.2	60.1
$\text{Vlaser}_{8\text{B}}$ (Yang et al., 2026)	86.7	97.9	63.2	81.9	113.7	76.8
SCIP (Ours)	103.8	83.5	92.0	90.2	126.6	126.5

B.2 Plug-and-play Generalizability of Visual Imagination

To evaluate the robustness of our proposed paradigm, we investigate whether the plan-and-verify scheme can benefit off-the-shelf VLMs without any of the SCIP-specific training described in Sec. 3. We use the original Qwen3-VL-2B-Instruct (Bai et al., 2025) as the backbone for both the policy $\mathcal{M}_\theta^p$ and the verifier $\mathcal{M}_\theta^v$, integrated with the pre-trained imagination model $\mathcal{W}_{\phi_{\text{ref}}}$.

Table 7: **Zero-shot performance gain** using imagination model $\mathcal{W}_{\phi_{\text{ref}}}$ on the original Qwen3-VL-2B-Instruct (Bai et al., 2025) backbone across EB-ALFRED and EB-Habitat benchmarks (Yang et al., 2025).

	EB-ALFRED						**EB-Habitat**					
Method	Avg	Base	Cpx.	Vis.	Com.	Spat.	Avg	Base	Cpx.	Vis.	Com.	Spat.
Qwen3-VL-2B-Instruct (Bai et al., 2025)	4.8	10.0	8.0	2.0	2.0	2.0	11.2	36.0	4.0	10.0	4.0	2.0
+ $\mathcal{W}_{\phi_{\text{ref}}}$ (w/o Sec. 3)	10.4	6.0	26.0	6.0	8.0	6.0	20.8	48.0	20.0	8.0	14.0	14.0

Table 8: Ablation of SCIP on EB-ALFRED (Yang et al., 2025). ✓/✗ indicates whether the learning strategy is applied; the first row uses the original Qwen3-VL-2B-Instruct with $\mathcal{W}_{\phi_{\text{ref}}}$ and without SFT. **Bold** marks the best result per column.

Learning Strategy		EB-ALFRED					
§3.3	§3.4	Avg	Base	Cpx.	Vis.	Com.	Spat.
✗	✗	10.4	6.0	26.0	6.0	8.0	6.0
✓	✗	47.6	52.0	58.0	52.0	**44.0**	32.0
✓	✓	**54.0**	**70.0**	**64.0**	**56.0**	42.0	**38.0**

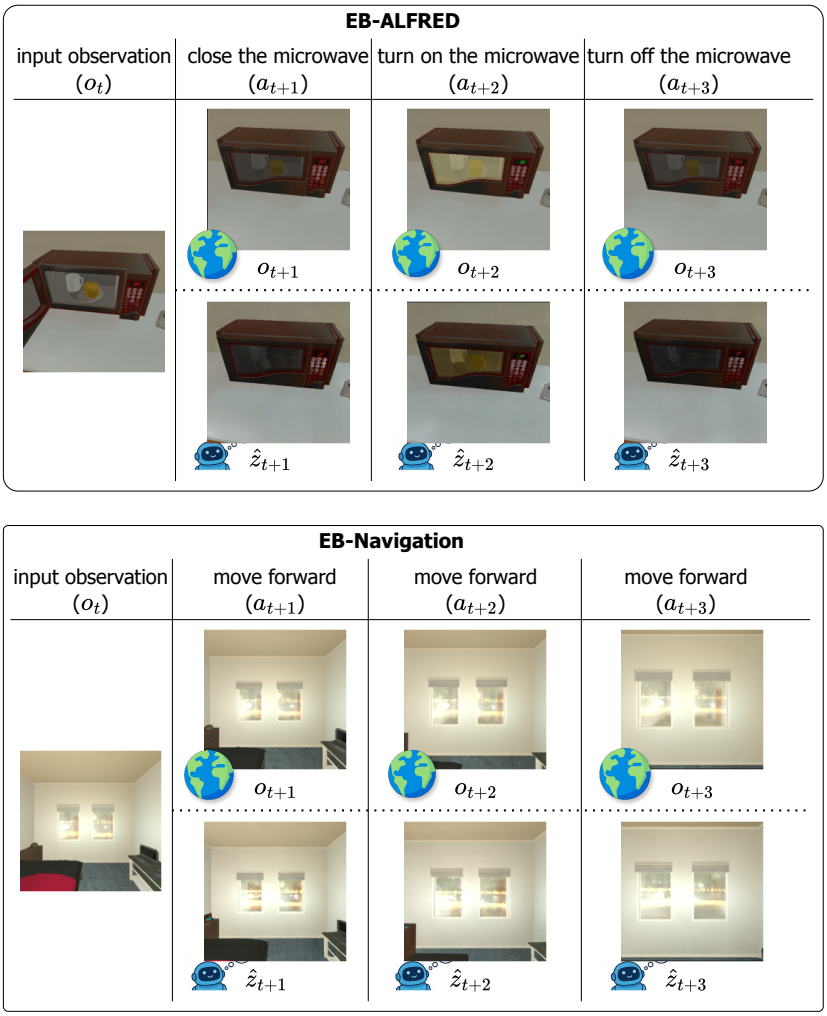


Figure 7: Visualization of imagined visual latents on EB-ALFRED and EB-Navigation when every action in the plan is valid. Given the input observation o_t and a sequence of valid actions $a_{t+1:t+3}$, $\mathcal{W}_\phi$'s imagined latents $\hat{z}_{t+1:t+3}$ (decoded back into images, bottom row) faithfully match the ground-truth observations $o_{t+1:t+3}$ returned by the simulator (top row), confirming that $\mathcal{W}_\phi$ correctly imagines the next state induced by each valid action.

As shown in Tab. 7, even without imagination-guided policy learning, the simple addition of a visual imagination feedback loop significantly improves the success rate across nearly all categories. For instance, on EB-Habitat, the average Success Rate improves from 11.2% to 20.8%. This suggests that the plan-and-verify paradigm generalizes to off-the-shelf VLMs, grounding their reasoning in visual consequences.

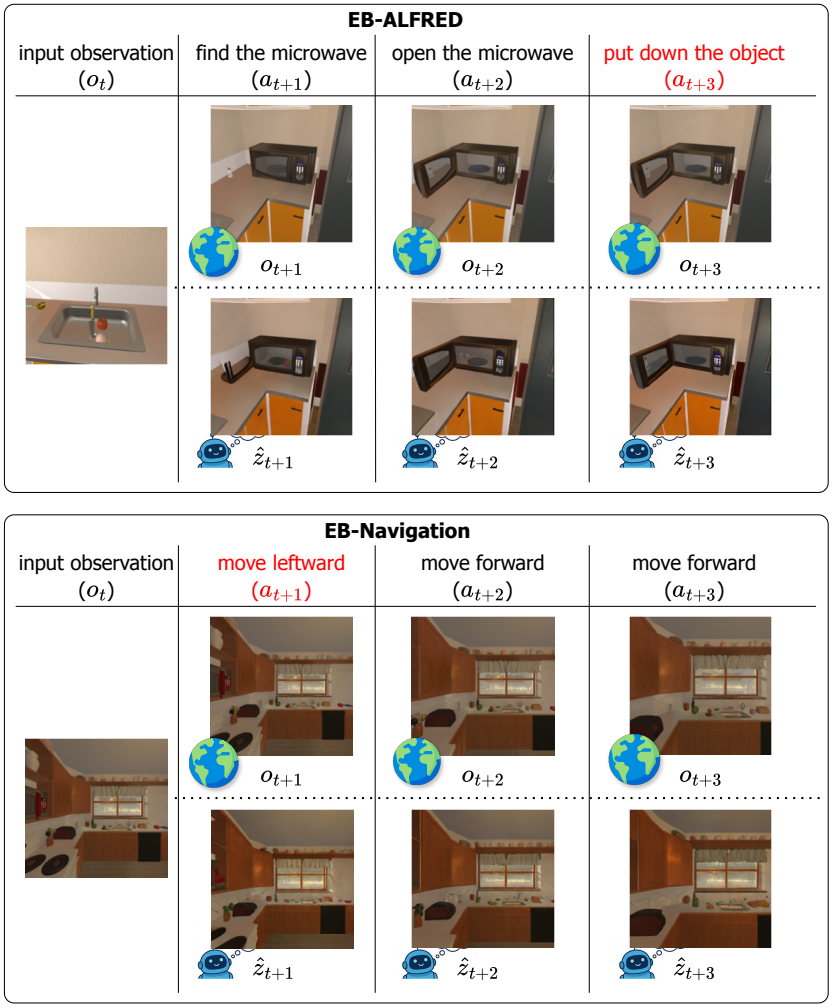


Figure 8: Visualization of imagined visual latents on EB-ALFRED and EB-Navigation when one of the actions in the plan is invalid. The action highlighted in red (a_{t+3} in EB-ALFRED and a_{t+1} in EB-Navigation) is invalid (e.g., put down the object when the robot holds no object), so the simulator returns the previous state. $\mathcal{W}_\phi$'s imagined latents $\hat{z}_{t+1:t+3}$ (bottom row) likewise stay in the previous state, faithfully matching the ground-truth observations $o_{t+1:t+3}$ returned by the simulator (top row), confirming that $\mathcal{W}_\phi$ correctly imagines the next state regardless of whether the action is valid.

C Additional Results

C.1 Additional Ablation of SCIP's Learning Schemes

In Tab. 8, we provide another ablation of SCIP's learning schemes on EB-ALFRED (Yang et al., 2025), which further verifies that the two learning schemes are complementary.

C.2 Visualization of Imagined Visual Latents

To verify that $\mathcal{W}_\phi$ faithfully reflects the actual environment dynamics, we visualize the imagined visual latents $\hat{z}_{1:K}$ on EB-ALFRED (Yang et al., 2025) and EB-Navigation (Yang et al., 2025) under two scenarios: (i) when every action in the plan is valid, and (ii) when one of the actions is invalid. We decode $\hat{z}_{1:K}$ back into images using our Stage-2 diffusion decoder (Sec. D.4) and compare them against the ground-truth observations $o_{1:K}$ returned by the simulator. As shown in Fig. 7, when every action is valid, $\hat{z}_{1:K}$ faithfully matches $o_{1:K}$ across all timesteps. As shown in Fig. 8, even when an action is invalid (e.g., `put down the object` when the robot holds no

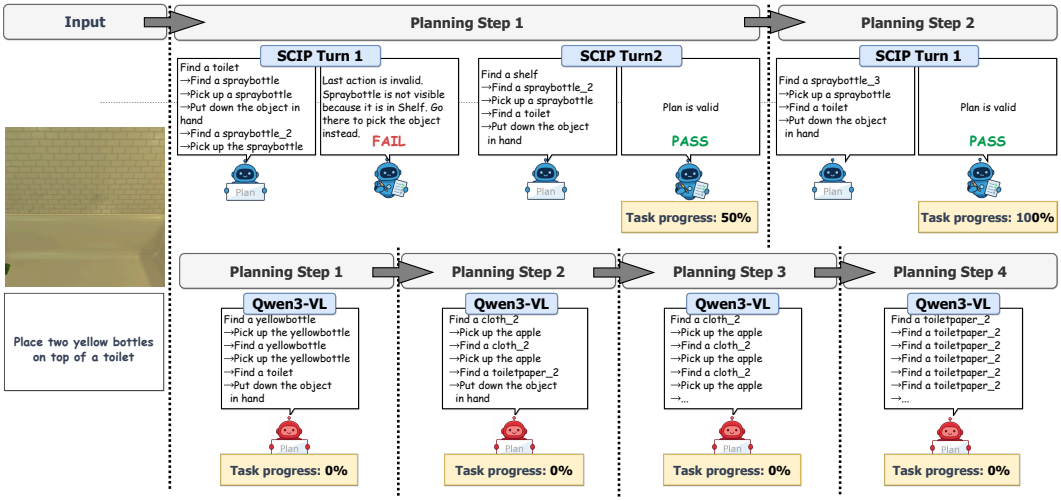


Figure 9: Episode-level qualitative comparison between SCIP and the base VLM Qwen3-VL on EB-ALFRED under the instruction "Place two yellow bottles on top of a toilet." At each planning step, SCIP iteratively refines its plan via the plan-and-verify procedure (Turn 1, Turn 2), executing the verified plan upon `PASS`; Qwen3-VL directly executes its proposed plan without verification. Task progress denotes the fraction of subgoals completed (lower is worse). SCIP completes the task with 100% task progress within two planning steps, while Qwen3-VL produces increasingly degenerate plans across four planning steps and stagnates at 0% task progress.

object), the simulator returns the previous state, and $\mathcal{W}_\phi$ likewise stays in the previous state. These results confirm that $\mathcal{W}_\phi$ correctly imagines the next state regardless of whether the action is valid, providing $\mathcal{M}_\theta^v$ with grounded visual evidence for plan verification.

C.3 Episode-Level Qualitative Comparisons.

We provide qualitative comparisons between SCIP and existing methods on representative episodes from EB-ALFRED and EB-Navigation, illustrating how the plan-and-verify procedure compounds across planning steps to prevent invalid plans from accumulating into either degenerate behaviors or ineffective navigation.

Comparison with Qwen3-VL (Bai et al., 2025) on EB-ALFRED. To complement the quantitative results in Tab. 1, Fig. 9 compares SCIP against the base VLM Qwen3-VL (Bai et al., 2025) on the instruction "Place two yellow bottles on top of a toilet.". At each planning step, SCIP's verifier $\mathcal{M}_\theta^v$ rejects the initial plan that picks up a spraybottle at the default location and prompts $\mathcal{M}_\theta^p$ to refine it after detecting from the imagined visual states that the spraybottle is in the shelf, whereas Qwen3-VL directly executes its proposed plan without verification. Across the entire episode, SCIP completes the task within two planning steps with task progress reaching 100%, while Qwen3-VL produces increasingly degenerate plans (e.g., repeating "Find a toiletpaper_2" across consecutive steps) and stagnates at 0% task progress.

Comparison with VAGEN (Wang et al., 2025a) on EB-Navigation. To complement the quantitative results in Tab. 3, Fig. 10 compares SCIP against VAGEN on the instruction "Navigate to the Mug in the room and be as close as possible to it.". At each planning step, SCIP's verifier $\mathcal{M}_\theta^v$ rejects invalid plans proposed by $\mathcal{M}_\theta^p$ (e.g., `Moveahead` or `Moveright` when blocked) and refines them across multiple turns until a valid plan passes verification, whereas VAGEN directly executes its proposed plan without verification. Across the entire episode, SCIP progressively reduces its distance to the Mug from 2.09 to 0.93 over five planning steps and reaches the goal, while VAGEN stagnates around 1.68–2.23.

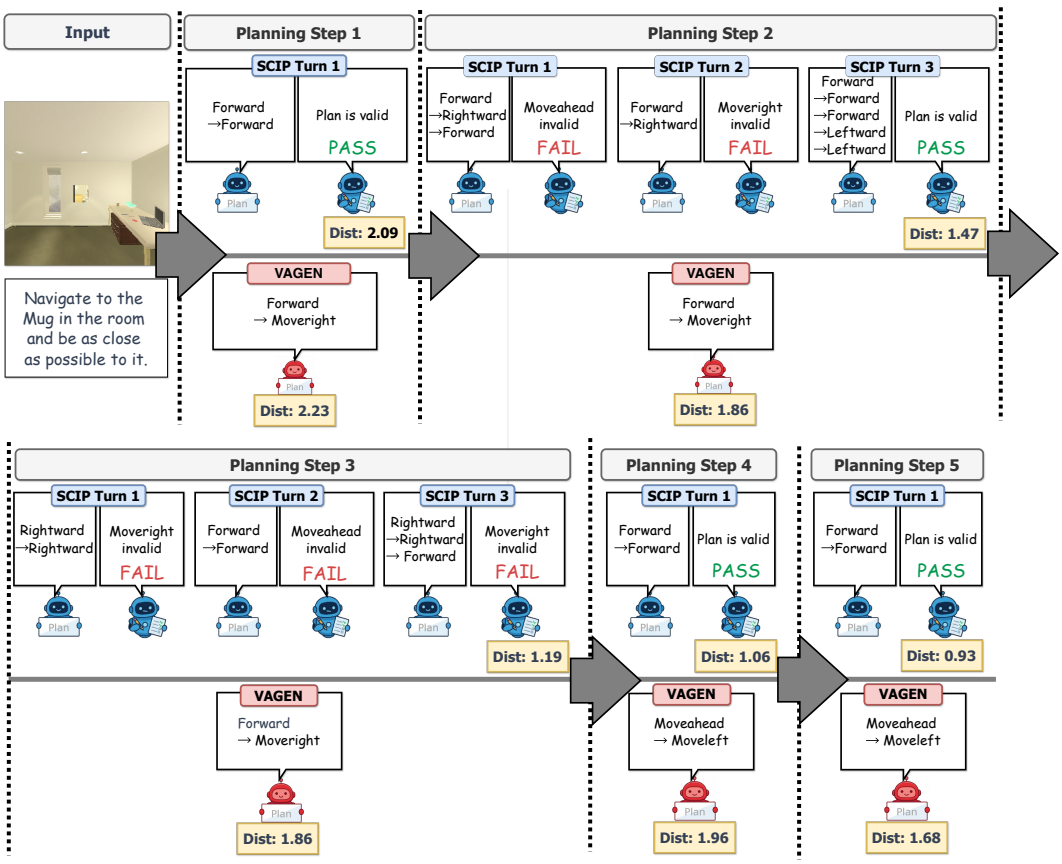


Figure 10: Episode-level qualitative comparison between SCIP and VAGEN on EB-Navigation under the instruction "Navigate to the Mug in the room and be as close as possible to it." At each planning step, SCIP iteratively refines its plan via the plan-and-verify procedure (Turn 1, 2, 3), executing the verified plan upon `PASS` or falling back to the last attempted plan upon reaching the maximum number of turns; VAGEN directly executes its proposed plan without verification. Dist: distance from the agent to the Mug after the planning step (lower is better). Over five planning steps, SCIP progressively reduces the distance from 2.09 to 0.93, while VAGEN stagnates around 1.68–2.23.

D Detailed Experimental Setting

D.1 Hyperparameters of SCIP

Hyperparameters. As for the GRPO objective, we set the group size to $n = 4$.[1] For the imagination-guided reward r_{visual}, we set the similarity threshold $\tau_{\text{base}} = 0.6$.[2] The role-specific reward weights are configured as $\lambda_{\text{visual}} = 1.5$ and $\lambda_{\text{improve}} = 0.5$. For verification-driven imagination learning, we set $\lambda_{\text{verify}} = 1.0$ and $\lambda_{\text{reg}} = 0.1$. All experiments are conducted on 4 NVIDIA H100 GPUs.

D.2 Design Motivation for Continuous Role-specific Rewards

Prior self-correction RL methods (Jiang et al., 2025; Kumar et al., 2024) employ binary rewards (i.e., 1 for correct and 0 for incorrect) to assess the correctness of the final answer in mathematical

[1] We omit the importance-ratio clipping and KL regularization terms in Eq. 4 for brevity, following the standard GRPO formulation (Shao et al., 2024).

[2] The sample index i is omitted in Eq. 2 for brevity; r_{visual} is computed per candidate plan within the group of n rollouts.

reasoning. However, directly applying these binary rewards to embodied planning yields sparse learning signals, since embodied plans are action sequences whose partial correctness cannot be captured by a single binary signal. To address this, in SCIP we instantiate two *continuous* role-specific rewards, r_{policy} and r_{verify}, that respectively supervise $\mathcal{M}_\theta^p$ and $\mathcal{M}_\theta^v$ with denser action-level signals: r_{policy} couples r_{action} (positional consistency with length penalty) and r_{improve} (turn-wise gain) to provide fine-grained credit assignment along the plan, while r_{verify} uses verdict confidence to capture how strongly $\mathcal{M}_\theta^v$ believes in its own verdict beyond a binary correct/incorrect signal.

D.3 Supervised Warm-Start of the VLM

We perform a supervised fine-tuning (SFT) warm-start that initializes the shared VLM backbone θ_{ref} before the alternating SCIP training (Sec. 3.3 & 3.4). As Qwen3-VL-2B-Instruct (Bai et al., 2025) is already a pre-trained foundation model, this stage is a supervised initialization rather than pre-training from scratch. The pipeline is instantiated separately for each EmbodiedBench (Yang et al., 2025) environment, as each environment has its own dataset, system prompt, action vocabulary, and evaluation protocol.

SFT Data Construction. We construct multi-task SFT datasets from the EmbodiedBench (Yang et al., 2025) trajectory corpora, which contain multi-step episodes collected by 11 frontier VLMs (GPT-4o (Hurst et al., 2024), Claude-3.5-Sonnet (Anthropic, 2024), Gemini-1.5-Pro (Team et al., 2024), *etc*.). We filter to *success-only* episodes and restrict to the evaluation subsets `base`, `complex_instruction`, and `visual_appearance` to avoid train–evaluation contamination with the held-out subtasks (`common_sense`, `spatial_relationship`).

Each SFT dataset contains two complementary task types:

Policy rows: Each row corresponds to one planning step from a successful trajectory. The prompt contains environment-specific system instructions, the current observation image, the task instruction, the action history from prior steps, and the numbered action space. The target response follows the evaluation-time JSON schema with four fields: `visual_state_description`, `reasoning_and_reflection`, `language_plan`, and `executable_plan` (a list of {`action_id`, `action_name`} pairs). Only steps where all actions succeed (`action_success` = 1.0) are included. Action history is threaded across steps so that the model learns sequential reasoning from prior interactions.

Verifier rows: These rows teach the model to judge the proposed action plans from policy $\mathcal{M}_\theta^p$. The prompt presents the same observation context together with a candidate plan, and the target response is a verdict JSON: {``verdict'': ``PASS'', ``explanation'': ...} or {``verdict'': ``FAIL'', ``explanation'': ...}. Both successful and failed steps are included so the model observes negative examples.

To balance the two task types, the minority task type is oversampled to a roughly 50/50 split. The combined dataset is then shuffled with a fixed seed.

SFT Training Configuration. We fine-tune only the language-model weights of Qwen3-VL-2B-Instruct (Bai et al., 2025) while keeping the visual encoder frozen. Tab. 9 lists the key hyperparameters.

Table 9: SFT training hyperparameters.

Parameter	Value
Base model	Qwen3-VL-2B-Instruct (Bai et al., 2025)
Training epochs	3
Global batch size	32
Micro batch size per GPU	8
Learning rate	2×10^{-5}
Max sequence length	4,096 tokens
Visual encoder	Frozen
Framework	FSDP (verl)

D.4 PRETRAINING OF IMAGINATION MODEL $\mathcal{W}_{\phi_{\text{REF}}}$

Unlike the VLM, which is already pre-trained and only warm-started (Sec. D.3), the imagination model $\mathcal{W}_{\phi_{\text{ref}}}$ is pre-trained from scratch on top of a frozen V-JEPA2 (Assran et al., 2025) encoder. It learns to predict the next visual state given the current state and a natural-language action, operating entirely in latent space. Training is split into two stages: **Stage 1** trains the core latent predictor, and **Stage 2** trains a diffusion decoder that converts latent predictions back to pixel-space images (Sec. C.2).

Training Data. We construct trajectory datasets from EmbodiedBench (Yang et al., 2025) environments. For each environment, a JSONL file stores individual transitions of the form (o_t, a_t, o_{t+1}), where o_t and o_{t+1} are RGB observation images and a_t is a natural-language action description (*e.g.*, "navigate to the TV stand" or "Move gripper to $(33, 43, 27)$ with rotation $(0, 60, 90)$, gripper open"). During training, consecutive transitions are grouped into multi-step sequences using a sliding window of length $W{=}4$ (4 unique states, 3 transitions), with a continuity check ensuring that o_{t+1} of each entry matches o_t of the next.

Stage 1: Action-Conditioned Next Visual State Prediction. The goal of stage 1 is to make the Imagination Model $\mathcal{W}_{\phi_{\text{ref}}}$ learn an action-conditioned next-state predictor in V-JEPA2 (Assran et al., 2025) latent space. In addition, we use a projector to align with the VLM vision encoder (Qwen3-VL-2B-Instruct (Bai et al., 2025)). The projector is a single linear layer mapping from V-JEPA2 (Assran et al., 2025) latent space (1408-dim) to the concatenated VLM embedding space (8192-dim), comprising the main PatchMerger output and three DeepStack feature channels extracted from intermediate ViT layers 5, 11, and 17 ($2048 \times (1 + 3) = 8192$).

The Imagination Model $\mathcal{W}_{\phi_{\text{ref}}}$ is trained with a composite loss:

$$\mathcal{L}_{\text{Stage1}} = \mathcal{L}_{\text{tf}} + \mathcal{L}_{\text{ro}} + \mathcal{L}_{\text{align}}, \tag{7}$$

where the three terms are defined as follows:

Teacher-forcing loss ($\mathcal{L}_{\text{tf}}$): Predicts the next state from the ground-truth current-state encoding and measures the ℓ_1 distance against the EMA encoder target (with decay $\rho{=}0.996$):

$$\mathcal{L}_{\text{tf}} = \left\| \mathcal{W}_{\phi_{\text{ref}}}(\mathbf{z}_t, \mathbf{a}_t) - \text{EMA}(o_{t+1}) \right\|_1. \tag{8}$$

Rollout loss ($\mathcal{L}_{\text{ro}}$): Performs autoregressive multi-step prediction, feeding each predicted state as input to the next step, and averages the ℓ_1 error over all transitions:

$$\mathcal{L}_{\text{ro}} = \frac{1}{W-1} \sum_{t=0}^{W-2} \left\| \hat{\mathbf{z}}_{t+1} - \text{EMA}(o_{t+1}) \right\|_1, \quad \hat{\mathbf{z}}_{t+1} = \mathcal{W}_{\phi_{\text{ref}}}(\hat{\mathbf{z}}_t, \mathbf{a}_t). \tag{9}$$

This loss penalizes error accumulation, which is critical for multi-step planning.

VLM alignment loss ($\mathcal{L}_{\text{align}}$): Aligns projected predictions with features from the frozen vision encoder of the VLM (Qwen3-VL-2B-Instruct (Bai et al., 2025)), including DeepStack outputs from ViT layers 5, 11, and 17:

$$\mathcal{L}_{\text{align}} = 1 - \cos\big(g_\phi(\hat{\mathbf{z}}_{t+1}),\ \mathbf{h}_{\text{vlm}}(o_{t+1})\big), \tag{10}$$

where g_ϕ is the projector ($1408 \to 8192$) and $\mathbf{h}_{\text{vlm}} \in \mathbb{R}^{64\times8192}$ is the concatenation of the main and three DeepStack feature channels of the frozen vision encoder.

Stage 2: Diffusion Decoder for Predicted State Visualization. Stage 2 trains a diffusion decoder to render the predicted latent states as pixel-space images, using the SD3.5-Medium architecture (Esser et al., 2024). The entire Stage 1 model is frozen; only the **latent-to-pixel projector** (a convolutional module mapping 1408-dim latent patches to a $16{\times}32{\times}32$ spatial latent map) and the diffusion decoder receive gradients. The training objective is the standard diffusion denoising loss:

$$\mathcal{L}_{\text{Stage2}} = \mathbb{E}_{\epsilon,t} \left\| \epsilon - \epsilon_\psi\big(\text{Proj}_{\text{pix}}(\hat{\mathbf{z}}_{t+1}),\ t\big) \right\|_2^2, \quad \hat{\mathbf{z}}_{t+1} = \mathcal{W}_{\phi_{\text{ref}}}(\hat{\mathbf{z}}_t, \mathbf{a}_t). \tag{11}$$

Training Hyperparameters. Both stages use the AdamW optimizer with layered learning rates: 1×10^{-5} for the pre-trained diffusion decoder in Stage 2 and 1×10^{-4} for randomly initialized components (Projector and V-JEPA2 (Assran et al., 2025) Predictor in Stage 1; latent-to-pixel projector in Stage 2), with the V-JEPA2 (Assran et al., 2025) encoder frozen throughout. Training is conducted for 5 epochs per environment with a per-GPU batch size of 16, 5% linear warmup followed by constant learning rate, weight decay of 0.01, and gradient clipping at max norm 1.0.