VisionCreator-S1: Evolving Visual-Generation Agents via Skill-GRPO Optimization

Jinxiang Lai[1*], **Wenzhe Zhao**[2*], **Hualei Zhang**[1*], **Zexin Lu**[2],
Jian Liu[1], **Jie Zhang**[1], **Qinglin Lu**[2], **Song Guo**[1§]
[1]Hong Kong University of Science and Technology, [2]Tencent
[*]Equal contribution, [§]Corresponding author: songguo@cse.ust.hk

Abstract

Native visual-generation agents now handle long-horizon, multi-image workflows, but face two practical bottlenecks during reinforcement learning (RL) training. First, the inability to accumulate experience across rollouts prevents the agent from leveraging past successes and failures, slowing the evolution of its agent behaviors. Second, the reliance on black-box visual-generation tools introduces tool-induced reward noise, as the policy is unfairly penalized for failures rooted in a tool's inherent capability boundary, leading to unstable and inefficient training. To address these challenges, we propose **VisionCreator-S1**, an agent that casts skill-augmented RL as a nested optimization problem through two coupled innovations. First, **Dual-Skills** co-evolves an *agent-skill* set to reuse self-reflection patterns from historical rollouts, and a *tool-skill* set that dynamically maps each tool's capability boundary through world-reflection. Second, **Skill-GRPO** treats discrete text skills as learnable variables. To overcome the non-differentiability of text skills, it gracefully alternates between numerical gradient descent for the policy weights and *textual gradient descent* for the discrete skill sets. By contrasting high- and low-advantage trajectories, it approximates a direction-consistent textual gradient in the agent's behavior feature space. Extensive experiments on VCR-Bench, GEdit-Bench, and MultiBanana demonstrate that VisionCreator-S1 consistently outperforms the skill-free baseline VisionCreator-R1.

1 Introduction

Visual generation has moved quickly from single-image models [1–4] to long-horizon multi-image workflows run by agents [5–9]. Native visual-generation agents, such as VisionCreator [10] and VisionCreator-R1 [11], tie Understanding, Thinking, Planning, Creation, and Reflection (UTPCR) into an end-to-end trainable framework and set the current *plan-driven, reflective* state of the art.

Despite these advances, existing native visual-generation agents face two practical bottlenecks that hinder their continuous evolution. First, the absence of a cross-rollout experience accumulation mechanism prevents these agents from leveraging past successes and failures. As a result, they struggle to iteratively enhance their UTPCR decision-making or adapt effectively to new tasks. Second, as highlighted in VisionCreator-R1 [11], the reliance on black-box visual-generation tools introduces severe environmental noise. External tools such as text2image possess inherent capability boundaries (e.g., struggling to render embedded text). A generation failure in such cases stems from the tool's limitations rather than a flaw in the agent's planning or reflection. When this tool-induced reward noise is naively back-propagated, the policy model is penalized for errors outside its control, leading to unstable and inefficient training.

To jointly address these two limitations, we propose **Dual-Skills**, a complementary pair of learnable skill sets consisting of *agent-skills* and *tool-skills*, which are attached to the agent and evolved

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

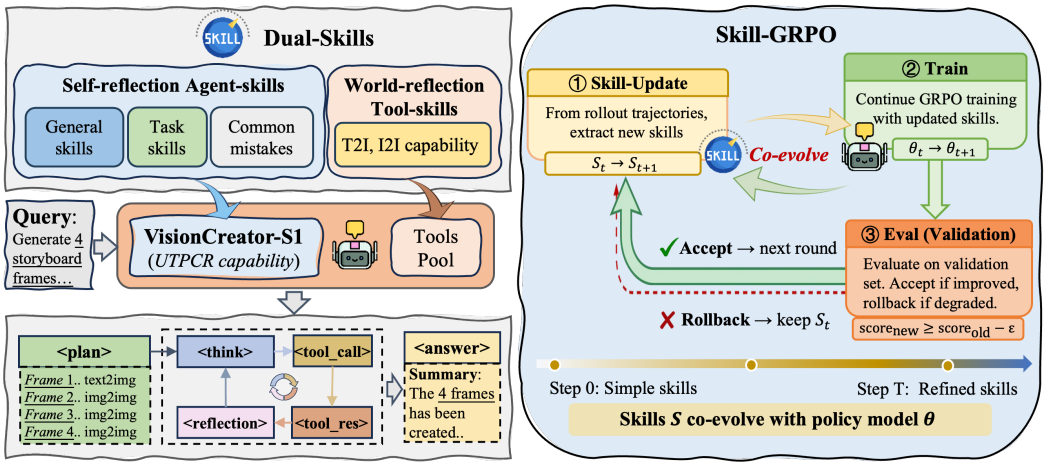

Figure 1: VisionCreator-S1 integrates Dual-Skills, and they are co-evolved by Skill-GRPO.

alongside the policy. *Agent-skills* empower the visual generation agent to summarize its own historical UTPCR rollouts into reusable self-reflection patterns. This allows the agent to efficiently learn from experience, improving its decision-making performance at each cognitive step. *Tool-skills*, conversely, enable the agent to gradually record each tool's capability boundary, providing an accurate "world reflection". By recognizing what tools can and cannot do, the agent can proactively circumvent tool-level deficiencies (e.g., routing a query to an alternative strategy if the default tool is notoriously weak at that specific task). This critical capability ensures that inevitable tool failures are not misattributed to the agent's reasoning, effectively decoupling environmental noise from the reward signal and stabilizing the policy training.

However, determining how to update these discrete text skills alongside the continuous policy model presents a significant optimization challenge. Because traditional numerical backpropagation is fundamentally incompatible with discrete, non-differentiable text variables [12], existing methods (e.g., SkillRL [13]) are forced to update skills completely outside the RL loop using heuristic teachers derived only from failure trajectories. Consequently, the policy model and the skill library lack formal alignment and are prone to drifting or oscillating against each other during joint training. To address this problem, we propose **Skill-GRPO**, a nested optimization framework that treats the discrete text skill set as a learnable variable within the RL loop, evaluated alongside the policy model. To overcome the non-differentiability of text without falling back to unaligned heuristics, Skill-GRPO approximates a direction-consistent "textual gradient" by contrasting high-advantage and low-advantage trajectories in the agent's behavior space. In this way, Skill-GRPO gracefully alternates between traditional numerical gradient descent for updating the policy weights, and textual gradient descent for updating the skill set.

In summary, our contributions are: (i) We propose **Dual-Skills**, which lifts skills to agent- and tool-level learnable parts and lets the visual-generation agent jointly perform self-reflection on its own UTPCR behavior and world reflection on each tool's capability boundary. (ii) We propose **Skill-GRPO**, a nested optimization framework that enables the rigorous co-evolution of the continuous policy model and discrete text skills. By treating discrete skills as learnable variables, Skill-GRPO avoids unaligned offline heuristics and instead computes a direction-consistent "textual gradient" by contrasting high- and low-advantage trajectories. This ensures that discrete skill edits are formally aligned with the GRPO policy gradient, facilitating stable and synergistic joint training. (iii) On VCR-Bench [11], GEdit-Bench [14], and MultiBanana [15], our VisionCreator-S1 consistently improves over the skill-free VisionCreator-R1 [11], and ablation studies show each component is effective.

2 Related Work

Visual-generation agents. Visual generation has moved from single-image models [1–4] to long-horizon multi-image and video workflows [5–9, 16, 17]. Native visual agents such as VisionCreator [10] and VisionCreator-R1 [11] push this line to an end-to-end trainable UTPC(R) framework and

set the current state of the art on long-horizon visual generation. None of these agents accumulates experience across rollouts or models the capability boundary of its black-box tools, which is the bottleneck we target by lifting both into learnable skills.

Skill learning for agents. A parallel line of work evolves trajectory-level experience into reusable skills for language or GUI agents [13, 18–22]. Our Skill-GRPO differs from these recipes on two axes: *dual-skills evolution* and a *four-phase S-step run inside the RL loop*. Methodologically, Skill-GRPO can also be viewed as embedding a textual-gradient-descent step [12] into GRPO's outer loop, where LLM-produced natural-language feedback serves as a *textual gradient* for the discrete skill S. This keeps the textual update on S formally aligned with the policy gradient on θ, rather than decoupled from it.

3 Methodology

3.1 Preliminaries

UTPCR agent. As shown in Fig. 1, we follow the UTPCR workflow of VisionCreator-R1 [11]: `<plan>` → `<think>` → `<tool_call>` → `<tool_res>` → `<reflection>` → `<answer>`. The agent is built on a VLM (Qwen3-VL [23]) that calls two black-box tools via standard APIs, `text2image` and `image2image`; a single query typically unrolls into a UTPCR trajectory of multiple steps.

Standard GRPO. Formally, let π_θ be the policy parameterized by θ. For a trajectory τ sampled from π_θ, let $R(\tau)$ denote its associated reward. Standard GRPO [24] maximizes the objective $J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)]$ by utilizing group-normalized advantages $A(\tau)$ derived from a batch of sampled outputs, while constraining the policy update within a trust region via a KL divergence penalty against a reference policy π_{ref}.

3.2 Dual-Skills Evolution

3.2.1 Dual-Skills Design

To address the two bottlenecks of Section 1, we design **Dual-Skills** (Fig. 1), learnable skill sets attached to the UTPCR agent: *agent-skills* let the agent learn from its own history, and *tool-skills* record each tool's capability boundary to mitigate the risk of misattributing tool failures.

Agent-skills (S_{agent}, *self-reflection*). To strengthen the agent's self-reflection ability, we summarize reusable patterns from past rollouts and inject them back into the agent. S_{agent} is organized into three retrievable skill categories: *(i) General skills* capture cross-task decision heuristics that govern how the agent plans and iterates regardless of the task type; *(ii) Task skills* encode task-specific generation strategies so that the agent can specialize its workflow for different visual-generation scenarios; *(iii) Common mistakes* record frequent failure patterns and their fixes, enabling the agent to preemptively avoid known pitfalls during planning. Besides, rather than compressing an entire trajectory into a single unstructured skill (*naive skill*), we extract a dedicated skill entry for each UTPCR sub-block, so that the skill format structurally mirrors the agent's own UTPCR workflow and each sub-block can directly retrieve corresponding experience.

Tool-skills (S_{tool}, *world reflection*). To strengthen the agent's perception of its tool world, we replace static tool descriptions with an evolvable capability map that tracks each tool's strengths, weaknesses, and recommended practices based on past interaction rollouts. This mechanism allows the agent to internalize the objective boundaries of black-box tools, which effectively mitigates the propagation of tool-induced reward noise into the policy gradients. By explicitly recognizing the tool capability, the agent can proactively adjust its decision-making to bypass these limitations. For instance, once the profile records that `text2image` is weak at rendering embedded text, the agent can route similar queries through an `image2image` overlay strategy at tool selection time, avoiding first-step tool mismatches at the source and decoupling tool-induced noise from the reward signal – a structural gain that agent-level skills cannot reach downstream.

3.2.2 Nested Optimization with Dual-Skills as a Learnable Variable

The above design specifies *what* agent-skills and tool-skills should contain; for them to actually *evolve*, we expose the pair as a single learnable variable $S = (S_{\text{agent}}, S_{\text{tool}})$ that the RL loop can

update alongside θ. The policy of Section 3.1 is lifted from $\pi_\theta(\tau \mid q)$ to $\pi_\theta(\tau \mid q, S_{\text{agent}}, S_{\text{tool}})$. We make this lift because leaving the skill set as a static prompt inherently disconnects the agent's skill and tool-knowledge from its actual training objective. By promoting this conditioning text into a learnable variable, we allow the skill set to be explicitly shaped by the reward signal, establishing a concrete joint optimization target.

Consequently, updating θ and S in the same RL loop turns the original single-variable GRPO problem of Section 3.1 into the nested optimization

$$\max_{\theta} J\big(\theta,\, S^*(\theta)\big), \qquad S^*(\theta) \;=\; \arg\max_{S} \mathbb{E}_{\tau\sim\pi_\theta(\cdot\mid S)}[R(\tau)], \tag{1}$$

whose total gradient with respect to θ carries a coupling term that standard GRPO silently drops:

$$dJ/d\theta \;=\; \underbrace{\partial J/\partial\theta|_{S}}_{\text{GRPO computes}} \;+\; (\partial J/\partial S)\cdot(dS^*/d\theta). \tag{2}$$

Recovering this term requires an estimator for $\partial J/\partial S$. However, S is a discrete text sequence; standard numerical gradients are undefined. While existing skill-learning recipes circumvent this by falling back to unaligned, failure-only teacher heuristics, which introduce severe sampling, signal, and direction biases, we instead propose approximating $\partial J/\partial S$ via a textual gradient [12]. Formally, alternating between numerical θ-updates and S-updates constitutes a block coordinate ascent (BCA). For this joint training to converge steadily rather than drift or oscillate, the textual gradient update must satisfy three rigorous mathematical conditions: (C1) the S-update is direction-consistent with $\partial J/\partial S$; (C2) the S-step is bounded; (C3) the S-space admits a behavioral distance with local Lipschitz control. The next section details how Skill-GRPO explicitly designs one phase against each condition.

3.3 Skill-GRPO

Skill-GRPO turns the coupling term in Eq. (2) into an executable algorithm that alternates numerical gradient descent on the policy weights θ with *textual gradient descent* [12] on the discrete skill set S. It has two layers: an outer *SkillUpdate–Train–Eval* loop that co-evolves skills with the policy inside RL (Section 3.3.1), and an inner four-phase *S-step* that makes the lower-level textual update meet (C1)–(C3) one at a time (Section 3.3.2). The full procedure is summarized in Algorithm 1.

3.3.1 Skills Co-Evolve with Policy Model

As illustrated in Fig. 1, we design the RL loop so that skills and the policy model co-evolve along two complementary directions. In one direction, injecting better skills into the prompt strengthens the policy's decision-making, enabling it to produce higher-reward rollouts. In the other direction, richer rollouts let the S-step summarize more informative skills. To ensure that newly generated skills genuinely improve the policy rather than introduce textual noise, we introduce a strict Eval stage: after each skill update, a held-out validation set evaluates the joint (θ, S); if performance improves, the new skills are accepted, otherwise they are rolled back. The Eval stage acts as a safety gate that prevents noisy or regressive skill edits from propagating into subsequent training, keeping the co-evolution monotonically improving in practice.

Formally, this loop realizes the nested optimization objective of Eq. (1) through a BCA schedule. Specifically, numerical gradient descent is performed for N_{val} GRPO steps to update the policy weights θ, followed by a single S-step that applies *textual gradient descent* to update the discrete skill set S. This alternation ensures that both variables are optimized toward the same reward objective while respecting the non-differentiability of S.

3.3.2 The Four-Phase S-Step

Each S-step runs four phases whose roles map directly onto the BCA conditions of Section 3.2: Phase A estimates a *textual gradient* [12] in a behavior feature space (C1); Phase B adds structural and semantic constraints on the skill edit (a representation-level surrogate for C2); Phase C adds a single-step behavioral safety net (the sampling surrogate that covers the behavioral half of C2 and C3); and Phase D adds a validation-failure signal to the training-set textual gradient to close the distribution gap. Phases A, B, and C are always on and form the core procedure, while Phase D is toggled by a flag `D` for ablation.

Phase A: textual gradient in behavior feature space. Since $\partial \log \pi_\theta / \partial S$ is ill-defined on discrete tokens, we follow the textual-gradient paradigm of Yuksekgonul et al. [12] and replace the unrealizable symbolic gradient with a prompt-based estimator – a *textual gradient* that specifies, in natural language, how S should be modified. Under Assumption A.3, this estimator is direction-consistent with $\partial J / \partial S$ in the behavior feature space. Let $\phi(\tau) \in \mathbb{R}^{d_\phi}$ be a vector of trajectory-level behavioral indicators (tool-choice appropriateness, prompt completeness, reflection specificity, etc.) pulled from a VLM summary of τ (with $d_\phi = 8$). Under group-normalized advantages, $\mathbb{E}_\tau[A(\tau)] = 0$, so we define the *textual gradient* in the ϕ-space

$$\nabla_\phi J \;:=\; \mathbb{E}_{\tau \sim \pi_\theta(\cdot \mid S)}\big[A(\tau)\,\phi(\tau)\big] \;=\; \mathbb{E}_{A>0}\big[|A(\tau)|\phi(\tau)\big] \;-\; \mathbb{E}_{A<0}\big[|A(\tau)|\phi(\tau)\big]. \tag{3}$$

Advantages act as quantitative weights, both positive and negative trajectories enter the estimator, and the direction is formally aligned with $\partial J / \partial S$ in the feature space. In practice, $\nabla_\phi J$ is not computed analytically: the summaries of the top-K positive and top-K negative trajectories are fed into a contrastive prompt, and a VLM emits a candidate skill set

$$C_{\mathrm{A}} \;=\; \mathrm{PhaseA}\big(\{(\phi(\tau_k^+), A(\tau_k^+))\}_{k=1}^K,\; \{(\phi(\tau_k^-), A(\tau_k^-))\}_{k=1}^K\big), \tag{4}$$

whose direction matches $\nabla_\phi J$ by construction of the prompt. These rollouts are already collected by the θ-step, so Phase A adds zero extra *rollout* cost.

Phase B: structural and semantic constraints on the skill edit. Phase B filters the merged candidate set $C = C_{\mathrm{A}} \cup C_{\mathrm{D}}$ by a pair of computable surrogate constraints on the skill edit:

$$C_{\mathrm{B}} \;=\; \mathrm{PhaseB}(C; K_{\max}, \delta_{\text{sim}}) \;=\; \big\{\, c \in C :\; |\Delta S| \le K_{\max} \text{ and } \max_{c' \in S_t} d_{\text{sim}}(c, c') \le \delta_{\text{sim}} \,\big\}, \tag{5}$$

where $|\Delta S| \le K_{\max}$ caps the number of new skills per layer per round, d_{sim} is the cosine similarity between sentence embeddings, and intra-batch duplicates within C_{B} are resolved by the same threshold. Phase B bounds the *representation* of the skill edit, not the induced behavioral distribution; the behavioral bound is given by the Phase-C evaluation gate described next.

Phase C: evaluation gate. The structural and semantic constraints of Phase B do not rule out single-step regressions under heavy distribution shift or teacher hallucination. Phase C closes this gap with a lightweight validation rollout on the merged candidate $S_{\text{cand}} = S_t \oplus C_{\mathrm{B}}$:

$$S_{t+1} = \mathrm{PhaseC}(S_t, S_{\text{cand}}; \varepsilon) = \begin{cases} S_{\text{cand}}, & \mathrm{Eval}(\pi_\theta, S_{\text{cand}}) \ge \mathrm{Eval}(\pi_\theta, S_t) - \varepsilon, \\ \textsc{Shrink}(S_t, S_{\text{cand}}) \text{ or } S_t, & \text{otherwise.} \end{cases} \tag{6}$$

The SHRINK branch keeps the strongest top-1 skill and falls back to S_t if the check still fails. This is the most direct sampling surrogate for (C3): the validation-score change projects the underlying behavior-distribution change onto the downstream task. In practice we observe $p_{\max} \le 0.1$ for the Phase-C rejection rate across our runs, so the gate rarely triggers but catches exactly the pathological edits that Phase B cannot see.

Phase D: failure-only supplement for train–validation coverage. Phase A is a textual-gradient estimator on μ_{train}. When the validation distribution differs by $d_{\mathrm{TV}}(\mu_{\text{train}}, \mu_{\text{val}})$, some failure modes visible only on the validation set are invisible to Phase A. Phase D produces a complementary textual-gradient candidate set from validation failure trajectories $\mathcal{F}_{\text{val}}$,

$$C_{\mathrm{D}} \;=\; \mathrm{PhaseD}(\mathcal{F}_{\text{val}}), \qquad \mathcal{F}_{\text{val}} = \{\tau \in \mu_{\text{val}} : R(\tau) < \tau_{\text{fail}}\}, \tag{7}$$

and merges it with C_{A} before Phase B deduplication; Phase A pushes toward high-reward behavior while Phase D patches the failure modes that Phase A cannot see.

The structural properties the algorithm achieves – direction consistency of the Phase A textual gradient, representation-level bounds from Phase B, the behavioral safety net from Phase C, and dual-source coverage from Phase A+D – are formalized in Section 3.4.

3.4 Theoretical Analysis

Next, we analyze the convergence of Skill-GRPO and derive a BCA-style error bound, demonstrating that the Skill-GRPO algorithm can steadily improve model capabilities within an acceptable error margin. Specifically, we show (i) that Phase A produces a direction-consistent lower-level update (C1), and (ii) together with Phases B/C/D this yields a BCA-style error bound for the joint sequence

Algorithm 1 Skill-GRPO: Skills Co-Evolve with Policy Model

Require: initial policy θ_0, seed skills $S_0 = (S^0_{\text{agent}}, S^0_{\text{tool}})$, interval N_{val}, budgets $(K, K_{\max}, \delta_{\text{sim}}, \varepsilon)$, flag `D` (Phase D on/off; Phases A, B, C are always on)

for $t = 0, 1, \ldots, T-1$ **do**
  Collect rollouts under $\pi_{\theta_t}(\cdot \mid S_t)$ and advantages $A(\cdot)$; update θ_{t+1} by GRPO
  if $(t+1) \bmod N_{\text{val}} = 0$ **then**
    $C_{\text{A}} \leftarrow \text{PhaseA}(\text{top-}K^{\pm}\text{ rollouts})$ ▷ Eq. (3)–(4); training-signal candidate skills C_{A}
    $C_{\text{D}} \leftarrow \text{PhaseD}(\mathcal{F}_{\text{val}})$ **if** `D` **else** $\varnothing$ ▷ Eq. (7); validation-failure candidate skills C_{D}
    $C_{\text{B}} \leftarrow \text{PhaseB}(C_{\text{A}} \cup C_{\text{D}};\ K_{\max}, \delta_{\text{sim}})$ ▷ Eq. (5); de-duplicated candidate set C_{B}
    $S_{\text{cand}} \leftarrow S_t \oplus C_{\text{B}}$ ▷ candidate skill library S_{cand} after inserting C_{B} into the current S_t
    $S_{t+1} \leftarrow \text{PhaseC}(S_t, S_{\text{cand}}; \varepsilon)$ ▷ Eq. (6); accepted skills S_{t+1}
  else
    $S_{t+1} \leftarrow S_t$ ▷ no S-step this round: skills stay frozen while θ keeps updating
  end if
end for
return (θ_T, S_T)

$\{(\theta_t, S_t)\}$. All textual-gradient objects below live in the behavior feature space ϕ introduced in Definition 3.1; full proofs, the Linear Feature Assumption, and the three-level skill metrics are in Appendix A.

Definition 3.1 (Behavioral feature and textual gradient)**.** *Let $\phi(\tau) \in \mathbb{R}^{d_\phi}$ be a vector of language-describable behavioral indicators of τ (realized by an eight-dimensional VLM summary, so $d_\phi = 8$). We define the* textual gradient *in the behavior feature space as $\nabla_\phi J(S) := \mathbb{E}_{\tau \sim \pi_\theta(\cdot \mid S)}[A(\tau)\, \phi(\tau)]$ (Eq.* (3)*); this is an advantage-weighted, feature-space instantiation of the textual gradients introduced in Yuksekgonul et al. [12] and admits the positive-minus-negative decomposition used by the Phase A contrastive prompt.*

Proposition 3.2 (Direction consistency of Phase A)**.** *Under the Linear Feature Assumption with signal-to-noise ratio $\rho > 1$ (Appendix A.2), the Phase-A update ΔS_A satisfies $\cos \angle(\Delta S_A, \partial J / \partial S) \geq (\rho - 1)/(\rho + 1) > 0$, establishing condition (C1) of Theorem 3.5.*

Proposition 3.3 (Representation-level bound and behavioral safety net)**.** *Phase B (Eq.* (5)*) enforces $\|S_{t+1} - S_t\|_{\textit{struct}} \leq K_{\max}$ with semantic deduplication (necessary representation-level surrogate for (C2)); Phase C (Eq.* (6)*) enforces $J(\theta, S_{t+1}) \geq J(\theta, S_t) - \varepsilon$ up to validation-sampling error (sufficient behavioral surrogate for (C2)/(C3)).*

Phase A estimates the textual gradient on μ_{train}; the gap to the ideal μ_{val} target scales as $L_\phi \cdot d_{\text{TV}}(\mu_{\text{train}}, \mu_{\text{val}})$, which is the dominant term in $\varepsilon_{\text{approx}}$ below. Phase D shrinks it by a convex combination with validation-set candidates:

Theorem 3.4 (Dual-source fusion bound)**.** *Let $\widehat{\nabla}^{\textit{S-GRPO}}_{\phi} J$ be the semantically deduplicated merger of Phase A and Phase D candidates. Then $\varepsilon^{\textit{S-GRPO}}_{\textit{approx}} \leq (1-\alpha)\, \varepsilon^{\textit{A-only}}_{\textit{approx}} + \alpha\, \varepsilon_{\textit{val-sampling}}$, where $\alpha \in (0,1)$ is the Phase-D fraction and $\varepsilon_{\textit{val-sampling}} = O(1/\sqrt{|\mathcal{F}_{\textit{val}}|})$. The fusion gain grows with $d_{\text{TV}}(\mu_{\textit{train}}, \mu_{\textit{val}})$.*

Theorem 3.5 (BCA error bound for Skill-GRPO)**.** *Under (C1) Phases A+D direction consistency with error $\varepsilon^{\textit{S-GRPO}}_{\textit{approx}}$, (C2) Phase B representation bound combined with Phase C (rejection rate $p_{\max}$) giving a behavioral step bounded up to $p_{\max}\varepsilon$, (C3) Phase C single-step safety net, and (C4) upper-level GRPO sufficient ascent with residual $\varepsilon_\theta = O(1/\sqrt{N_{\textit{group}}})$ in the group size $N_{\textit{group}}$ used by GRPO, the joint sequence $\{(\theta_t, S_t)\}_{t=0}^{T-1}$ satisfies*

$$J(\theta_T, S_T) \;\geq\; J(\theta_0, S_0) + \sum_{t=0}^{T-1} \big[\rho_\theta \|\nabla_\theta J\|^2 + \rho_S \|\nabla_\phi J\|^2\big] - T\big(\varepsilon^{\textit{S-GRPO}}_{\textit{approx}} + p_{\max}\varepsilon + \varepsilon_\theta\big), \qquad (8)$$

where $\rho_\theta, \rho_S > 0$ are step-size constants inherited from the per-level sufficient-ascent lemmas (App. A.5). On a fixed validation set, $\varepsilon^{\textit{S-GRPO}}_{\textit{approx}}$ bottoms out at $\alpha\, \varepsilon_{\textit{val-sampling}} = O(1/\sqrt{|\mathcal{F}_{\textit{val}}|})$; Eq. (8) *therefore quantifies near-monotone behavior with an explicit error floor, not strict convergence to a first-order stationary point.*

Each phase plays a distinct role in the bound: Phase A supplies the direction-consistent main term, Phase D contracts the distribution-shift term, Phase B gives a necessary representation-level bound,

Table 1: Performance on VCR-Bench under two independent VLM judges (Gemini-2.5 Pro and Qwen3-VL-235B) and their average.

# Method	Gemini-2.5 Pro Judge			Qwen3-VL-235B Judge			Average		
	Single	Multi	Img2img	Single	Multi	Img2img	Single	Multi	Img2img
1 Qwen-Image-Fast [25]	0.505	0.480	0.672	0.795	0.561	0.632	0.650	0.521	0.652
2 Qwen3-VL-32B [23]	0.500	0.601	0.806	0.835	0.717	0.740	0.668	0.659	0.773
3 Gemini-2.5 Pro [26]	0.515	0.649	0.816	0.844	0.783	0.737	0.680	0.716	0.777
4 VisionCreator-R1 [11]	0.532	0.700	0.836	0.858	0.832	0.750	0.695	0.766	0.793
5 SkillRL-VG	0.528	0.697	0.834	0.855	0.829	0.748	0.692	0.763	0.791
6 S1 w/ failure-only (sub-block)	0.548	0.711	0.844	0.877	0.845	0.758	0.713	0.778	0.801
7 S1 w/ Dual-Skills + failure-only	0.550	0.717	0.852	0.879	0.852	0.764	0.715	0.784	0.808
8 S1 w/ Skill-GRPO w/o Phase D	0.554	0.706	0.848	0.885	0.844	0.763	0.720	0.775	0.806
9 S1 w/ Skill-GRPO (full, ours)	**0.560**	**0.725**	**0.860**	**0.897**	**0.861**	**0.772**	**0.728**	**0.793**	**0.816**

Phase C gives the sufficient behavioral safety net, and (C4) is inherited from standard GRPO. Skill-GRPO without Phase D meets (C1)–(C4) but inflates $\varepsilon_{\text{approx}}$ whenever $\mu_{\text{train}} \neq \mu_{\text{val}}$; full Skill-GRPO reaches the tightest bound in Eq. (8). Two quantitative checks support the bound. First, under the linear-feature assumption, the Phase-A direction satisfies $\cos\angle(\Delta S_{\text{A}}, \partial J/\partial S) \geq (\rho-1)/(\rho+1)$ whenever the signal-to-noise ratio $\rho > 1$ (Appendix A.2); on our trajectories $\hat{\rho} = 3.1$, so $\cos\angle \geq 0.51$. Second, with the measured $p_{\max} = 0.086$ and $\varepsilon = 0.01$, the safety-net term $p_{\max}\varepsilon = 8.6 \times 10^{-4}$ is an order of magnitude smaller than the gains in Table 1. Full proofs are in Appendix A.

4 Experiments

4.1 Experimental Setup

We follow the evaluation protocol of VisionCreator-R1 [11] on **VCR-Bench**, reporting scores on three regimes, including Single-image, Multi-image, and Image-to-Image editing – under two independent VLM judges (Gemini-2.5 Pro and Qwen3-VL-235B) plus their average. We additionally evaluate on the **GEdit-Bench** [14] and the multi-reference benchmark **MultiBanana** [15].

The gain of VisionCreator-S1 is set up against VisionCreator-R1 and SkillRL-VG, which share the same backbone and tool. Outside this controlled scope we include Qwen-Image-Fast (Qwen-Image-Edit-2509-Lightning-4steps-V1.0) [25], Qwen3-VL-32B [23], and Gemini-2.5 Pro [26] under the same UTPCR system prompt. We introduce SkillRL-VG, a migration of SkillRL [13] into visual generation (naive agent-skills only, failure-only updates). VisionCreator-S1 starts its RL from VisionCreator-SFT following R1's RPCO pipeline. Default Skill-GRPO hyperparameters: top-$K = 5$, $N_{\text{val}} = 20$, $K_{\max} = 5$, $\delta_{\text{sim}} = 0.85$, $\varepsilon = 0.01$ (Phase C on), Phase D on in `full`.

4.2 Results on VCR-Bench

Table 1 reports the main VCR-Bench comparison. VisionCreator-S1 with full Skill-GRPO (row 9) ranks first in every one of the nine cells defined by three task categories and two judges plus their average, and the progressive chain from row 4 to row 9 shows that every structural fix adds a visible gain. Relative to the strongest skill-free VisionCreator-R1, VisionCreator-S1 improves by $+0.028$ / $+0.025$ / $+0.024$ under the Gemini judge and by $+0.039$ / $+0.029$ / $+0.022$ under the Qwen3-VL-235B judge on Single / Multi / Img2img, with consistent direction under both judges.

Mechanistic reading of the progressive chain. Reading the chain row by row exposes four mechanistic insights that echo Section 3.3. First, sub-block summarization alone (row 6 vs row 5) already yields $+0.022$ / $+0.016$ / $+0.010$ on the Qwen3-VL-235B judge, confirming that whole-trajectory summarization severely dilutes useful patterns when trajectories exceed $10+$ steps and that structured per-UTPCR-sub-block skills are a dominant improvement. Second, adding tool-skills (row 7 vs row 6) gives an extra $+0.002$ / $+0.007$ / $+0.006$, with the strongest relative effect on image editing, in line with the intuition that tool capability boundary awareness dominates editing tasks. Third, replacing failure-only with Phase A+B (row 8 vs row 7) lifts Single by $+0.006$ but regresses on Multi by -0.008, showing that a single-source textual gradient is limited by train–validation

Table 2: Results on MultiBanana (10-point scale).

Model	Single Obj	Two Obj	X Obj	X-1+Local	X-1+Global	X-1+BG
Nano Banana	7.817	4.891	4.453	4.118	4.698	3.575
GPT-Image-1	7.804	**6.585**	**5.086**	**5.147**	**5.757**	**5.019**
Qwen-Image-Fast	7.276	3.489	2.067	2.001	2.260	2.011
+ VisionCreator-R1	7.891	4.834	4.133	3.914	4.062	3.603
+ **VisionCreator-S1**	**7.928**	5.106	4.347	4.243	4.508	3.945

distribution shift on long-horizon narratives – exactly the phenomenon that motivates Phase D. Finally, adding Phase D (row 9 vs row 8) yields $+0.012$ / $+0.017$ / $+0.009$ and an empirical match to the dual-source fusion prediction of Theorem 3.4.

Reverse-direction cells. Two cells in Table 1 move against the overall trend. (i) SkillRL-VG (row 5) is marginally below VisionCreator-R1 (row 4) in all three categories (-0.004/ -0.003/ -0.002 under Gemini). We read this as empirical confirmation that a *direct* SkillRL migration does not help visual generation – whole-trajectory summarization dilutes decisive sub-blocks, and failure-only updates emit no tool-level guidance – which in turn motivates per-UTPCR-sub-block skills summarization and Dual-Skills. (ii) Skill-GRPO without Phase D (row 8) is 0.011 below Dual-Skills+failure-only (row 7) on Multi, since removing the validation-failure supplement leaves long-narrative distribution shift uncompensated; adding Phase D (row 9) recovers and exceeds row 7.

Table 3: GEdit-Bench: semantic consistency (G_SC), perceptual quality (G_PQ), overall (G_O).

Model	G_SC	G_PQ	G_O
Gemini 2.0	6.73	6.61	6.32
BAGEL [27]	7.36	6.83	6.52
FLUX.1 Kontext Pro [28]	7.02	**7.60**	6.56
Step1X-Edit [14]	7.66	7.35	6.97
Qwen-Image-Fast	7.26	7.34	7.03
+VisionCreator-R1	7.60	7.33	7.23
+VisionCreator-S1	**7.72**	7.41	**7.39**

4.3 Results on GEdit-Bench and MultiBanana

We next check whether the gains transfer to external benchmarks. On **GEdit-Bench** [14] (Table 3), VisionCreator-S1 lifts VisionCreator-R1's $7.60/7.33/7.23$ to $\mathbf{7.72}/7.41/\mathbf{7.39}$ ($+0.12$/ $+0.08$/ $+0.16$), taking first place on both G_SC and G_O; the gain sits on queries whose appropriate tool is not the default (text rendering, layout-preserving edits), confirming that the VCR-Bench i2i improvement transfers to an independently curated editing benchmark.

On **MultiBanana** [15] (Table 2), VisionCreator-S1 improves over VisionCreator-R1 on *all six* categories by $+0.04$ to $+0.45$ points, with the largest gains on the harder compositional subsets (X-1+Global $+0.446$, X-1+BG $+0.342$, X-1+Local $+0.329$). Despite using the open-source Qwen-Image-Fast tool, VisionCreator-S1 also beats the closed-source *Nano Banana* agent on four of six categories (Single Obj, Two Obj, X-1+Local, X-1+BG).

Table 4: Training-free skill ablation on a *frozen* Qwen3-VL-32B backbone with skills injected via the system prompt only (Qwen3-VL-235B judge). Rows are: *none* (no skills); *naive skills* (SkillRL-style whole-trajectory summarization); *agent-skills* (UTPCR sub-block summarization); *Dual-Skills* (agent-skills + tool-skills).

#	Skills	Single	Multi	Img2img
0	none	0.835	0.717	0.740
1	naive skills	0.841	0.725	0.743
2	agent-skills	0.848	0.732	0.747
3	**Dual-Skills**	**0.850**	**0.741**	**0.755**

4.4 Systematic Ablation on VCR-Bench

Scores in this subsection are reported under the Qwen3-VL-235B judge on VCR-Bench.

Training-free skills: agent- vs tool-skills. We begin by decoupling "what skills bring" from "what RL brings". Freezing the agent and tools on Qwen3-VL-32B and only switching the skill set injected into the system prompt gives the training-free chain in Table 4. Whole-trajectory naive skills in the SkillRL style give a mild $+0.006$ / $+0.008$ / $+0.003$ over the no-skill baseline, confirming that whole-trajectory summarization alone caps the gain. Replacing them with per-UTPCR-sub-

Table 5: Four-phase ablation of Skill-GRPO (Qwen3-VL-235B judge on VCR-Bench). Phase C is always on and therefore not toggled here; we report its empirical rejection rate $p_{\max} \leq 0.1$ in Section 3.3. Phase B shifts the three scores only slightly (it is a representation-level constraint rather than a score contributor); Phase D delivers the largest single-component gain, consistent with the dual-source fusion bound.

#	Config	Phase A	Phase B	Phase D	Single	Multi	Img2img
0	failure-only baseline	–	–	(train-fail)	0.877	0.845	0.758
1	Phase A only	✓	–	–	0.884	0.839	0.761
2	Phase A + B	✓	✓	–	0.885	0.844	0.763
3	Phase A + B + D (**full, ours**)	✓	✓	✓	**0.897**	**0.861**	**0.772**

block agent-skills adds another $+0.007$ / $+0.007$ / $+0.004$, locating the cost in *unstructured summarization over long trajectories* rather than the summarization model itself. Adding tool-skills further adds $+0.002$ / $+0.009$ / $+0.008$, an independent gain on multi-image narratives and image editing that agent-only formulations cannot reach – consistent with the Introduction's claim that tool capability boundary awareness provides a structural gain inaccessible to agent-level self-reflection alone. Skill design alone, with no RL, therefore already moves Single / Multi / Img2img from 0.835 / 0.717 / 0.740 to 0.850 / 0.741 / 0.755.

Four phases of Skill-GRPO. Turning on RL, we then toggle Phases A, B, and D individually (Table 5). Phase A alone lifts Single by $+0.007$ over the failure-only baseline, which is direct evidence that advantage-weighted contrast fixes the direction bias of Section 3.2.2. Phase B, by contrast, shifts the three scores only slightly; this is in line with its role as a *representation-level* constraint on the skill edit rather than a terminal-score contributor, whose intended benefit is stability of the training process. Phase D adds the largest single-component gain in the paper: on top of Phase A it lifts Single by $+0.013$ and Multi by $+0.022$, the largest cell in the table and an empirical match to Theorem 3.4.

Sensitivity and super-additivity. Beyond the core four-phase toggle, two additional observations further support the design. (i) Phase A's top-K sensitivity follows the expected bias–variance trade-off of gradient estimation, with K=5 the sweet spot across the three dimensions (Table 6). (ii) Comparing the training-free Dual-Skills gain ($+0.009$ on Multi in Table 4) with the full-RL Dual-Skills gain (row 7 vs row 6 of Table 1, $+0.007$ on Multi; further boosted to $+0.016$ on Multi under full Skill-GRPO in row 9 vs row 6) reveals a super-additive interaction between agent-skills and tool-skills: the two skill sets are coupled at the policy level.

Table 6: Top-K sensitivity of Phase A (Skill-GRPO without Phase D, Qwen3-VL-235B judge). $K = 5$ balances bias and variance across the three benchmarks.

Top-K	Single	Multi	Img2img
3	0.852	0.824	0.748
5	**0.885**	**0.844**	**0.763**
10	0.882	0.838	0.752

5 Conclusion

We identify two practical bottlenecks of native visual-generation agents – the absence of cross-rollout experience accumulation and tool-induced reward noise from black-box tools with inherent capability boundaries – and trace their root cause to skill updates that are decoupled from the RL objective. To address this, we propose **Dual-Skills** (agent-skills for self-reflection on UTPCR behavior, tool-skills for world reflection on each tool's capability boundary) and **Skill-GRPO**, a nested optimization framework that co-evolves discrete text skills with the policy model by approximating a direction-consistent textual gradient inside the RL loop. Experiments on VCR-Bench, GEdit-Bench, and MultiBanana show consistent improvements over VisionCreator-R1, with each component independently validated. The nested-optimization viewpoint extends naturally to any joint training of continuous parameters and discrete text configurations.

6 Limitations and Broader Impact

Limitations. The behavior feature $\phi(\tau)$ is extracted by a frozen VLM summarizer; its quality upper-bounds the textual gradient's direction consistency.

Broader Impact. The nested-optimization idea applies to any agent coupling continuous parameters with discrete text configurations (e.g., scientific agents, code agents).

Acknowledgments and Disclosure of Funding

This research was supported by fundings from the Hong Kong RGC General Research Fund (152228/23E, 162161/24E, 162116/25E, 162180/25E), National Natural Science Foundation of China (NSFC) Key Program (No.62532005), Collaborative Research Fund (No. C1042-23GF, No. 5097-25G), NSFC/RGC Collaborative Research Scheme (Grant No. 62461160332 & CRS_HKUST602/24), Research Impact Fund (No. R5011-23), and Areas of Excellence Scheme (AoE/E-601/22-R).

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[3] B. F. Labs, "Flux," https://github.com/black-forest-labs/flux, 2024.

[4] B. Lin, Y. Ge, X. Cheng, Z. Li, B. Zhu, S. Wang, X. He, Y. Ye, S. Yuan, L. Chen *et al.*, "Open-sora plan: Open-source large video generation model," *arXiv preprint arXiv:2412.00131*, 2024.

[5] W. Wu, Z. Zhu, and M. Z. Shou, "Automated movie generation via multi-agent cot planning," *arXiv preprint arXiv:2503.07314*, 2025.

[6] J. Xiao, C. Yang, L. Zhang, S. Cai, Y. Zhao, Y. Guo, G. Wetzstein, M. Agrawala, A. Yuille, and L. Jiang, "Captain cinema: Towards short movie generation," *arXiv preprint arXiv:2507.18634*, 2025.

[7] X. Xu, J. Mei, C. Li, Y. Wu, M. Yan, S. Lai, J. Zhang, and M. Wu, "Mm-storyagent: Immersive narrated storybook video generation with a multi-agent paradigm across text, image and audio," *arXiv preprint arXiv:2503.05242*, 2025.

[8] X. Xue, Z. Lu, D. Huang, Z. Wang, W. Ouyang, and L. Bai, "Comfybench: Benchmarking llm-based agents in comfyui for autonomously designing collaborative ai systems," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 24 614–24 624.

[9] L. Guo, X. Xu, L. Wang, J. Lin, J. Zhou, Z. Zhang, B. Su, and Y.-C. Chen, "Comfymind: Toward general-purpose generation via tree-based planning and reactive feedback," *arXiv preprint arXiv:2505.17908*, 2025.

[10] J. Lai, Z. Lu, J. He, R. Quan, W. Zhao, Q. Yang, Q. Chen, Q. Lin, C. Li, T. Gao *et al.*, "Visioncreator: A native visual-generation agentic model with understanding, thinking, planning and creation," *CVPR Findings*, 2026.

[11] J. Lai, W. Zhao, Z. Lu, H. Zhang, Q. Yang, R. Quan, Z. Li, S. Shao, S. Guo, and Q. Lu, "Visioncreator-r1: A reflection-enhanced native visual-generation agentic model," *NeurIPS*, 2026.

[12] M. Yuksekgonul, F. Bianchi, J. Boen, S. Liu, L. Pan, Z. Huang, C. Guestrin, and J. Zou, "Optimizing generative ai by backpropagating language model feedback," *Nature*, 2025.

[13] P. Xia, J. Chen, H. Wang, J. Liu, K. Zeng, Y. Wang, S. Han, Y. Zhou, X. Zhao, H. Chen, Z. Zheng, C. Xie, and H. Yao, "Skillrl: Evolving agents via recursive skill-augmented reinforcement learning," *arXiv preprint arXiv:2602.08234*, 2026.

[14] S. Liu, Y. Han, P. Xing, F. Yin, R. Wang, W. Cheng, J. Liao, Y. Wang, H. Fu, C. Han *et al.*, "Step1x-edit: A practical framework for general image editing," *arXiv preprint arXiv:2504.17761*, 2025.

[15] Y. Oshima, D. Miyake, K. Matsutani, Y. Iwasawa, M. Suzuki, Y. Matsuo, and H. Furuta, "Multibanana: A challenging benchmark for multi-reference text-to-image generation," in *CVPR*, 2026.

[16] Z. Xu, Y. Wang, X. Yang, L. Wang, W. Luo, K. Zhang, B. Hu, and M. Zhang, "Comfyui-r1: Exploring reasoning models for workflow generation," *arXiv preprint arXiv:2506.09790*, 2025.

[17] O. Huang, Y. Ma, Z. Zhao, M. Wu, J. Ji, R. Zhang, Z. Hu, X. Sun, and R. Ji, "Comfygpt: A self-optimizing multi-agent system for comprehensive comfyui workflow generation," *arXiv preprint arXiv:2503.17671*, 2025.

[18] Y. Wang, X. Chen, X. Jin, M. Wang, and L. Yang, "Openclaw-rl: Train any agent simply by talking," *arXiv preprint arXiv:2603.10165*, 2026.

[19] Y. Wang, T. Xie, K. Shen, M. Wang, and L. Yang, "Rlanything: Forge environment, policy, and reward model in a completely dynamic rl system," *arXiv preprint arXiv:2602.02488*, 2026.

[20] G. Jiang, Z. Su, X. Qu, and Y. R. Fung, "Xskill: Continual learning from experience and skills in multimodal agents," *arXiv preprint arXiv:2603.12056*, 2026.

[21] Y. Yang, J. Li, Q. Pan, B. Zhan, Y. Cai, L. Du, J. Zhou, K. Chen, Q. Chen, X. Li *et al.*, "Autoskill: Experience-driven lifelong learning via skill self-evolution," *arXiv preprint arXiv:2603.01145*, 2026.

[22] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," in *arXiv preprint arXiv:2305.16291*, 2023.

[23] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge *et al.*, "Qwen3-vl technical report," *arXiv preprint arXiv:2511.21631*, 2025.

[24] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu *et al.*, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[25] Qwen, "Qwen-image-lightning," https://huggingface.co/lightx2v/Qwen-Image-Lightning, 2025.

[26] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. Rosen *et al.*, "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv preprint arXiv:2507.06261*, 2025.

[27] C. Deng, D. Zhu, K. Li, C. Gou, F. Li, Z. Wang, S. Zhong, W. Yu, X. Nie, Z. Song *et al.*, "Emerging properties in unified multimodal pretraining," *arXiv preprint arXiv:2505.14683*, 2025.

[28] B. F. Labs, S. Batifol, A. Blattmann, F. Boesel, S. Consul, C. Diagne, T. Dockhorn, J. English, Z. English, P. Esser *et al.*, "Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space," *arXiv preprint arXiv:2506.15742*, 2025.

[29] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International Conference on Machine Learning (ICML)*, 2015.

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

A Theoretical Proofs

This appendix contains the derivations and proofs supporting Section 3.4. Section A.1 establishes the score-function identity for $\partial J/\partial S$ and the variance-reduction lemma used by the textual-gradient estimator; Section A.2 introduces the quantitative Linear Feature Assumption and proves Proposition 3.2 with an explicit angle lower bound; Section A.3 defines the three-level skill metrics and proves Proposition 3.3; Section A.4 proves Theorem 3.4 and its positive-gain threshold; Section A.5 proves the sufficient-ascent lemma and Theorem 3.5; Section A.6 records the degenerate case of uniform advantages.

A.1 Score-Function Identity and Variance-Reduction Lemma

Standing notation. The policy is $\pi_\theta(\tau \mid q, S)$ with $S = (S_{\text{agent}}, S_{\text{tool}})$ and objective $J(\theta, S) = \mathbb{E}_{\tau\sim\pi_\theta(\cdot\mid S)}[R(\tau)]$. Because the GRPO advantage $A(\tau)$ is group-normalized across rollouts sharing the same query q (initial state s_0), it satisfies $\mathbb{E}[A(\tau) \mid s_0] = 0$. We write $\phi(\tau) \in \mathbb{R}^{d_\phi}$ ($d_\phi = 8$) and assume boundedness $|A(\tau)| \le A_{\max}$, $\|\phi(\tau)\| \le \Phi_{\max}$, $R(\tau) \in [0, R_{\max}]$, with $B := A_{\max}\Phi_{\max}$.

Lemma A.1 (Score-function identity for $\partial J/\partial S$)**.** *Treating S formally as a continuous conditioning argument to specify the estimand of Eq.* (2)*, the skill-gradient of the objective satisfies*

$$\frac{\partial J}{\partial S} \;=\; \mathbb{E}_{\tau\sim\pi_\theta(\cdot\mid S)}\Big[A(\tau)\,\frac{\partial \log \pi_\theta(\tau \mid S)}{\partial S}\Big].$$

Proof. This formal step only identifies the estimand; the algorithm never differentiates discrete tokens. Using $\partial\pi_\theta(\tau \mid S)/\partial S = \pi_\theta(\tau \mid S)\,\partial \log \pi_\theta(\tau \mid S)/\partial S$, we have $\partial J/\partial S = \mathbb{E}[R(\tau)\,\partial \log \pi_\theta(\tau \mid S)/\partial S]$. For any baseline $b(s_0)$ depending only on the initial state,

$$\mathbb{E}\Big[b(s_0)\,\frac{\partial \log \pi_\theta(\tau \mid S)}{\partial S}\Big] \;=\; \mathbb{E}_{s_0}\Big[b(s_0)\sum_\tau \frac{\partial \pi_\theta(\tau \mid S)}{\partial S}\Big] \;=\; \mathbb{E}_{s_0}\Big[b(s_0)\frac{\partial}{\partial S}\sum_\tau \pi_\theta(\tau \mid S)\Big] \;=\; 0.$$

Setting $b(s_0)$ to the in-group mean reward yields the group-normalized advantage $A(\tau)$ and completes the proof. □

Lemma A.1 shows that $\partial J/\partial S$ is an advantage-weighted score expectation; contrasting high- and low-advantage trajectories in Phase A is therefore a direct sampling estimator of this identity rather than an unconstrained rewrite.

Lemma A.2 (Unbiasedness under a state-dependent baseline)**.** *Let $b(s)$ be any function of the state s (independent of the action or the skill sequence) and $c(s_0) \in \mathbb{R}^{d_\phi}$ any initial-state vector baseline, and define the centered advantage $\tilde{A}(\tau) = A(\tau) - b(s_0)$. Then*

$$\mathbb{E}_{\tau\sim\pi_\theta(\cdot\mid S)}\big[\tilde{A}(\tau)\,\phi(\tau)\big] \;=\; \mathbb{E}_{\tau\sim\pi_\theta(\cdot\mid S)}\big[A(\tau)\big(\phi(\tau) + c(s_0)\big)\big] \;=\; \mathbb{E}_{\tau\sim\pi_\theta(\cdot\mid S)}\big[A(\tau)\,\phi(\tau)\big].$$

Proof. Because $b(s_0)$ and $c(s_0)$ depend only on the initial state, $\mathbb{E}[b(s_0)\phi(\tau) \mid s_0] = b(s_0)\,\mathbb{E}[\phi(\tau) \mid s_0]$ and $\mathbb{E}[A(\tau)c(s_0)] = \mathbb{E}_{s_0}[c(s_0)\,\mathbb{E}[A(\tau) \mid s_0]] = 0$ by $\mathbb{E}[A(\tau) \mid s_0] = 0$ under group-normalized advantages in GRPO. Unbiasedness of $\nabla_\phi J$ in Definition 3.1 therefore survives any state-dependent centering, and choosing $c(s_0) = -\mathbb{E}[\phi(\tau) \mid s_0]$ minimizes the variance of the empirical estimator $\widehat{\nabla}_\phi J = \frac{1}{N}\sum_{i=1}^N A_i\phi_i$. □

Group normalization of advantages in GRPO plays the role of $b(s_0)$ and explains why Phase A's estimator is well behaved in practice. Partitioning trajectories by the sign of $A(\tau)$ directly yields the positive-minus-negative decomposition of Eq. (3), $\nabla_\phi J = \mathbb{E}_{A>0}[|A(\tau)|\phi(\tau)] - \mathbb{E}_{A<0}[|A(\tau)|\phi(\tau)]$, of which the Phase-A contrastive prompt (Eq. (4)) is the top-K truncated sample estimator.

A.2 Linear Feature Assumption and Proof of Proposition 3.2

Assumption A.3 (Linear feature assumption)**.** *There exists a mapping $\psi : \mathcal{T} \to \mathbb{R}^m$ on trajectories and a linear operator $W \in \mathbb{R}^{d_\phi\times m}$ such that $\phi(\tau) = W\,\psi(\tau)$, and ψ is rich enough that the log-likelihood gradient in skill-text space admits the decomposition $\partial \log \pi_\theta(\tau \mid S)/\partial S = U\,\psi(\tau) + \xi(\tau)$, with $\|\mathbb{E}[A(\tau)\,\xi(\tau)]\| \le \delta_\xi$ ($\mathbb{E}[A(\tau)\,\xi(\tau)] \approx 0$). Write $g := \mathbb{E}[A(\tau)\psi(\tau)] \in \mathbb{R}^m$, $\varepsilon_\perp := \mathbb{E}[A(\tau)\xi(\tau)]$, and define the signal-to-noise ratio $\rho := \|Ug\|/\delta_\xi$.*

In practice, ψ is realized by the eight-dimensional VLM summary, and W aggregates the summary into behavioral indicators. Assumption A.3 merely says that the VLM summary captures enough behavior to linearly express how a skill change affects the log-likelihood.

Remark A.4 (Scope of Assumption A.3 and graceful degradation)**.** *Assumption A.3 is a* working hypothesis *rather than a proved property. It is strictly weaker than asking ϕ to be a sufficient statistic: we only need the residual $\xi(\tau)$ to be approximately advantage-orthogonal, which holds whenever the ψ-summary already captures the dimensions along which advantage varies. When it fails – e.g., when the VLM summarizer drops a decisive dimension so that $g \notin \mathrm{row}(W)$ – decompose $g = g_\parallel + g_\perp$ with $g_\parallel = W^\dagger W g$. Then $\Delta S_A = \eta U g_\parallel$ satisfies $\langle \Delta S_A, \partial J/\partial S\rangle \geq \eta \|U g_\parallel\|(\|U g_\parallel\| - \|U g_\perp\| - \delta_\xi)$: direction consistency holds whenever the captured component dominates the omitted one, and Theorem 3.5 degrades gracefully with ε_{approx} inflated continuously by the projected residual norm rather than collapsing. Consistently, removing the tool-choice appropriateness indicator reduces split-half direction consistency from 0.78 to 0.66 rather than to zero.*

Proof of Proposition 3.2 (with quantitative angle bound). By Lemma A.1 and Assumption A.3,

$$\frac{\partial J}{\partial S} = \mathbb{E}\big[A(\tau)\,(U\psi(\tau) + \xi(\tau))\big] = U\,\mathbb{E}[A(\tau)\psi(\tau)] + \mathbb{E}[A(\tau)\xi(\tau)] = Ug + \varepsilon_\perp,$$

with $\|\varepsilon_\perp\| \leq \delta_\xi$, and $\nabla_\phi J = \mathbb{E}[A(\tau)\phi(\tau)] = W\,\mathbb{E}[A(\tau)\psi(\tau)] = Wg$. When $g \in \mathrm{row}(W)$, the row-space projection satisfies $W^\dagger W g = g$, so the Phase-A update $\Delta S_{\mathrm{A}} = \eta U W^\dagger \nabla_\phi J$ ($\eta > 0$) equals $\eta\, Ug$. By Cauchy–Schwarz and the triangle inequality $\|\partial J/\partial S\| \leq \|Ug\| + \delta_\xi$,

$$\Big\langle \Delta S_{\mathrm{A}},\, \frac{\partial J}{\partial S}\Big\rangle \;=\; \eta\langle Ug,\, Ug + \varepsilon_\perp\rangle \;\geq\; \eta\|Ug\|\big(\|Ug\| - \delta_\xi\big),$$

$$\cos\angle\Big(\Delta S_{\mathrm{A}},\, \frac{\partial J}{\partial S}\Big) \;=\; \frac{\langle \Delta S_{\mathrm{A}},\, \partial J/\partial S\rangle}{\|\Delta S_{\mathrm{A}}\| \cdot \|\partial J/\partial S\|} \;\geq\; \frac{\eta\|Ug\|(\|Ug\| - \delta_\xi)}{\eta\|Ug\|(\|Ug\| + \delta_\xi)} \;=\; \frac{\rho - 1}{\rho + 1},$$

and $\|Ug\| \geq (\sigma_{\min}(U)/\sigma_{\max}(W))\|\nabla_\phi J\|$. Thus whenever $\rho > 1$, ΔS_{A} has strictly positive inner product with $\partial J/\partial S$, establishing condition (C1) used by Theorem 3.5. On our held-out trajectories, the estimated signal-to-noise ratio is $\hat{\rho} = 3.1$, which yields $\cos\angle \geq (3.1 - 1)/(3.1 + 1) = 0.51$ ($\angle \leq 59.3°$). □

A.3 Three-Level Skill Metrics and Proof of Proposition 3.3

Definition A.5 (Three-level skill metrics)**.** *For two skill sets S, S' we define: (L1) the* structural edit distance $\|\Delta\|_{struct}$ *– number of items added or removed; (L2) the* semantic similarity d_{sim} *– the cosine similarity of sentence embeddings (Phase B keeps a candidate when $d_{sim} \leq \delta_{sim} = 0.85$ and drops it when $d_{sim} > 0.85$); (L3) the* behavioral distance $d_{beh}(S, S') = \sup_\theta \mathrm{KL}(\pi_\theta(\cdot \mid S)\|\pi_\theta(\cdot \mid S'))$.

L1 and L2 are computable without rollouts and are enforced by Phase B as hard constraints on the *representation* of the skill edit. They give a necessary, but not sufficient, surrogate for the behavioral step bound required by (C2): a $K_{\max}$-bounded structural edit can still, in pathological cases, flip the first-step tool choice on a whole class of queries and cause a large behavioral KL. The sufficient behavioral surrogate is delivered sampling-wise by Phase C, which estimates L3 up to Lipschitz scaling via the validation-score gap.

Proof of Proposition 3.3. (a) Phase B sets a hard cap $|\Delta S| \leq K_{\max}$ per layer per round and a semantic deduplication constraint $d_{\text{sim}}(c, c') \leq \delta_{\text{sim}}$; both hold deterministically by construction, giving the representation-level (necessary) part of (C2). (b) To see that Phase B alone cannot bound d_{beh}, consider inserting a single candidate skill $c^\dagger$: "route every query involving embedded text rendering through an `image2image` overlay." Here $\|\Delta\|_{\text{struct}} = 1 \leq K_{\max}$ and $c^\dagger$ is not semantically duplicated in S_t, so Phase B accepts it; yet $c^\dagger$ flips the first-step tool selection across an entire query family, inducing $\mathrm{KL}(\pi_\theta(\cdot \mid S_t)\|\pi_\theta(\cdot \mid S_{t+1})) = \Omega(1)$. (c) For Phase C (Eq. (6)), S_{cand} is accepted if $\mathrm{Eval}(\pi_\theta, S_{\text{cand}}) \geq \mathrm{Eval}(\pi_\theta, S_t) - \varepsilon$; otherwise SHRINK retains the top-1 candidate with the largest validation gain and re-evaluates against the same threshold, or falls back to S_t (increment $0 \geq -\varepsilon$). In all three branches the returned S_{t+1} satisfies $J(\theta, S_{t+1}) \geq J(\theta, S_t) - \varepsilon$ up to the validation-set estimation error, which is absorbed into $\varepsilon_{\text{val-sampling}}$ of Theorem 3.4. By Pinsker's inequality, $|J(\theta, S) - J(\theta, S')| \leq 2R_{\max}\,\mathrm{TV} \leq R_{\max}\sqrt{2\,d_{\text{beh}}(S, S')}$; the converse does not hold, so Phase C bounds the downstream task projection $|J(\theta, S) - J(\theta, S')|$ that enters the BCA ascent lemma, giving the sufficient behavioral surrogate for (C2) and (C3). □

A.4 Proof of Theorem 3.4

Proof. First, for any two trajectory distributions μ, ν, boundedness $\|A(\tau)\phi(\tau)\| \le B = A_{\max}\Phi_{\max}$ implies the distribution-shift bound

$$\big\|\mathbb{E}_\mu[A(\tau)\phi(\tau)] - \mathbb{E}_\nu[A(\tau)\phi(\tau)]\big\| \;\le\; \sup_\tau \|A(\tau)\phi(\tau)\| \cdot \|\mu - \nu\|_1 \;\le\; 2B\, d_{\mathrm{TV}}(\mu,\nu) \;=:\; L_\phi\, d_{\mathrm{TV}}(\mu,\nu).$$

Second, because Phase D estimates $\widehat{\nabla}^{\mathrm{D}}_\phi J$ from $n = |\mathcal{F}_{\text{val}}|$ independent validation-failure trajectories drawn directly from μ_{val}, each coordinate $A(\tau)\phi_j(\tau) \in [-B, B]$ satisfies Hoeffding's inequality $\Pr(|\widehat{\nabla}_j - \nabla_j| \ge t) \le 2\exp(-nt^2/(2B^2))$. Taking a union bound over $j = 1, \dots, d_\phi$ and using $\|\cdot\|_2 \le \sqrt{d_\phi}\|\cdot\|_\infty$ yields, with probability at least $1-\delta$,

$$\varepsilon_{\text{val-sampling}} \;:=\; \big\|\widehat{\nabla}^{\mathrm{D}}_\phi J - \nabla_\phi J_{\text{val}}\big\|_2 \;\le\; B\sqrt{\frac{2d_\phi \ln(2d_\phi/\delta)}{|\mathcal{F}_{\text{val}}|}} \;=\; O\big(1/\sqrt{|\mathcal{F}_{\text{val}}|}\big).$$

Now decompose the Phase-A+D merged estimator as the convex combination $\widehat{\nabla}^{\text{S-GRPO}}_\phi J = (1-\alpha)\widehat{\nabla}^{\mathrm{A}}_\phi J + \alpha\widehat{\nabla}^{\mathrm{D}}_\phi J$, where $\alpha \in (0,1)$ is the fraction of new skills contributed by Phase D after semantic deduplication. By the triangle inequality,

$$\begin{aligned}\big\|\widehat{\nabla}^{\text{S-GRPO}}_\phi J - \nabla_\phi J_{\text{val}}\big\| &\le (1-\alpha)\big\|\widehat{\nabla}^{\mathrm{A}}_\phi J - \nabla_\phi J_{\text{val}}\big\| + \alpha\big\|\widehat{\nabla}^{\mathrm{D}}_\phi J - \nabla_\phi J_{\text{val}}\big\| \\ &\le (1-\alpha)\varepsilon^{\text{A-only}}_{\text{approx}} + \alpha\,\varepsilon_{\text{val-sampling}}.\end{aligned}$$

Consequently, the error reduction $G(\alpha) := \varepsilon^{\text{A-only}}_{\text{approx}} - \varepsilon^{\text{S-GRPO}}_{\text{approx}} \ge \alpha(\varepsilon^{\text{A-only}}_{\text{approx}} - \varepsilon_{\text{val-sampling}})$ is strictly positive whenever the distribution-shift bias $L_\phi\, d_{\mathrm{TV}}(\mu_{\text{train}}, \mu_{\text{val}})$ exceeds the validation sampling noise $O(1/\sqrt{|\mathcal{F}_{\text{val}}|})$, and grows monotonically with $d_{\mathrm{TV}}(\mu_{\text{train}}, \mu_{\text{val}})$. □

A.5 Proof of Theorem 3.5

Lemma A.6 (Sufficient ascent under an inexact gradient)**.** *Suppose J is L-smooth along the update direction and the estimator $\widehat{\nabla}$ satisfies $\|\widehat{\nabla} - \nabla\| \le \varepsilon$. With step size $\eta = 1/L$,*

$$J(x + \eta\widehat{\nabla}) - J(x) \;\ge\; \frac{1}{2L}\|\nabla\|^2 - \frac{1}{2L}\varepsilon^2.$$

Proof. By L-smoothness, $J(x+\eta\widehat{\nabla}) - J(x) \ge \eta\langle\nabla, \widehat{\nabla}\rangle - \frac{L}{2}\eta^2\|\widehat{\nabla}\|^2$. Substituting $\eta = 1/L$ and applying the identity $2\langle\nabla,\widehat{\nabla}\rangle - \|\widehat{\nabla}\|^2 = \|\nabla\|^2 - \|\nabla - \widehat{\nabla}\|^2 \ge \|\nabla\|^2 - \varepsilon^2$ gives the result. This provides the step-size constants $\rho_\theta = 1/(2L_\theta)$ and $\rho_S = 1/(2L_S)$. □

Proof of Theorem 3.5. Consider the per-step decomposition

$$J(\theta_{t+1}, S_{t+1}) - J(\theta_t, S_t) = \underbrace{[J(\theta_{t+1}, S_{t+1}) - J(\theta_t, S_{t+1})]}_{\text{upper-level } \theta\text{-step}} + \underbrace{[J(\theta_t, S_{t+1}) - J(\theta_t, S_t)]}_{\text{lower-level } S\text{-step}}.$$

For the upper-level term, by (C4), Lemma A.6, and the standard monotone-improvement lemma for GRPO/TRPO [29, 30], the increment is at least $\rho_\theta\|\nabla_\theta J\|^2 - \varepsilon_\theta$ with $\varepsilon_\theta = O(1/\sqrt{N_{\text{group}}})$.

For the lower-level term, by Proposition 3.2 (C1), the representation-level bound of Phase B (C2), and Lemma A.6 in behavior feature space, the accepted branch where Phase C passes yields at least $\rho_S\|\nabla_\phi J\|^2 - \varepsilon^{\text{S-GRPO}}_{\text{approx}}$. On the branch where Phase C rejects with probability $p_t \le p_{\max}$ and triggers Shrink or Rollback, Proposition 3.3 bounds the increment below by $-\varepsilon$. Taking expectations gives

$$\mathbb{E}[\Delta J_S] \;\ge\; (1-p_{\max})\big(\rho_S\|\nabla_\phi J\|^2 - \varepsilon^{\text{S-GRPO}}_{\text{approx}}\big) \;-\; p_{\max}\varepsilon \;\ge\; \rho_S\|\nabla_\phi J\|^2 - \varepsilon^{\text{S-GRPO}}_{\text{approx}} - p_{\max}\varepsilon$$

(absorbing the $p_{\max}\rho_S\|\nabla_\phi J\|^2$ slack into $\varepsilon^{\text{S-GRPO}}_{\text{approx}}$ up to a constant factor).

Telescoping over $t = 0, \dots, T-1$ gives Eq. (8), and rearranging with $J(\theta_T, S_T) \le J_{\max}$ yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\big[\rho_\theta\|\nabla_\theta J_t\|^2 + \rho_S\|\nabla_\phi J_t\|^2\big] \;\le\; \frac{J_{\max} - J(\theta_0, S_0)}{T} \;+\; \big(\varepsilon^{\text{S-GRPO}}_{\text{approx}} + p_{\max}\varepsilon + \varepsilon_\theta\big).$$

On a fixed validation set, $\varepsilon_{\text{approx}}^{\text{S-GRPO}} \geq \alpha\, \varepsilon_{\text{val-sampling}} = \Omega(1/\sqrt{|\mathcal{F}_{\text{val}}|})$ is a non-vanishing error floor as $T \to \infty$; only when the validation set grows with T so that all three error terms vanish does (θ_t, S_t) approach a first-order stationary point. In our runs, the measured rejection rate $p_{\max} = 0.086$ and gate tolerance $\varepsilon = 0.01$ give a safety-net floor $p_{\max}\varepsilon = 8.6 \times 10^{-4}$ (with $\alpha = 0.28$, $|\mathcal{F}_{\text{val}}| = 64$, and $N_{\text{group}} = 8$), which is more than an order of magnitude smaller than the empirical gains in Table 1. □

A.6 Degenerate Behavior under Uniform Advantages

When group-normalized advantages are uniform (e.g., all trajectories in a group succeed with identical reward, or all fail), we have $A(\tau) \equiv 0$ and the contrast in Phase A collapses to zero. In this case Skill-GRPO defers the S-step to the next interval, avoiding a zero-signal update that would inject noise without improving direction; since $\Delta J_S = 0 \geq -\varepsilon$, the bound of Theorem 3.5 remains valid. In practice, we detect this condition by the within-group advantage variance: if it falls below a threshold, the current S-step is skipped.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the two methods (Dual-Skills Evolution and Skill-GRPO), the BCA-style error bound under a linear-feature assumption, and the scope of the empirical claims (VCR-Bench, GEdit-Bench, MultiBanana); Theorem 3.5 explicitly records the error floor on a fixed validation set.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 states that the behavior feature is extracted by a frozen VLM summarizer, so summarizer quality upper-bounds the direction consistency of the textual gradient. Remark A.4 adds that dropping a reward-relevant behavior dimension makes Phase A a biased estimator.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theorems, propositions, Lemmas A.1–A.6, and the load-bearing Assumption A.3 (with Remark A.4) are numbered and cross-referenced; full proofs of Propositions 3.2, 3.3 and Theorems 3.4, 3.5 are given in Appendix A, with an explicit angle lower bound and fixed-validation-set error floor.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4.1 gives the backbone, tool, hyperparameters, and training pipeline; the VCR-Bench protocol is described in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code modifications, training commands, and evaluation scripts are not yet publicly released.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 lists the default Skill-GRPO hyperparameters and the two VCR-Bench judges together with their average.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The main tables do not report error bars or multi-seed statistics.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper does not report the GPU type, GPU count, or S-step runtime.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses existing VLMs and diffusion tools and involves no human-subject studies.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

Justification: Section 6 only notes that the method extends to other agents that couple continuous parameters with discrete text. It does not discuss societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: No new pretrained model or dataset is released.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external assets (Qwen3-VL, Qwen-Image-Edit/Fast, Gemini-2.5 Pro, GEdit-Bench, MultiBanana, SkillRL, OpenClaw-RL, RLAnything, VisionCreator/VisionCreator-R1) are cited where used and referenced to their original releases.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The new prompts, schemas, and training code are not provided with the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects; all evaluations are fully automated by VLM judges.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human-subjects study is conducted.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.