LOOK BACK TO MOVE FORWARD: LEARNING TO REVISIT FOR AGENTIC COMPOSITIONAL IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image models have made remarkable progress in visual fidelity and instruction following, yet complex compositional generation remains challenging. Existing agentic approaches address this limitation through iterative generation and editing, adapting their actions based on interaction history. However, they typically fix the continuation state (*i.e.*, the image from which refinement proceeds) to the latest image. Since iterative editing is not necessarily monotonic, the agent may continue refining from the latest image even when it has already degraded. In this work, we introduce **GenRevisit**, an agentic framework that extends action selection to *joint continuation-state and action selection*. The agent can not only use interaction history to adjust editing instructions but also select an image candidate to edit, jointly deciding *where to continue from and how to refine*. We train GenRevisit with SFT followed by agentic reinforcement learning. To address the coarse credit assignment of final outcome reward in GRPO, which assigns the same trajectory-level advantage to all actions, we further propose Normalized Action Credit Assignment (NACA) to redistribute positive advantage according to each action's contribution to reducing the remaining objective. On GenEval2, GenRevisit achieves GM scores of 84.3 and 74.4 with Qwen-Image-2512 and Z-Image-Turbo, surpassing the strongest baselines using the same generators by 12.7 and 10.9 points respectively. GenRevisit achieves state-of-the-art performance on GenEval, GenEval++, and GenEval2 among baselines. Code is available at `https://anonymous.4open.science/r/GenRevisit-7274/`.

1 INTRODUCTION

Recent text-to-image (T2I) models have made substantial progress in visual fidelity and instruction following [1–3], while satisfying object counts, attributes, and spatial relations in a single generation attempt remains challenging [4, 5]. To address this challenge, Iterative generation uses feedback on intermediate results to guide subsequent generation and editing [6–8]. As iterative generation produces multiple intermediate outcomes, the appropriate corrective action may vary from round to round. Fixed refinement procedures may therefore struggle to adapt to the evolving generation process, motivating an agentic sequential decision-making formulation in which the agent learns to select the next operation based on the accumulated interaction history [9].

However, these policies mainly focus on learning what action to take next. An editing operation requires both a source image (*i.e.*, the image from which refinement proceeds) and an editing instruction, but the source is often predetermined by the workflow. Although the interaction history records what action has attempted on each state and the resulting feedback, it works only for revising the instruction based on the fixed state if we just learn the action selection policy. This design has room for improvement because if we can't jointly decide continuation state and action to do, we may miss more promising state to continue for better results. For example, as illustrated in Figure 1, an edit to I_0 produces I_1 but violates previously satisfied constraints. A latest-only policy then gains additional feedback, yet its next edit must still start from the degraded image I_1. The agent may thus spend its limited budget restoring constraints that remain satisfied in I_0. This highlights the need for a policy that learns to jointly select a continuation state and specify an editing instruction tailored to it, rather than only adapting the instruction to a fixed source.

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.

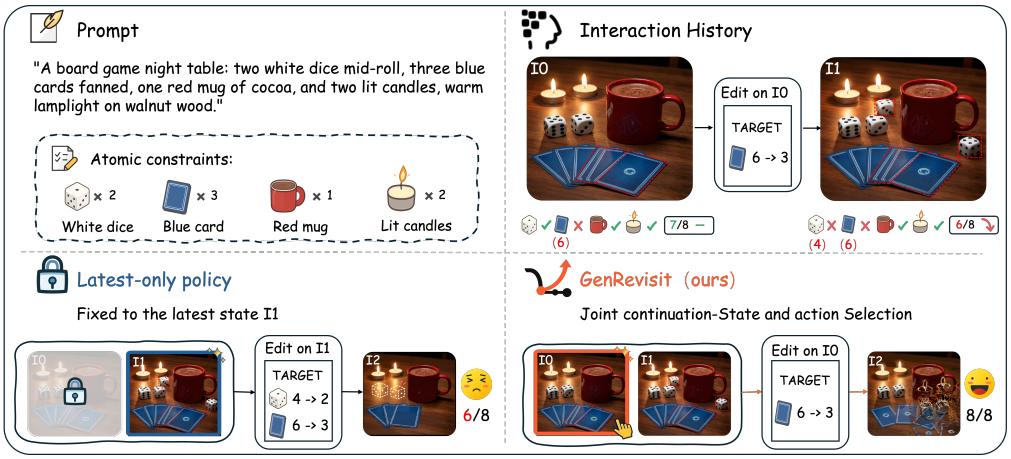


Figure 1: An illustrative comparison between latest-state refinement and our GenRevisit.

To this end, we introduce GenRevisit, a framework that expands decision space from action selection to joint continuation-state and action selection. We treat image candidates accumulated in the interaction history as candidate continuation states, each corresponding to a possible starting point for further refinement. At each decision step, GenRevisit uses a multimodal planner to choose the next action from generation, editing, querying, and submission, based on decision context. When it edits, it jointly decides a source image and an editing instruction to correct remaining errors while preserving satisfied constraints. This allows the planner to revisit a stronger image candidate and open new refinement branches, rather than being restricted to the latest image.

We train our proposed GenRevisit in two stages. First, we collect multi-round trajectories with GPT-5.5 and initialize the planner by supervised fine-tuning (SFT). We supervise actions that improve the image, including recovery from degraded states. Failed actions are excluded from the SFT loss but kept in the history with their feedback, because they still inform later decisions. We then use reinforcement learning (RL) with GRPO [10] to further improve the planner through its own interactions. The terminal reward measures how well the final submitted image satisfies the task constraints. However, the standard GRPO assigns the same advantage to every action in a trajectory, so a positive trajectory advantage can reinforce actions that worsen the image. To address this issue, we propose Normalized Action Credit Assignment (NACA). In positive-advantage trajectories, NACA redistributes credit among generation and editing actions based on the fraction of the remaining utility gap that each action closes.

Our experiments demonstrate the effectiveness of GenRevisit from three perspectives: **(i) Compositional generation:** GenRevisit achieves 84.3 GM on GenEval2 [5], exceeding the strongest agentic baseline by 12.7 points. Results on GenEval [4] and GenEval++ [11] further support cross-benchmark generalization. **(ii) Cross-generator transfer:** We use GenRevisit with Z-Image-Turbo [12] without additional training. It also outperforms the strongest agentic baseline with the same generator by 10.9 points on GenEval2. **(iii) Benefit of joint selection:** Joint selection outperforms latest-only and best-only rules by 5.0 and 2.8 GM points, in matched-budget teacher continuations. The policy trained on joint demonstrations achieves 81.5 GM on GenEval2, exceeding its best-only-trained counterpart by 4.0 points.

Our contributions are summarized as follows:

- We propose GenRevisit, a history-conditioned framework for joint continuation-state and action selection, making the choice of where to continue an explicit policy decision rather than a predetermined starting condition.
- We combine action-level SFT with RL and introduce NACA, which redistributes image-action credit in positive-advantage trajectories according to normalized local progress.
- GenRevisit achieves state-of-the-art performance on GenEval, GenEval++ and GenEval2 among baselines, surpassing the best GenEval2 results for Qwen-Image-2512 and Z-Image-Turbo by 12.7 and 10.9 points respectively, showing strong compositional generation.

2 Related Work

Compositional evaluation. As text-to-image models are increasingly expected to satisfy constraints involving multiple objects, attributes, and relations simultaneously, compositional faithfulness has become a core test of generation capability. Earlier benchmarks such as GenEval, T2I-CompBench, and ConceptMix organize evaluation around predefined compositional dimensions [4, 13, 14], whereas TIFA, Davidsonian Scene Graph, and GenEval 2 move toward prompt-specific atomic verification by decomposing each request into individually verifiable rubrics [15, 16, 5]. Inspired by this paradigm, we use constraint-level atomic judgments as fine-grained environment feedback for multi-round agentic generation, allowing decisions to be grounded in how individual constraints evolve across attempts.

Iterative image generation. Iterative generation has progressed from prompt rewriting, repeated editing, and visual reflection [6–8] to closed-loop systems that evaluate intermediate results and invoke generation or editing tools accordingly [17–19]. More recent approaches make this feedback loop increasingly explicit: Qwen-Image-Agent incorporates verification feedback into subsequent refinement, while R3 organizes feedback-driven retry and repair [20, 21]. Generation Navigator further formulates iterative generation as state-dependent action selection, learning when to stop, refine, or regenerate [9]. We build on this progression by extending the decision from what to do next to which source state to act from: historical candidates are retained as selectable branching points, allowing the policy to jointly choose a source state and the subsequent operation.

3 Preliminaries

Agentic image generation can be viewed as a sequential decision process in which a planning policy interacts with external tools. Given a prompt specifying multiple compositional requirements, the objective is to produce an image that satisfies these requirements within a total budget of image generation and editing calls. At each decision step, the planner selects an action according to its planning policy, conditioned on the current decision context. The environment executes the action and updates the decision context with the resulting observations and feedback. We formalize this interaction in terms of states, actions, and trajectories.

States and actions. Let x denote the input prompt and B the total budget of image generation and editing calls. At decision step t, the interaction state s_t denotes the decision context available to the planning policy, including x, the available history of actions, images, and feedback, and the remaining budget. The policy π_θ, parameterized by θ, selects an executable action as

$$a_t \sim \pi_\theta(\cdot \mid s_t). \tag{1}$$

Each action specifies an operation and its associated parameters. Here, s_t represents the overall decision context rather than an individual image candidate.

Interaction trajectories. The environment executes a_t and incorporates the resulting observations and feedback into the next decision state s_{t+1}. The sequence of interactions forms a trajectory

$$\tau = (s_0, a_0, s_1, a_1, \ldots, s_{T-1}, a_{T-1}, s_T), \tag{2}$$

where T is the number of decision steps. The interaction terminates when a final image is selected as the output or another prescribed termination condition is met. The total number of generation and editing calls is bounded by B and need not equal T.

While s_t denotes the complete interaction state available for decision making, image candidates accumulated in the interaction history serve as candidate continuation states, each representing a possible starting point for further refinement. We refer to the candidate selected for this purpose as the *continuation state*. Whether the continuation state is fixed to the latest candidate or explicitly selected from the interaction history is determined by the action interface. Section 4.1 describes a history-conditioned interaction framework that enables joint continuation-state and action selection.

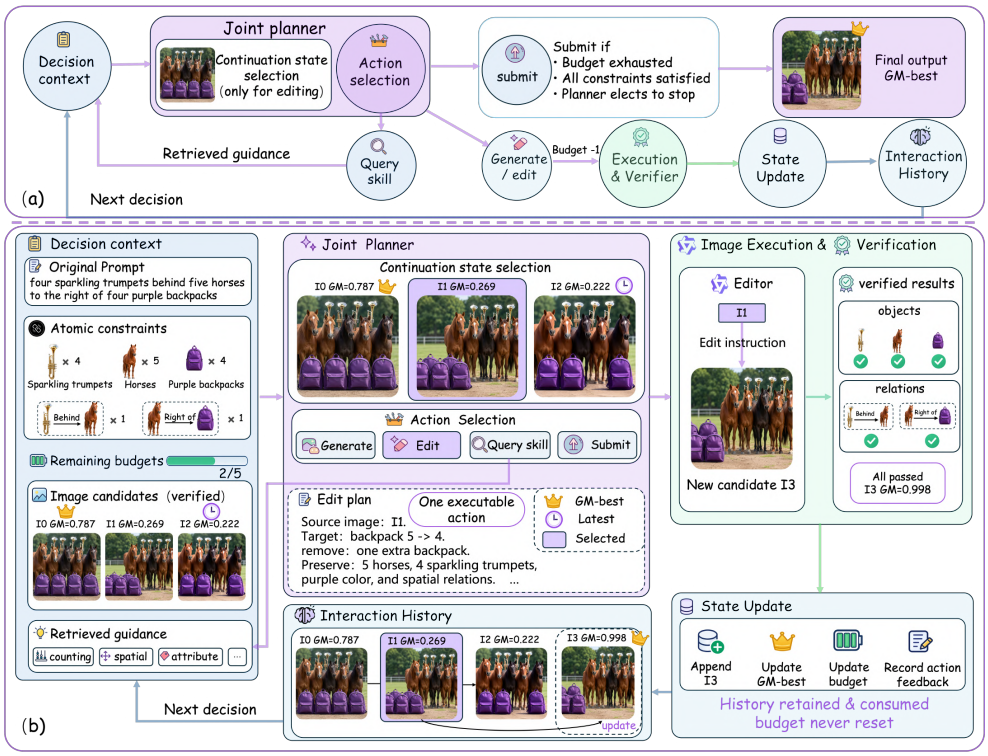


Figure 2: Overview of GenRevisit. Panel (a) presents the overall workflow. At each decision step, the multimodal planner selects the next action based on the decision context and, when editing, jointly selects a source image as the continuation state and a corresponding edit specification. The resulting image is verified and recorded in the interaction history to inform subsequent decisions, while submission returns the historical GM-best image. Panel (b) illustrates an editing example. The planner selects I_1, which is neither the latest candidate I_2 nor the GM-best candidate I_0, and corrects the backpack count while preserving the other constraints. The resulting candidate I_3 passes all constraint-level checks and becomes the new GM-best image.

4 METHOD

An overview of our GenRevisit is shown in Figure 2. GenRevisit combines a history-conditioned interaction framework for joint continuation-state and action selection with a two-stage planner training procedure comprising supervised fine-tuning and reinforcement learning. We detail the history-conditioned interaction framework in Section 4.1, supervised fine-tuning in Section 4.2, and reinforcement learning with normalized action credit assignment in Section 4.3.

4.1 HISTORY-CONDITIONED INTERACTION

Decision context. The decision context s_t defined in Section 3 is instantiated as shown in Figure 2. The compositional requirements of prompt x are represented by fixed atomic constraints $\mathcal{C} = \{c_j\}_{j=1}^{K}$, each paired with a VQA-style verification question. Historical image candidates are provided as visual inputs alongside the actions that produced them and their verification feedback, allowing the planner to assess potential editing sources. Guidance returned by previous skill queries is also included in the decision context.

Joint continuation-state and action selection. At each decision step, the agent produces one structured action

$$a_t = (o_t, \eta_t) \sim \pi_\theta(\cdot \mid s_t), \qquad \eta_t = (k_t, u_t) \text{ when } o_t = \textsc{Edit},$$

where $o_t \in \mathcal{A} = \{\textsc{Generate}, \textsc{Edit}, \textsc{Retrieve}, \textsc{Submit}\}$ denotes the operation type and η_t contains its operation-specific arguments. Generation and editing specify the constraints to target and to

preserve together with an executable instruction, and guidance retrieval requests constraint-specific advice. Submission returns the historical GM-best image and terminates the interaction.

For editing, k_t indexes an available historical image candidate I_{k_t} from the interaction history, which serves as the continuation state, and u_t specifies the editing instruction and the constraints to target and preserve. This implements joint continuation-state and action selection: conditioned on s_t, the planner jointly selects the source image I_{k_t} and its edit instruction u_t within a single structured action. The planner thus decides both where to continue refinement and how to address the selected image's remaining errors while preserving satisfied constraints.

Execution and verification and update. The environment executes the selected action a_t, and for editing, it applies the jointly selected edit specification u_t to the source image I_{k_t}. Generation and editing each consume one image call. Every resulting image, including the initial generation, is evaluated against the fixed constraints $\mathcal{C}$, yielding a GM score and constraint-level feedback. After verification, the environment appends the new candidate, its executed action, and feedback to the interaction history, retaining the source image and editing instruction for edits. It updates the historical GM-best candidate and remaining budget to form the next decision context s_{t+1}. Regressive attempts remain in the history, and editing an earlier candidate creates a new branch without discarding subsequent history or restoring consumed budget.

4.2 Supervised Fine-Tuning For Cold Start

Teacher trajectorie construction. We use GPT-5.5 as a teacher agent to collect 1,000 multi-round trajectories in the history-conditioned interaction environment described in Section 4.1, covering 1,000 prompts from the GenEval2 training set used by Flow-DPPO [22]. Training targets are selected from the teacher trajectories based on action outcomes. For edits, we retain actions that improve the image being edited. Degraded actions and their feedback remain in the context for later decisions while no imitation loss. The prompt-level split yields 3,473 training targets across 800 training prompts, with 100 prompts each reserved for validation and testing. Appendix B describes the full data construction process and reports the resulting dataset statistics.

Masked supervised fine-tuning. For each selected teacher action a_t, we construct a training sequence z_j by serializing the corresponding interaction state s_t followed by the action response, with the associated images provided as conditioning inputs. The state s_t contains only information available before a_t; the action's outcome is used solely for offline target selection. For editing targets, the response includes both the image choice k_t and the edit specification u_t. We define a binary mask $m_{j,\ell}$ that equals one on the action-response tokens and zero on all preceding context tokens, and minimize

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{Z}\sum_{j}\sum_{\ell} m_{j,\ell}\log \pi_\theta\big(z_{j,\ell} \mid z_{j,<\ell}\big), \qquad Z = \sum_{j}\sum_{\ell} m_{j,\ell}. \tag{1}$$

Here, ℓ indexes token positions and Z is the total number of supervised tokens. Only the selected action response receives imitation loss, while prior actions and feedback, including failed attempts, remain available as conditioning context. The resulting SFT policy serves as the initialization for reinforcement learning in Section 4.3.

4.3 Reinforcement Learning with Normalized Action Credit

Terminal reward and trajectory advantage. For each prompt, we sample G trajectories under the policy π_{old}. We define image utility as $U(I) = P(I) + \alpha g(I)$, where $P(I)$ counts passed constraints and $g(I) \in [0,1]$ is the Soft-TIFA geometric mean of the verifier's correct-answer probabilities across constraints [5]. With $\alpha = 0.25$, the passed-constraint count takes priority, while GM breaks ties. This priority is intentional: $P(I)$ provides an explicit signal for constraint completion, while $g(I)$ retains sensitivity to continuous improvements among images with the same pass count. For K constraints, the upper bound is $D = K + \alpha$. For the image I_i^{sub} submitted in trajectory i, the terminal reward and group-relative advantage are

$$R_i = U(I_i^{\mathrm{sub}}), \qquad A_i = \frac{R_i - \bar{R}}{\sigma_R}, \tag{2}$$

where $\bar{R}$ and σ_R are the group mean and population standard deviation. Failure handling and the exclusion of zero-variance groups are described in Appendix C.2.

Normalized action credit assignment. Standard GRPO assigns the same trajectory advantage to every action, potentially reinforcing regressive actions in positive-advantage trajectories. NACA redistributes positive advantage among generation and editing actions according to normalized local progress, without changing the terminal reward. For an image action $a_{i,t}$ producing $I_{i,t}$, the reference utility $U_{i,t}^{\mathrm{ref}}$ is the utility of the image being edited, zero for initial generation, and the highest available image utility before subsequent generation. We define

$$q_{i,t} = \frac{[U(I_{i,t}) - U_{i,t}^{\mathrm{ref}}]_+}{D - U_{i,t}^{\mathrm{ref}}}, \tag{3}$$

for $U_{i,t}^{\mathrm{ref}} < D$, with $q_{i,t} = 0$ when $U_{i,t}^{\mathrm{ref}} = D$ and $[z]_+ = \max(z, 0)$. This ratio measures the fraction of the remaining utility gap closed by the action, rather than its absolute improvement. An edit can therefore make positive progress without exceeding the historical best GM score.

Let $\mathcal{I}_i$ index valid, evaluated image actions and $L_{i,t}$ their trainable response-token counts. For positive-advantage trajectories, effective positive-progress actions receive

$$w_{i,t} = \min(c_i, \lambda_i q_{i,t}), \qquad \sum_{t \in \mathcal{I}_i} L_{i,t} w_{i,t} = \sum_{t \in \mathcal{I}_i} L_{i,t}. \tag{4}$$

Here, c_i caps the weights and λ_i scales them to preserve the token-weighted total. Other image actions receive zero weight; if no effective progress is available, all weights remain one. Appendix C.4 specifies the effective-progress threshold, cap, and numerical solver. The action advantages are

$$\widetilde{A}_{i,t} = \begin{cases} A_i w_{i,t}, & A_i > 0 \text{ and } t \in \mathcal{I}_i, \\ A_i, & \text{otherwise.} \end{cases} \tag{5}$$

Nonpositive advantages remain unchanged, since weighting negative advantages by progress could penalize improving actions more strongly. Retrieval and submission also retain A_i.

Policy optimization. Each sampled action-response token inherits its action's advantage, including the image choice and edit specification for editing. At each RL round, the preceding checkpoint serves as both π_{old} and the frozen KL reference π_{ref}. Let $\mathcal{B}$ contain the trajectories retained for an optimizer step and $\mathcal{T}_i$ their mask-selected response tokens. For token $y_{i,\ell}$ with multimodal prefix $H_{i,\ell}$, define $\rho_{i,\ell} = \pi_\theta(y_{i,\ell} \mid H_{i,\ell}) / \pi_{\mathrm{old}}(y_{i,\ell} \mid H_{i,\ell})$. We maximize

$$\mathcal{J}_{\mathrm{NACA}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{\ell \in \mathcal{T}_i} \Big[\min\big\{ \rho_{i,\ell} \widetilde{A}_{i,\ell},\ \mathrm{clip}(\rho_{i,\ell}, 1 - \epsilon_-, 1 + \epsilon_+) \widetilde{A}_{i,\ell} \big\} - \beta \mathcal{K}_{i,\ell} \Big], \tag{6}$$

where ϵ_- and ϵ_+ are the clipping offsets, and β weights the KL penalty $\mathcal{K}_{i,\ell}$ against π_{ref}. The objective sums over response tokens and averages over trajectories, without length normalization. NACA changes advantages, not response masks: retained responses with zero action advantage still contribute their KL terms. Context and environment observations are inputs only. Appendix C gives the numerical solver, KL estimator, and training details.

5 Experiments

5.1 Setup

Implementation Details. We adopt Qwen3-VL-8B-Instruct [23] as the planner backbone, with Qwen-Image-2512 [24] and Qwen-Image-Edit-2511 [25] as frozen generation and editing tools. The skill library is predefined and remains unchanged during evaluation. The SFT training split contains 800 prompt groups and 3,473 selected action targets. SFT uses a learning rate of 10^{-5}, a global batch size of 32, and two epochs. The RL group size is four, with 1,280 sampled trajectories in total. Detailed dataset statistics, training settings, and tool configurations are provided in Appendices A.3, B, and C.

Table 1: Main results on GenEval2. GM averages prompt-level geometric means of atom scores; Skill Avg. averages the five skill-wise AM scores. Bold and underlined scores mark the best and second-best results among (i) all direct-generation models and Qwen-Image-2512 workflows, or (ii) Z-Image-Turbo workflows alone. † marks the best external agentic baseline within each generator block; ↑/↓ shows absolute score-point changes relative to it.

Generator	Agent	Method	Overall↑		Skill-wise AM↑				
			GM	Skill Avg.	Object	Attribute	Position	Count	Verb
CLOSED-SOURCE IMAGE GENERATION MODELS									
Nano Banana [5]		Direct generation	44.6	<u>83.5</u>	99.0	91.4	70.2	70.1	**86.7**
Nano Banana 2 [18]		Direct generation	63.4	81.7	97.0	<u>95.1</u>	83.7	76.4	56.1
OPEN-SOURCE IMAGE GENERATION MODELS									
Bagel [26]		Direct generation	23.1	66.6	92.9	75.9	50.6	55.6	57.8
Qwen-Image [3]		Direct generation	33.8	77.3	<u>99.1</u>	85.6	60.2	70.3	<u>71.1</u>
Qwen-Image-2512 [3]		Direct generation	29.0	62.0	97.9	65.2	58.1	66.9	22.1
Z-Image-Turbo [18]		Direct generation	31.0	63.4	97.8	75.4	62.2	62.8	18.6
AGENTIC GENERATION WORKFLOWS									
Qwen-Image-2512	GPT-4	Rewrite [18]	39.8	69.3	97.3	81.1	73.8	64.0	30.3
		Promptist [27]	27.4	62.5	97.3	64.4	59.4	63.0	28.6
		Search [28]	42.1	67.5	98.3†	71.1	66.1	73.9	28.2
	Kimi-K2.5	Maestro [6]	48.0	72.4	97.3	84.9	80.9	68.0	31.0
		Mind-Brush [29]	32.1	–	–	–	–	–	–
		SCOPE [30]	46.3	–	–	–	–	–	–
		CRAFT [31]	66.3	82.4†	97.9	91.3	89.9	79.0	53.7†
		GEMS [18]	70.4	81.3	96.9	91.4	91.4	<u>81.3</u>†	45.5
		GenRouter [32]	<u>71.6</u>†	81.8	98.3†	92.3†	<u>91.6</u>†	80.7	46.3
	Qwen3-VL-8B	Base policy	49.7	74.6	97.9	88.9	64.4	72.4	49.6
		GenRevisit	**84.3** ↑12.7	**86.7** ↑4.3	**99.4** ↑1.1	**97.2** ↑4.9	**92.5** ↑0.9	**90.3** ↑9.0	53.9 ↑0.2
Z-Image-Turbo	Kimi-K2.5	GEMS [18]	<u>63.5</u>†	79.2	97.9	92.4	85.4	79.0	41.1
		GenRouter [32]	57.6	76.4	97.9	89.5	76.8	80.3	37.4
	Sonnet 4.5	COMFYCLAW [33]	62.0	70.3	96.2	85.4	72.2	66.8	32.2
	Qwen3-VL-8B	**GenRevisit**	**74.4** ↑10.9	79.4	98.1	96.6	85.9	74.0	42.6

Benchmark and Evaluation. We use GenEval2 as our primary benchmark for complex compositional generation and report its GM score. We additionally report overall scores on GenEval [4] and GenEval++ [11] to assess compositional generation beyond GenEval2. We also evaluate the same policy with Z-Image-Turbo on GenEval2 to test cross-generator transfer without additional training. We use greedy planner decoding with at most five generation or editing calls per prompt, including the initial generation, and evaluate the final submitted image. Baseline scores are taken from the cited sources under their reported configurations. Detailed evaluation protocols are provided in Appendix A.4.

5.2 EVALUATIONS

Main results on GenEval2. Table 1 shows that our method achieves 84.3 GM with Qwen-Image-2512, exceeding the strongest agentic baseline using the same generator by 12.7 points. It also attains the highest Skill Avg. of 86.7 and leads in four of the five skill categories. Counting shows the largest skill-wise gain, reaching 90.3 and exceeding the best competing score in this category by 9.0 points. Verb remains a comparatively weak category at 53.9. These results are obtained with an average of 2.62 image calls per prompt across all 800 prompts. With Z-Image-Turbo, the same policy achieves 74.4 GM without additional training, exceeding the strongest agentic baseline using this generator by 10.9 points. This result shows that the learned policy remains effective when paired with a different generator.

Cross-Benchmark Evaluation. As shown in Table 2, our method achieves the highest overall scores on GenEval [4] and GenEval++ [11], at 0.9844 and 0.8929, respectively. The strong counting performance on GenEval2 is also evident on GenEval (0.9875) and GenEval++ (1.000). These re-

Table 2: Cross-benchmark evaluation on GenEval and GenEval++.

(a) GenEval

Method	Agent	Generator	Overall↑
DIRECT GENERATION			
FLUX.1-dev [34, 19]	–	FLUX.1-dev	67.0
Qwen-Image [3, 19]	–	Qwen-Image	87.0
Qwen-Image-2512† [18]	–	Qwen-Image-2512	74.0
Z-Image-Turbo† [18]	–	Z-Image-Turbo	82.0
AGENTIC GENERATION AND REFINEMENT			
GEMS [18]	Kimi K2.5	Qwen-Image-2512	79.0
GEMS [18]	Kimi K2.5	Z-Image-Turbo	86.0
ReflectionFlow‡ [34]	Qwen2.5-VL-7B	FLUX.1-dev	91.0
RAISE [35]	Mistral-Small-3.2-24B	FLUX.1-dev + Kontext	94.0
CoIG [36]	Gemini 2.5 Flash	Nano Banana	94.8
OctoT2I [37]	Qwen2-0.5B	5-model pool	96.0
GenRevisit	**Qwen3-VL-8B**	**Qwen-Image-2512**	**98.4**

(b) GenEval++

Method	Agent	Generator	Overall↑
DIRECT GENERATION			
FLUX.1-dev [34, 19]	–	FLUX.1-dev	32.5
Qwen-Image [3, 19]	–	Qwen-Image	66.8
GPT-Image-1.5 [38]	–	GPT-Image-1.5	75.0
Gemini-3.1 Flash-Image [38]	–	Gemini-3.1	77.5
AGENTIC GENERATION AND REFINEMENT			
ReflectionFlow‡ [34]	Qwen2.5-VL-7B	FLUX.1-dev	36.1
FiRe-GRPO (3rd) [39]	Janus-Pro-7B	Janus-Pro-7B	49.1
GenAgent [19]	Qwen2.5-VL-7B	FLUX.1-dev	56.1
GenAgent [19]	Qwen2.5-VL-7B	Qwen-Image	72.5
Mind-Brush [29]	GPT-5.1	Qwen-Image + Edit-2512	78.2
GenClaw [38]	Claude Opus 4.6	Gemini-3.1 Flash-Image	87.9
GenRevisit	**Qwen3-VL-8B**	**Qwen-Image-2512**	**89.3**

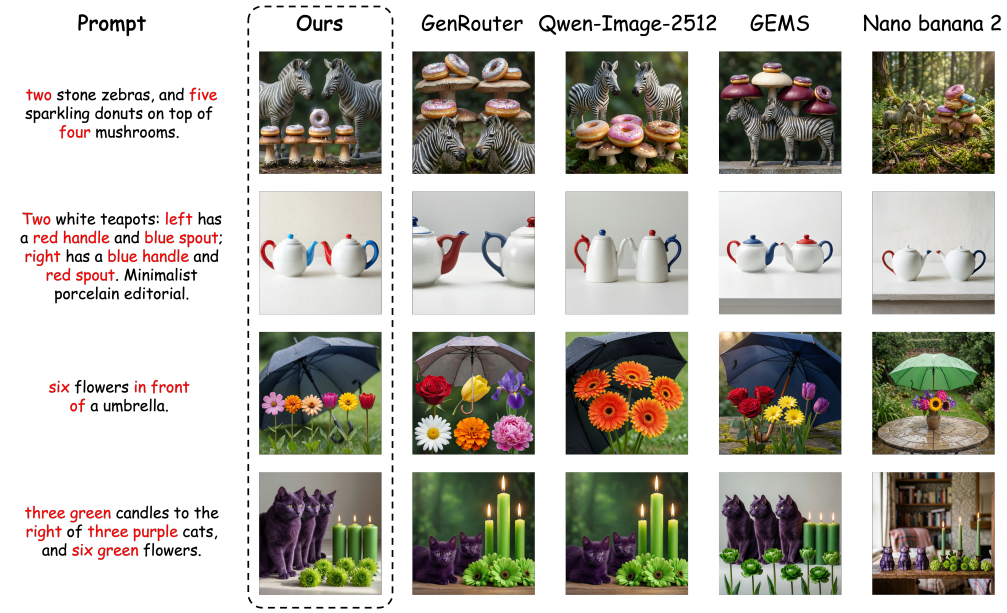


Figure 3: Qualitative comparison of GenRevisit and baseline models.

sults suggest that the learned policy can guide compositional generation beyond GenEval2. Detailed per-category results and evaluation protocols are provided in Appendices E.4 and A.4.

Qualitative Comparison. As show in Figure 3, GenRevisit presents qualitative comparisons on prompts involving counts, colors, attribute and relations. Our method more consistently satisfies these constraints, including the object counts, position-specific handle and spout colors of the teapots, and the counts and spatial arrangements of flowers, candles, and cats. These examples complement the quantitative results, illustrating improved compositional alignment while maintaining visual coherence. Additional outputs and an executed revisit trajectory are provided in Appendix F.5.

5.3 ABLATION STUDY

Table 3 examines the effects of our workflow and policy learning on GenEval2. Direct generation with Qwen-Image-2512 achieves 29.0 GM, while an untuned Qwen3-VL-8B planner reaches 49.7 using our workflow, with access to constraint-level feedback, interaction history, and historical images as editing sources. SFT raises GM to 81.5, with the largest skill-wise gains over the untuned planner in Position (+27.3) and Count (+16.4). This improvement highlights the value of learning

Table 3: Ablation study on GenEval2.

Variant	Overall			Skill-wise AM				
	GM	ΔGM	Skill Avg.	Object	Attribute	Position	Count	Verb
Qwen-Image-2512	29.0	–	62.0	97.9	65.2	58.1	66.9	22.1
Untuned Qwen3-VL-8B-Instruct†	49.7	+20.7	74.6	97.9	88.9	64.4	72.4	49.6
SFT only	81.5	+31.8	85.3	98.9	96.1	91.7	88.8	51.1
SFT + GRPO	80.3	−1.2	85.1	98.7	96.0	91.1	88.4	51.4
SFT + NACA	**84.3**	+4.0	**86.7**	**99.4**	**97.2**	**92.5**	**90.3**	**53.9**

† Untuned Qwen3-VL-8B-Instruct uses our agent workflow. ΔGM is relative to the preceding row (NACA vs. standard GRPO).

the refinement policy rather than relying on an untuned planner to execute the workflow. Standard GRPO yields 80.3 GM, below the SFT policy's 81.5. In contrast, GRPO with NACA reaches 84.3 GM, outperforming standard GRPO by 4.0 points and SFT by 2.8 points. The NACA variant also achieves the highest scores across all five skills among the evaluated variants. These results support redistributing positive trajectory credit according to the progress made by individual image actions.

5.4 Analysis

Although we intuitively might assume that continuing to edit from the latest state would yield better results, by analyzing the agent's decision trajectories, we find it implicitly selects the continuation state in three patterns: **(i)** following an edit that breaks satisfied constraints without resolving others, the agent often returns to an earlier, higher-scoring candidate; **(ii)** conversely the agent may continue from the latest image despite a lower overall score if it satisfies previously unmet constraints; **(iii)** an earlier candidate that preserves constraints violated by later edits may be selected even if it is neither latest nor highest-scoring, providing a cleaner starting point for correcting the remaining errors. We further quantify the benefits of dynamic source selection through two experiments:

Controlled teacher continuations. To quantify the benefit of joint selection, we select 200 prefixes from a pool of 800 teacher trajectories where the task remains unsolved after three image calls. Each prefix provides a shared starting history for Joint, Latest-only, and Best-only, with two image calls remaining. All three use the same teacher and interaction protocol, differing only in edit-source access. Latest-only uses the latest image, Best-only uses the current highest-GM image, and Joint selects the source together with its editing instruction. Joint achieves a mean final GM of 55.2, compared with 52.4 for Best-only and 50.2 for Latest-only. Its improvement over Best-only is 2.8 points (paired 95% CI: [0.46, 5.15]). Joint also increases the proportion of prefixes producing a new candidate above the shared prefix best from 57.0% with Best-only to 65.0%. These results support the value of joint selection beyond fixing the editing source to the current highest-scoring image.

Learning from joint demonstrations. A second experiment examines whether joint demonstrations lead to better performance after SFT. An additional 800 Best-only teacher trajectories are collected using the same prompts and initial images as the Joint trajectories. The two demonstration sets are used to train separate student policies via SFT. Both policies are then evaluated on the official GenEval2 evaluation set of 800 prompts under the same evaluation protocol used for our main results. The student trained on Joint demonstrations achieves 81.5 GM, compared with 77.5 for its Best-only-trained counterpart, an improvement of 4.0 points (paired 95% CI: [2.81, 5.18]). The advantage is thus observed not only in controlled teacher continuations but also in the benchmark performance of the trained policies.

6 Conclusion

We introduce GenRevisit, which learns to jointly select a continuation state and an action. SFT retains failed actions and outcomes as recovery context rather than supervision. NACA assigns action-level credit using normalized local progress in non-monotonic trajectories. Experiments demonstrate improved compositional generation across benchmarks and cross-generator transfer without additional planner training. Our findings suggest that an image's value as a starting point depends on both its current constraint satisfaction and what a suitable edit can achieve, highlighting the importance of learning source selection and editing together.

AI USE STATEMENT

During manuscript preparation and research discussions, we used generative AI tools, including ChatGPT and OpenAI Codex, for language editing and drafting, manuscript organization, LaTeX formatting, and feedback on experimental comparisons and result interpretation. The authors reviewed and revised the AI-assisted content and retain full responsibility for the methods, claims, and reported results.

Generative AI also formed part of the research methodology. GPT-5.5 served as the teacher for collecting multi-round interaction trajectories. The resulting actions, tool-generated images, and verification feedback were used to construct selective SFT supervision, as described in Section 4.2 and the appendix. Qwen3-VL-8B-Instruct provided the planner backbone, while Qwen-Image-2512 and Qwen-Image-Edit-2511 served as frozen generation and editing tools. Z-Image-Turbo was used for cross-generator evaluation. Model-generated images are included in the qualitative comparisons.

REFERENCES

[1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[2] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Kamyar Seyed Hosseini, Kamyar Ghasemipour, Raphael Gontijo Lopes, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pages 36479–36494, 2022.

[3] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, et al. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[4] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023.

[5] Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. GenEval 2: Addressing benchmark drift in text-to-image evaluation. *arXiv preprint arXiv:2512.16853*, 2025.

[6] Xingchen Wan, Han Zhou, Ruoxi Sun, Hootan Nakhost, Ke Jiang, Rajarishi Sinha, and Sercan O. Arik. Maestro: Self-improving text-to-image generation via agent orchestration. *arXiv preprint arXiv:2509.10704*, 2025.

[7] Shantanu Jaiswal, Mihir Prabhudesai, Nikash Bhardwaj, Zheyang Qin, Amir Zadeh, Chuan Li, Katerina Fragkiadaki, and Deepak Pathak. Iterative refinement improves compositional image generation. *arXiv preprint arXiv:2601.15286*, 2026.

[8] Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-DiT: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[9] Jinming Liu, Ruoyu Feng, Yuqi Wang, Wenjun Zeng, and Xin Jin. Generation navigator: A state-aware agentic framework for image generation. *arXiv preprint arXiv:2605.17969*, 2026.

[10] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[11] Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, Conghui He, and Weijia Li. Echo-4o: Harnessing the Power of GPT-4o Synthetic Images for Improved Image Generation. *arXiv preprint arXiv:2508.09987*, 2025. URL https://arxiv.org/abs/2508.09987.

[12] Z-Image Team, Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, Xin Jin, Liangchen Li, Zhen Li, Zhong-Yu Li, David Liu, Dongyang Liu, Qilong Wu, Feng Yu, Zechao Zhan, Chi Zhang, Shifeng Zhang, Ruikai Zhou, and Shilin Zhou. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

[13] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench: A comprehensive benchmark for open-world compositional text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pages 78723–78747, 2023.

[14] Xindi Wu, Dingli Yu, Yangsibo Huang, Olga Russakovsky, and Sanjeev Arora. ConceptMix: A compositional image generation benchmark with controllable difficulty. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[15] Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A. Smith. TIFA: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20406–20417, 2023.

[16] Jaemin Cho, Yushi Hu, Roopal Garg, Peter Anderson, Ranjay Krishna, Jason Baldridge, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. In *International Conference on Learning Representations*, 2024.

[17] Bangji Yang, Ruihan Guo, Jiajun Fan, Chaoran Cheng, and Ge Liu. M3: High-fidelity text-to-image generation via multi-modal, multi-agent and multi-round visual reasoning. *arXiv preprint arXiv:2602.06166*, 2026.

[18] Zefeng He, Siyuan Huang, Xiaoye Qu, Yafu Li, Tong Zhu, Yu Cheng, and Yang Yang. GEMS: Agent-native multimodal generation with memory and skills. *arXiv preprint arXiv:2603.28088*, 2026.

[19] Kaixun Jiang, Yuzheng Wang, Junjie Zhou, Pandeng Li, Zhihang Liu, Chen-Wei Xie, Zhaoyu Chen, Yun Zheng, and Wenqiang Zhang. GenAgent: Scaling text-to-image generation via agentic multimodal reasoning. *arXiv preprint arXiv:2601.18543*, 2026.

[20] Zekai Zhang, Jiahao Li, Jie Zhang, Kaiyuan Gao, Kun Yan, Lihan Jiang, Ningyuan Tang, Sheng-ming Yin, Tianhe Wu, Xiaoyue Chen, Xiao Xu, Yan Shu, Yanran Zhang, Yixian Xu, Yuxiang Chen, Zhendong Wang, Zihao Liu, Zikai Zhou, Huishuai Zhang, Dongyan Zhao, and Chenfei Wu. Qwen-Image-Agent: Bridging the context gap in real-world image generation. *arXiv preprint arXiv:2606.26907*, 2026.

[21] Junjie Wang, Xinghua Lou, Jason Li, Ye Tian, Keyu Chen, Yulin Li, Bin Kang, Jacky Mai, Yanwei Li, Zhuotao Tian, and Liqiang Nie. Benchmarking and evolving reason-reflect-rectify for reflective visual generation. *arXiv preprint arXiv:2605.19639*, 2026.

[22] Bowen Ping, Xiangxin Zhou, Penghui Qi, Minnan Luo, Liefeng Bo, and Tianyu Pang. Flow-DPPO: Divergence proximal policy optimization for flow matching models. *arXiv preprint arXiv:2606.11025*, 2026. URL https://arxiv.org/abs/2606.11025.

[23] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

[24] Qwen Team. Qwen-Image-2512: Official model card. Hugging Face, 2025. URL https://huggingface.co/Qwen/Qwen-Image-2512.

[25] Qwen Team. Qwen-Image-Edit-2511: Official model card. Hugging Face, 2025. URL https://huggingface.co/Qwen/Qwen-Image-Edit-2511.

[26] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhong Gou, Feng Li, Zeyu Wang, Shuyuan Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[27] Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pages 66923–66939, 2023.

[28] Nanye Ma, Shan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Ming Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhong Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

[29] Jun He, Junyan Ye, Zilong Huang, Dongzhi Jiang, Chenjue Zhang, Leqi Zhu, Renrui Zhang, Xiang Zhang, and Weijia Li. Mind-Brush: Integrating Agentic Cognitive Search and Reasoning into Image Generation. *arXiv preprint arXiv:2602.01756*, 2026.

[30] Tianfei Ren, Zhipeng Yan, Yiming Zhao, Zhen Fang, Yu Zeng, Guohui Zhang, Hang Xu, Xiaoxiao Ma, Shiting Huang, Ke Xu, Wenxuan Huang, Lionel Z. Wang, Lin Chen, Zehui Chen, Jie Huang, and Feng Zhao. SCOPE: Structured Decomposition and Conditional Skill Orchestration for Complex Image Generation. *arXiv preprint arXiv:2605.08043*, 2026.

[31] Viacheslav Kovalev, Alexander Kuvshinov, Alexander Buzovkin, Dmitry Pokidov, and Denis Timonin. CRAFT: Continuous reasoning and agentic feedback tuning for multimodal text-to-image generation. *arXiv preprint arXiv:2512.20362*, 2025.

[32] Harold Haodong Chen, Zhiyu Hou, Wen-Jie Shu, Weilin Ruan, Yingjie Xu, Litao Guo, and Ying-Cong Chen. GenRouter: Unified workflow routing for agentic image generation. *arXiv preprint arXiv:2608.16721*, 2026.

[33] Zongxia Li, Dawei Liu, Fuxiao Liu, Yuhang Zhou, Xiyang Wu, Jingxi Chen, Jing Xie, Xiaomin Wu, and Lichao Sun. COMFYCLAW: Self-Evolving Skill Harnesses for Image Generation Workflows. *arXiv preprint arXiv:2607.01709*, 2026.

[34] Le Zhuo et al. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. *arXiv preprint arXiv:2504.16080*, 2025.

[35] Liyao Jiang, Ruichen Chen, Chao Gao, and Di Niu. RAISE: Requirement-adaptive evolutionary refinement for training-free text-to-image alignment. *arXiv preprint arXiv:2603.00483*, 2026.

[36] Young Kyung Kim, Oded Schlesinger, Yuzhou Zhao, J. Matias Di Martino, and Guillermo Sapiro. Chain-of-image generation: Toward monitorable and controllable image generation. *arXiv preprint arXiv:2512.08645*, 2025.

[37] Xu Jiang, Bin Chen, Gehui Li, Yule Duan, Ronggang Wang, and Jian Zhang. OctoT2I: A self-evolving agentic text-to-image router. *arXiv preprint arXiv:2606.01803*, 2026.

[38] Junyan Ye, Jun He, Zilong Huang, Dongzhi Jiang, Xuan Yang, Rui Chen, and Weijia Li. GenClaw: Code-Driven Agentic Image Generation. *arXiv preprint arXiv:2605.30248*, 2026. URL `https://arxiv.org/abs/2605.30248`.

[39] Yongjin Kim, Yoonjin Oh, Yerin Kim, Hyomin Kim, Jeeyoung Yun, Yujung Heo, Minjun Kim, and Sungwoong Kim. FiRe: Fine-grained multimodal reasoning for enhanced image generation. *arXiv preprint arXiv:2604.13491*, 2026.

[40] Chieh-Yun Chen, Min Shi, Gong Zhang, and Humphrey Shi. T2I-Copilot: A training-free multi-agent text-to-image system for enhanced prompt interpretation and interactive generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19396–19405, 2025.

[41] Chunhan Li, Qifeng Wu, Jia-Hui Pan, Ka-Hei Hui, Jingyu Hu, Yuming Jiang, Bin Sheng, Xihui Liu, Wenjuan Gong, and Zhengzhe Liu. coDrawAgents: A multi-agent dialogue framework for compositional image generation. *arXiv preprint arXiv:2603.12829*, 2026.

APPENDIX OVERVIEW

Implementation details, training data, the continuation-state argument, and recorded cases are organized below. Section E reports the continuation protocols and cross-benchmark category results.

A Experimental Setup and Interaction Protocol 15
- A.1 Models and Frozen Components 15
- A.2 Decision Inputs, Actions, and Candidate Access 15
- A.3 Training Settings 17
- A.4 Benchmark and Evaluation Protocols 18

B Teacher Data and Selective Supervision 19
- B.1 Prompt Selection and Teacher Collection 19
- B.2 Action-Level Target Selection 19
- B.3 Decision-Time Inputs and Loss Masks 20
- B.4 Data Splits and Corpus Statistics 20

C Reinforcement Learning Details 21
- C.1 Rollout Initialization and Policy References 21
- C.2 Terminal Rewards and Group Advantages 21
- C.3 Reference Utility and Local Progress 22
- C.4 Token-Weighted Credit Allocation 23
- C.5 Response Masks and Policy Optimization 25

D Optimization Potential of Historical Continuation 26
- D.1 Interaction State and Continuation State 26
- D.2 Feasible-Set Monotonicity 26
- D.3 Interpretation and Scope 26

E Analysis Protocols and Additional Results 27
- E.1 Controlled Teacher Continuations 27
- E.2 Learning from Joint Demonstrations 27
- E.3 Historical-Source Access after Regression 27
- E.4 Cross-Benchmark Category Results 27

F Case Studies and Qualitative Comparisons 29
- F.1 Joint and Best-only Continuations 29
- F.2 Historical Continuation and Latest-only Refinement 30
- F.3 Current Quality and Continuation Potential 31
- F.4 Action Credit on a Recorded Trajectory 32
- F.5 Additional Qualitative Outputs 33

G Prompts and Tool Interfaces 37
- G.1 Teacher and Student Prompt Contracts 37
- G.2 Action Fields and Examples 37
- G.3 Guidance Retrieval and Verifier Interfaces 39

H Limitations 51

A Experimental Setup and Interaction Protocol

This appendix separates the planner's decisions from the information and rules supplied by the environment. We first describe the models, candidate access, and training settings. Appendix B specifies the teacher supervision, and Appendix C gives the RL objective and credit-allocation algorithm. The training chain is Joint selective SFT followed by five rounds of NACA RL.

A.1 Models and Frozen Components

The planner is initialized from Qwen3-VL-8B-Instruct [23], and GPT-5.5 supplies teacher trajectories. Generation and editing use separate, frozen backends: Qwen-Image-2512 [24] creates a new image, while Qwen-Image-Edit-2511 [25] modifies the selected candidate. The environment routes the action to the corresponding backend. The planner specifies the image content and, for editing, the image to modify; it does not select a model, provider, sampling configuration, or image seed.

Table A.1: Model roles and trainable components. Backend routing is controlled by the environment, not predicted in an action.

Role	Model or component	Use
Planner	Qwen3-VL-8B-Instruct	Language parameters are updated; vision tower and multimodal projector are frozen.
Teacher	GPT-5.5	Collects multi-round demonstrations with executed actions and verifier feedback.
Generator	Qwen-Image-2512	Produces a new candidate without an editing input.
Editor	Qwen-Image-Edit-2511	Edits the candidate named in the action.
Verifier	Qwen3-VL-8B-Instruct	Frozen VQA evaluator; provides atomic feedback and Soft-TIFA scores.
Guidance	Fixed skill library	Returns requested guidance; does not choose the next action or image.

The cross-generator evaluation replaces the generation backend with Z-Image-Turbo without additional planner training. The generation and editing roles should be distinguished: an interface for replacing the generator does not, by itself, specify a different editing model. The exact evaluation-time tool binding belongs to the corresponding run configuration.

A.2 Decision Inputs, Actions, and Candidate Access

A task consists of an original prompt, a fixed set of atomic constraints, and an image-call budget of five. Constraint identifiers and evaluator questions remain fixed throughout the episode. After each image action, the verifier records a status for every constraint and the correct-answer probabilities used to calculate Soft-TIFA GM. Statuses distinguish `pass`, `fail`, and `uncertain`; uncertainty is not silently replaced by a confident failure or success.

The Joint input presents all completed candidates in completion order, together with their images, evaluations, actual parent identifiers, and executed actions. Image pixels are supplied as multimodal message content and explicitly bound to candidate and artifact identifiers. They are not an additional field generated by the planner. Table A.2 summarizes the information available for a decision; the exact interface fields are given in Appendix G.

One structured action. Each response contains exactly one of generation, editing, guidance retrieval, and submission. An editing response jointly specifies the candidate identifier, target constraints, preservation constraints, and executable instruction. These are parts of one action, not independent image-selection and instruction-generation stages. Targets and preservation constraints must be disjoint and refer to the fixed task. Preservation fields specify the content to retain during editing.

Access and ranking. Let candidate j have GM g_j, passed-constraint count P_j, and completion index j. The Joint control ranks candidates lexicographically by $(g_j, P_j, -j)$. It also retains the

Table A.2: Decision-time information in the Joint input. All entries are environment-provided inputs.

Component	Information
Task	Original prompt, fixed atomic requirements and questions, maximum image attempts.
Candidate history	Every completed candidate, its bound image, parent identifier, GM, constraint results, and the action that produced it.
Control	Current rankings, visible and editable candidate identifiers, legal submissions, and remaining image budget.
Retrieved guidance	Actual returned skill content and its associated target constraints.
Query control	Available skills, completed and pending queries, and remaining query allowance.
Action error, when present	The last unresolved format or validation error, used to request a new complete action.

underlying reducer's pass-first identifier, ranked by $(P_j, g_j, -j)$, under an explicitly different name. The GM-best identifier controls submission. The planner can edit any completed candidate, whether or not it is latest, GM-best, or uniquely satisfies a constraint. There is no prescribed fraction of historical edits and no requirement to avoid the current best.

Table A.3: Candidate permissions under the GM-control interface. Rankings and permissions are recomputed after each completed image. This table defines the interfaces, not additional trained variants.

Interface	Visible candidates	Editable candidates	Submission
Joint	All completed images	All completed images	GM-best
Best-only	All completed images	Current GM-best	GM-best
Latest-only	All completed images	Current latest	GM-best

Three distinct uses of image scores

These control rankings are distinct from the RL utility $U(I) = P(I) + 0.25g(I)$ and from the source-relative GM criterion for SFT selection. The control determines which submission is legal; SFT selection determines which demonstrated actions are imitated; RL utility determines rewards and local progress. None of these roles is substituted for another.

Budget and termination. The first generation is included in the five-image budget. A new generation creates a root candidate; an edit creates a child of the declared input image. Neither restarting nor editing an earlier image resets the budget or removes later observations. Guidance retrieval and submission do not create image candidates. Infrastructure retries are handled by the execution layer rather than recorded as fictitious low-scoring candidates.

With no completed image, the agent cannot edit or submit. Once the image budget is exhausted, it must submit the GM-best with the budget reason. Otherwise, if that candidate passes all constraints, it must submit with the all-pass reason. If neither condition holds, it may continue or submit with `no_productive_action_remaining`. Budget exhaustion takes precedence when the last image is also all-pass. The planner decides whether to stop early when neither mandatory stopping condition holds. Submission uses the fixed GM-best rule.

History and guidance. The environment rebuilds persistent state from validated actions, execution artifacts, and evaluation events. Failed or ineffective executed edits remain available as evidence. Raw rejected output does not become persistent free-form memory, although an unresolved action error can be returned for correction. Candidate records carry the image history and executed actions, rather than a separate free-form memory summary. Internal transition records may contain more detail than the student input.

At most two successful skill queries are permitted between completed image evaluations. Pending queries and already active skills cannot be queried again. Returned guidance supports the planner's

instruction but does not choose its editing input, alter candidate permissions, or add image calls. The six documented skills and their interface are listed in Appendix G.3.

A.3 Training Settings

Selective SFT establishes the structured joint decision behavior. The resulting checkpoint initializes five RL rounds. Only the planner's language parameters are optimized; image backends, verification, and the planner's visual components remain frozen. Tables A.4 and A.5 separate the supervision/export settings from the RL optimizer settings.

Table A.4: SFT settings and supervision interface, aligned with the main-paper configuration.

Setting	Value
Planner initialization	Qwen3-VL-8B-Instruct
Prompt membership	800 train / 100 validation / 100 test
Teacher prompt pool	1,000 prompts across the three splits
Learning rate / global batch size / epochs	10^{-5} / 32 / 2
Selected action targets	3,473 train / 437 validation / 413 test
Targets per exported record	One canonical assistant action
Training target	Complete action response, including image choice and edit specification
Context loss	Masked; task, images, previous actions, and feedback remain inputs
Data format	LLaMA-Factory action-response records
Supervision scope	Action response only; no separate reasoning target
Frozen modules	Vision tower, multimodal projector, external backends and verifier

Table A.5: RL settings for the five-round training chain. Hardware is specified only for the final round where it was confirmed.

Setting	Value
Training protocol	
Initialization	Joint selective SFT checkpoint
Rounds / prompt groups per round	5 / 64
Rollouts per group	4
Collected trajectories	256 per round; 1,280 total
Optimizer steps	8 per round; 40 total
Configured update batch	8 prompt groups, or 32 trajectories, before filtering
Data reuse / image budget	1 epoch per round / 5 calls per trajectory
Sampling and references	
Temperature / top-p / top-k	0.7 / 0.95 / disabled (-1)
Rollout policy	Preceding checkpoint, fixed during collection
KL reference	Frozen parent checkpoint, refreshed each RL round
Optimization and credit	
Optimizer / weight decay	AdamW / 0
Initial learning rate	10^{-6}, with linear decay within each round
Clipping offsets / KL coefficient	$(0.20, 0.28)$ / 0.02
Entropy coefficient / gradient clip	0 / 1
Loss reduction	Trajectory mean, sampled-response token sum
Microbatch and accumulation	One action segment at a time; accumulate the step's assigned segments
Utility coefficient / nominal cap	$\alpha = 0.25$ / 8; effective cap can be higher
Nonpositive advantages	Broadcast unchanged
Credit preparation	After rollout, using execution events and response masks
Final-round distributed training	FSDP with four ranks

The 320 prompt-group instances need not correspond to 320 distinct prompts. The configured batch size is not a fixed gradient-accumulation count: the number of action segments depends on the sampled trajectories. Groups with effectively zero reward variance are excluded from optimizer data. The threshold, dynamic cap, and exact loss are defined in Appendix C.

A.4 Benchmark and Evaluation Protocols

GenEval2 [5] is the primary compositional benchmark, with final image quality reported using Soft-TIFA GM. For the fixed K atomic questions associated with an image I, let $p_k(I)$ be the verifier's probability of the correct answer. The documented score computation is

$$g(I) = \exp\left(\frac{1}{K}\sum_{k=1}^{K}\log\max\{p_k(I), 10^{-300}\}\right). \tag{A.1}$$

Missing, non-finite, or out-of-range probabilities are rejected before aggregation. GM is not the proportion of passed constraints. Discrete statuses support planning and pass-count computation, while the continuous probabilities define the GM score.

The reported GenEval2 datasets contain 800 prompts. The planner's ordinary evaluation decoding is greedy, with a maximum of five generation or editing calls including the initial generation. Evaluation scores refer to the selected submitted image rather than automatically to the last generated image. Cross-generator evaluation retains the planner weights and changes the generation backend without additional training.

The Qwen-backed evaluation reports the final submitted images for all 800 prompts, yielding 84.3 GM. We also rescore the same 800 submitted images with the official evaluator. On the GM$\times 100$ scale, the absolute difference between the online and official mean scores is 0.00000082, and the maximum per-image absolute difference is 0.00000437. Thus, online scoring and official rescoring agree numerically on this evaluation set.

Execution validates action structure and source eligibility before dispatching image calls. Error handling uses action-validity feedback, deterministic target–preserve overlap resolution, and source-ID correction under the applicable source-eligibility rule. Editing instructions remain planner-generated, and continuation retains the executed prefix and remaining image budget. The current Joint prompt and the archived historical-access protocol are specified separately in Appendices G.5 and E.3.

GenEval [4] and GenEval++ [11] provide the cross-benchmark evaluations in the main paper. Their benchmark-specific results are reported separately from GenEval2 Soft-TIFA GM. External baseline values are taken from the cited publications, author-provided results, or reproductions under the configurations documented by those sources.

B TEACHER DATA AND SELECTIVE SUPERVISION

The teacher data provide examples of complete decisions under the history-conditioned interface. Supervision is selected after actions have been executed, but each training input contains only the information available before its target action. This separates useful evidence from behavior that the student should imitate.

B.1 PROMPT SELECTION AND TEACHER COLLECTION

GPT-5.5 collects multi-round trajectories on prompts from the GenEval2 training pool used by Flow-DPPO [22]. The original prompt split contains 800 training, 100 validation, and 100 test memberships. These are prompt counts, not counts of executed trajectories or selected action responses. The teacher prompt pool contains 1,000 prompts across these splits; the selected action responses are summarized in Table B.2.

Teacher collection uses the same four operation types and fixed generation/editing roles described in Appendix A. At an editing step, the teacher selects the image and the corresponding instruction together. It can use completed historical candidates, their evaluations, and earlier executed instructions, rather than restricting editing to a preselected best image.

Teacher collection and student training use different input wrappers. The teacher receives task, planner-context, candidate-observation, and continuation-control objects. The student receives a dedicated structured decision-time input. Both expose the GM-control permissions, but the teacher's longer instructions are not claimed to be the student's verbatim system prompt. In particular, the teacher's four-part editing guidance is a semantic structure inside one instruction string, not four additional action fields.

B.2 ACTION-LEVEL TARGET SELECTION

Selective supervision retains individual actions using their observed outcomes. Let g_{child} denote the new image's GM, g_{edit} the GM of the image being edited, and g_{archive} the largest GM available before the action. The improvement tolerance is $\varepsilon_{\mathrm{SFT}} = 10^{-9}$.

Table B.1: Selection rules for positive SFT targets. All comparisons use the record before the candidate action and its actual execution outcome.

Action	Selection rule
Initial generation	Retain the required first generation.
Editing	Retain when $g_{\mathrm{child}} > g_{\mathrm{edit}} + \varepsilon_{\mathrm{SFT}}$.
Subsequent generation	Retain when $g_{\mathrm{child}} > g_{\mathrm{archive}} + \varepsilon_{\mathrm{SFT}}$.
Submission	Retain the actually executed submission when it selects the current GM-best and uses a valid reason.
Guidance retrieval	Require a real returned response. The first subsequent executed image action must itself be selected, and its target constraints must intersect the query's target constraints.

An edit can be selected even when its output remains below the archive best, provided that it improves its own editing input. Constraint-level regressions are recorded separately and do not automatically veto a GM-positive target. This SFT rule is not the pass-count-first utility used in RL.

Actions that do not meet the selection rule are not imitation targets. Their validated execution records, images, and feedback remain available to later decisions. A GM-decreasing bridge action does not become a positive target merely because a later descendant succeeds. The student can therefore learn a later recovery without being trained to reproduce the earlier degrading action.

The export distinguishes a narrower recovery-positive label: an edit uses a non-latest image and exceeds the pre-action GM archive. Other retained actions are other positive. These labels should not be confused with editing an earlier candidate alone. Source-relative GM change, archive-relative GM change, and changes in passed constraints are recorded for audit; they are not additional planner outputs.

B.3 Decision-Time Inputs and Loss Masks

Each selected target is one canonical assistant action paired with its decision-time input. The input is reconstructed from the preceding immutable event prefix and contains the associated candidate images in the documented order. The target's execution outcome is used only for offline selection; it is not inserted into that action's input. Previously completed unsuccessful actions remain legitimate context for later targets.

An exported row contains a system message, a user message with the rendered multimodal state, and one assistant action. Only the assistant response is supervised. For editing, this includes the candidate choice, target and preservation sets, and complete instruction. The export uses the action-response template with prompt training disabled. The number of image placeholders must equal the number of bound images; the binding preserves their identities rather than relying on an unlabelled positional guess.

For serialized sequence z_j, define $m_{j,\ell} = 1$ on its selected response tokens and zero elsewhere. The objective is

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{Z} \sum_{j} \sum_{\ell} m_{j,\ell} \log \pi_\theta(z_{j,\ell} \mid z_{j,<\ell}), \qquad Z = \sum_{j} \sum_{\ell} m_{j,\ell}. \tag{B.1}$$

The associated images are conditioning inputs. Excluding a failed action from the target does not remove its earlier occurrence from a later input, and does not assign it a negative imitation loss. The resulting policy initializes the five-round RL training chain.

B.4 Data Splits and Corpus Statistics

All trajectories for the same prompt stay in the same split. Records are deduplicated using the split, rendered input, the identities of the ordered images, and canonical target action. Copying an identical shared prefix therefore does not create duplicate supervision weight; different real actions at the same state can remain distinct targets. Training samples action rows without an additional episode-level weighting rule.

Table B.2: Selected targets in the current Joint SFT export. Counts are action responses, not distinct prompts or raw trajectories.

Split	Generate	Edit	Retrieve	Submit	Total
Train	961	1,076	584	852	3,473
Validation	108	140	103	86	437
Test	110	114	99	90	413
Total	1,179	1,330	786	1,028	4,323

The export records 6,858 image bindings and 3,418 unique image artifacts across the selected records. These quantities differ because the same image can appear in more than one decision context. Training uses 800 distinct prompts; action-response counts in Table B.2 are not prompt counts. Validation and test submission-target totals likewise do not establish their raw collection sizes.

Validation and test rows are rendered and selected under the same compatibility rules as training rows. Their next-token losses measure conditional action modeling. They are not new online image-generation evaluations and should not be reported as final benchmark image quality.

C Reinforcement Learning Details

Starting from the Joint SFT checkpoint, we train the planner for five RL rounds with NACA. This section specifies the rollout state, per-round policy references, rewards, token-weighted action credit, and optimization loss. Training settings are summarized in Table A.5.

C.1 Rollout Initialization and Policy References

A group shares a task, not a generated image. A GRPO group contains $G = 4$ complete trajectories with the same prompt, fixed atomic constraints, and initial budget of five image calls. Each trajectory begins before image generation, with no completed image candidates or image evaluations. It executes its own first `generate_image` call and then its own subsequent actions. No shared first image is supplied to the four trajectories.

For the initial generation, the image seed is `base_seed + candidate_index`. Here, `candidate_index` distinguishes rollout candidates in the group, not image-attempt identifiers within a trajectory. Thus first images may differ. This rule does not assert that every later image call uses that same seed formula. The first generation counts toward each trajectory's five-image budget and is included in NACA's image-action credit calculation.

This differs from a matched-prefix continuation experiment, which copies an already executed and evaluated prefix into comparison arms. Such a prefix can contain a shared first image or several shared images, leaving four or fewer image calls. That continuation protocol does not describe the initialization of these RL groups.

Training-round references. Let $\pi^{(0)}$ be the Joint selective SFT policy and $\pi^{(r)}$ the checkpoint after RL round r, for $r = 1, \ldots, 5$. Round r collects trajectories using $\pi^{(r-1)}$ and initializes the trainable planner from the same checkpoint. A frozen copy of $\pi^{(r-1)}$ provides the KL reference:

$$\pi_{\text{old}}^{(r)} = \pi^{(r-1)}, \qquad \pi_{\text{ref}}^{(r)} = \text{freeze}(\pi^{(r-1)}). \tag{C.1}$$

The two policies have identical starting weights but different roles: $\pi_{\text{old}}^{(r)}$ supplies rollout probabilities, while $\pi_{\text{ref}}^{(r)}$ anchors the KL penalty. Their stored log-probabilities remain fixed as π_θ is updated within the round. The next round loads the newly trained checkpoint rather than retaining the original SFT policy as its reference.

Each round completes collection before optimization. Recorded execution events and sampled-response masks are then used to compute NACA advantages, which are stored alongside the rollout data and consumed by the optimizer. Each round uses one epoch and eight optimizer steps. In the final round, the learning rates are $10^{-6}, 8.75 \times 10^{-7}, \ldots, 1.25 \times 10^{-7}$. Only planner language parameters are updated; the vision tower, multimodal projector, image backends, and verifier remain frozen.

C.2 Terminal Rewards and Group Advantages

For an image I, utility is $U(I) = P(I) + 0.25g(I)$, where $P(I)$ counts passed constraints and $g(I)$ is the Soft-TIFA geometric mean. The terminal reward uses the image explicitly submitted by the planner, not an automatically substituted historical best. The RL utility is intentionally distinct from the GM used for benchmark reporting and GM-based image selection. For two candidate images, its difference can be written as

$$\Delta U = \Delta P + 0.25\,\Delta g. \tag{C.2}$$

Because $g(I) \in [0, 1]$, an increase of at least one in passed-constraint count satisfies

$$\Delta P \geq 1 \quad \Longrightarrow \quad \Delta U \geq 1 - 0.25 = 0.75 > 0.$$

Thus, the RL utility intentionally prefers an image that passes more constraints, even when its GM is lower. When the pass count is unchanged, the utility preserves the GM ordering. SFT target selection and final image selection continue to use their specified GM-based criteria. A policy-invalid trajectory receives $R_i = -1$ and remains in its prompt group when computing the reward mean

Table C.1: Termination and failure handling during RL data preparation.

Case	Handling
Normal submission	Reward is the utility of the submitted image.
Policy-invalid trajectory	Reward is -1 and participates in group statistics. Retained sampled responses are trained if their group is not filtered.
Image backend, transport, or verifier failure	Retry or recover; do not substitute a score or treat the failure as a policy violation.
Image budget exhausted, followed by valid submission	Score the submitted image normally; budget exhaustion alone adds no penalty.
No valid submission	Use the recorded termination class to distinguish policy invalidity from incomplete execution or infrastructure failure; do not invent a submission.
Zero-variance group	Remove the group from optimizer data, including KL.

and variance. Infrastructure failures are handled through retry or recovery; they are not assigned fabricated image scores or invalid-policy rewards.

For a group of $G = 4$ trajectories, define

$$\bar{R} = \frac{1}{G}\sum_{i=1}^{G} R_i, \qquad \sigma_R = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(R_i - \bar{R})^2}, \tag{C.3}$$
$$A_i = \begin{cases} (R_i - \bar{R})/\sigma_R, & \sigma_R > 10^{-12}, \\ 0, & \text{otherwise.} \end{cases}$$

No additive epsilon is used in the denominator. Groups at or below the standard-deviation threshold are excluded from optimizer assignment, so they contribute neither a policy-gradient term nor a KL term.

Policy-invalid cases include parsing errors, invalid candidate references, and action-rule violations. In the final round, 16 of 256 trajectories were policy-invalid (6.25%); these trajectories were retained rather than removing their groups solely for containing a policy error.

C.3 Reference Utility and Local Progress

Let $\mathcal{V}_{i,t}$ contain the evaluated image candidates available before action $a_{i,t}$, and let $I_{i,k_{i,t}}$ be the image selected for an edit. The reference utility is

$$U^{\text{ref}}_{i,t} = \begin{cases} 0, & \text{initial generation,} \\ U(I_{i,k_{i,t}}), & \text{editing,} \\ \max_{I \in \mathcal{V}_{i,t}} U(I), & \text{subsequent generation.} \end{cases} \tag{C.4}$$

An edit is measured against the image it actually modifies. A new generation has no selected image to edit, so its progress is measured against the highest utility already available. This reference uses U, not the GM-first submission ranking.

For a task with K constraints, let $D = K + 0.25$. For an action producing $I_{i,t}$, normalized local progress is

$$q_{i,t} = \begin{cases} \dfrac{[U(I_{i,t}) - U^{\text{ref}}_{i,t}]_+}{D - U^{\text{ref}}_{i,t}}, & U^{\text{ref}}_{i,t} < D, \\ 0, & U^{\text{ref}}_{i,t} = D, \end{cases} \tag{C.5}$$

where $[z]_+ = \max(z, 0)$. The normalization measures the fraction of the remaining utility gap closed by an action, allowing both initial generation and later repair to receive credit without comparing their absolute gains alone.

Positive progress depends on utility, not on every individual constraint or GM increasing. For example, an increase in passed constraints can raise utility despite a lower GM. Conversely, a non-increasing utility gives $q_{i,t} = 0$. An edit can receive positive progress without surpassing the historical best. These are local comparisons, not estimates of an action's causal contribution to the final submitted image.

C.4 TOKEN-WEIGHTED CREDIT ALLOCATION

An image action is eligible when its evaluation has completed, it is not policy-invalid, and contains at least one trainable response token. Let $\mathcal{I}_i$ index these actions and $L_{i,t} > 0$ be the number of mask-selected tokens in action $a_{i,t}$. With $\delta = 10^{-6}$, define

$$\begin{aligned} T_i &= \sum_{t \in \mathcal{I}_i} L_{i,t}, \quad E_i = \{t \in \mathcal{I}_i : q_{i,t} > \delta\}, \\ T_i^+ &= \sum_{t \in E_i} L_{i,t}, \quad Z_i = \sum_{t \in \mathcal{I}_i} L_{i,t} q_{i,t}. \end{aligned} \tag{C.6}$$

Only positive trajectory advantages are redistributed. If E_i is empty or $Z_i \le \delta$, all eligible image actions use $w_{i,t} = 1$. Otherwise, the effective cap and weights are

$$\begin{aligned} c_i &= \max\left(8, \frac{T_i}{T_i^+}\right), \\ w_{i,t} &= \begin{cases} \min(c_i, \lambda_i q_{i,t}), & t \in E_i, \\ 0, & t \in \mathcal{I}_i \setminus E_i, \end{cases} \qquad \sum_{t \in \mathcal{I}_i} L_{i,t} w_{i,t} = T_i. \end{aligned} \tag{C.7}$$

Thus, 8 is a nominal cap, not a global maximum. The constraint requires $c_i T_i^+ \ge T_i$: otherwise even saturating every positive-progress action cannot redistribute the full token-weighted total. The implementation raises an infeasible nominal cap to T_i/T_i^+ and records that adjustment. There is no additional fixed upper bound.

The scale λ_i is found with 100 bisection iterations. The initial bracket is $[0, \max(1, c_i / \min_{t \in E_i} q_{i,t})]$. Credit conservation is checked to tolerance $\max(10^{-6}, T_i 10^{-10})$. Algorithm C.1 summarizes the allocation. There is no positive weight floor: in the normal allocation branch, actions with $q_{i,t} \le 10^{-6}$ receive zero weight.

Algorithm C.1 NACA weights for one trajectory

Require: A_i, eligible image actions $\mathcal{I}_i$, token counts $L_{i,t}$, local progress $q_{i,t}$; $\delta = 10^{-6}$
Set $w_{i,t} \leftarrow 1$ for all $t \in \mathcal{I}_i$.
if $A_i \le 0$ **then**
  return w_i
end if
Compute T_i, E_i, T_i^+, and Z_i using Eq. equation C.6.
if $E_i = \varnothing$ or $Z_i \le \delta$ **then**
  return w_i ▷ Uniform fallback
end if
$c_i \leftarrow \max(8, T_i/T_i^+)$; record a cap adjustment if $8 < T_i/T_i^+$.
$\lambda_{\text{lo}} \leftarrow 0$; $\lambda_{\text{hi}} \leftarrow \max(1, c_i / \min_{t \in E_i} q_{i,t})$.
for $n = 1, \ldots, 100$ **do**
  $\lambda \leftarrow (\lambda_{\text{lo}} + \lambda_{\text{hi}})/2$
  $S \leftarrow \sum_{t \in E_i} L_{i,t} \min(c_i, \lambda q_{i,t})$
  if $S < T_i$ **then**
    $\lambda_{\text{lo}} \leftarrow \lambda$
  else
    $\lambda_{\text{hi}} \leftarrow \lambda$
  end if
end for
Form w_i from the resulting scale using Eq. equation C.7.
Check $|\sum_{t \in \mathcal{I}_i} L_{i,t} w_{i,t} - T_i| \le \max(10^{-6}, T_i 10^{-10})$.
return w_i

The resulting action advantage is

$$\widetilde{A}_{i,t} = \begin{cases} A_i w_{i,t}, & A_i > 0 \text{ and } t \in \mathcal{I}_i, \\ A_i, & \text{otherwise.} \end{cases} \tag{C.8}$$

For a positive trajectory, the allocation preserves $\sum_{t \in \mathcal{I}_i} L_{i,t} \widetilde{A}_{i,t} = A_i T_i$. This is conservation of advantage coefficients, not of the loss, gradient norm, or trajectory-length-normalized weight.

Nonpositive advantages remain unchanged across the trajectory; weighting negative advantages by local progress could otherwise penalize improving actions more strongly. Retrieval, submission, and other retained response segments outside $\mathcal{I}_i$ also retain their original trajectory advantage.

C.5 Response Masks and Policy Optimization

The response mask selects every sampled token in an action response. For a valid response, training therefore covers the full action JSON, including the image choice, target and preserve sets, instruction, and structural tokens. The protocol uses direct action JSON without a separate reasoning channel or reasoning mask. If extra text makes a response invalid, its retained sampled tokens can still receive the invalid trajectory's training signal. History, tool results, and verifier feedback are conditioning inputs, not sampled response targets.

Let $\mathcal{B}$ be the trajectories retained for an optimizer step, $\mathcal{T}_i$ the mask-selected response tokens in trajectory i, and $\widetilde{A}_{i,\ell}$ the advantage of the action containing token $y_{i,\ell}$. For its multimodal prefix $H_{i,\ell}$, define

$$\begin{aligned} \rho_{i,\ell} &= \frac{\pi_\theta(y_{i,\ell} \mid H_{i,\ell})}{\pi_{\text{old}}(y_{i,\ell} \mid H_{i,\ell})}, \\ d_{i,\ell} &= \log \pi_{\text{ref}}(y_{i,\ell} \mid H_{i,\ell}) - \log \pi_\theta(y_{i,\ell} \mid H_{i,\ell}), \\ \mathcal{K}_{i,\ell} &= \exp(d_{i,\ell}) - d_{i,\ell} - 1. \end{aligned} \tag{C.9}$$

This is the configured low-variance KL estimator. The corresponding minimization loss is

$$\begin{aligned} \mathcal{L}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{\ell \in \mathcal{T}_i} \Big[- \min\big\{ & \rho_{i,\ell} \widetilde{A}_{i,\ell}, \\ & \text{clip}(\rho_{i,\ell}, 0.8, 1.28) \widetilde{A}_{i,\ell} \big\} + 0.02 \mathcal{K}_{i,\ell} \Big]. \end{aligned} \tag{C.10}$$

The loss uses trajectory-mean, token-sum reduction: tokens are summed within trajectories, then trajectories are averaged. There is no division by individual trajectory length or the total token count. Group statistics use the four returns within each prompt group; $|\mathcal{B}|$ instead counts the retained trajectories assigned to an optimizer step, potentially spanning several prompt groups. Longer responses therefore contribute more token-level terms; NACA's coefficient conservation does not remove this length effect.

The trainer processes one action segment at a time and accumulates all segments assigned to an optimizer step. Multiplying local loss sums by the world size divided by the retained trajectory count, followed by FSDP gradient averaging, gives the reduction in Eq. equation C.10. Accumulation is determined by the number of segments, not a fixed microstep count.

Zero action credit is not a loss mask

NACA changes the advantage but not the response mask. Under normal positive-credit allocation, an image action with $w_{i,t} = 0$ has no positive policy-gradient term but retains its KL term. KL uses the same response mask for generation, editing, retrieval, submission, and retained invalid responses. This differs from a zero-variance group, which is removed before optimization and contributes no KL.

Table C.2: Credit and KL for responses in retained optimizer data. The first two rows assume effective positive progress is available.

Response or input	Advantage	KL term
Eligible image action, $A_i > 0$, $q_{i,t} > \delta$	$A_i w_{i,t}$	Retained
Eligible image action, $A_i > 0$, $q_{i,t} \le \delta$	0	Retained
Eligible image action, positive-trajectory fallback	A_i	Retained
Image action in a nonpositive-advantage trajectory	A_i	Retained
Retrieval, submission, or retained invalid response	A_i	Retained
Context, tool results, and verifier feedback	Not a target	Excluded
Zero-variance group	Group excluded	Excluded

D Optimization Potential of Historical Continuation States

Historical-image access changes the set of continuations available to a policy. The result below concerns that feasible set and the optimal reachable value, not the performance of a particular learned checkpoint. It preserves the same terminal utility and controlled comparison as the main formulation.

D.1 Interaction State and Continuation State

The interaction state s_t contains the task, previous actions and observations, available images, and remaining budget. A continuation state is one image I_{k_t} chosen as the input to an edit. The policy produces

$$a_t = (o_t, \eta_t) \sim \pi_\theta(\cdot \mid s_t), \qquad \eta_t = (k_t, u_t) \quad \text{when } o_t = \textsc{Edit}, \tag{D.1}$$

where o_t is one of generation, editing, retrieval, and submission, and u_t contains the edit specification. The image choice and edit specification are components of the same decision. Selecting an earlier image does not remove observations obtained after its creation or restore spent image calls. Generation creates a new root; editing creates a child of the selected candidate.

D.2 Feasible-Set Monotonicity

Consider a decision state with interaction history H_t and b remaining image calls. A source-access rule Γ maps each reachable history H to the set $\Gamma(H)$ of images that may be selected as editing inputs. Hold the prompt, observations, frozen generation and editing backends, non-edit actions, submission set, and image-call accounting fixed. Let Π_Γ denote the policies respecting this access rule. Using $U(I) = P(I) + 0.25g(I)$, define

$$V^\star_\Gamma(H_t, b) = \sup_{\pi \in \Pi_\Gamma} \mathbb{E}_\pi[U(I_{\text{sub}}) \mid H_t, b]. \tag{D.2}$$

Proposition 1. Monotonicity under source-set expansion

Suppose $\Gamma_1(H) \subseteq \Gamma_2(H)$ for every history reachable from (H_t, b). Then

$$V^\star_{\Gamma_2}(H_t, b) \geq V^\star_{\Gamma_1}(H_t, b). \tag{D.3}$$

Proof. Every policy feasible under Γ_1 is also feasible under Γ_2: the expanded rule adds editing choices but removes none of the actions available to the restricted policy. Hence $\Pi_{\Gamma_1} \subseteq \Pi_{\Gamma_2}$, and optimizing the same terminal objective over the larger policy class cannot decrease its value. □

D.3 Interpretation and Scope

The inclusion must hold over reachable histories, not just at the initial branch point. A comparison that also changes observations, available generation actions, submission rules, or budgets is outside the stated controlled setting. Even when the inclusion holds, a trained policy need not realize the larger class's optimal value. The proposition neither guarantees strict improvement nor requires the planner to reject the current best image.

Current quality and continuation potential are distinct. Two candidates can preserve different useful content and require different edits, even when one currently has a lower score. Appendix F.3 gives an observed successful continuation from a lower-scoring candidate. Appendix F.1 reports an executed Joint/Best-only pair from a common prefix. The former is an observation about one trajectory; the latter compares two realized continuations. Neither is a proof that every edit from the higher-scoring image would fail.

E Analysis Protocols and Additional Results

E.1 Controlled Teacher Continuations

To quantify the benefit of joint selection, we select 200 prefixes from a pool of 800 teacher trajectories where the task remains unsolved after three image calls. Each prefix provides a shared starting history for Joint, Latest-only, and Best-only, with two image calls remaining. All three use the same teacher and interaction protocol, differing only in edit-source access. Latest-only uses the latest image, Best-only uses the current highest-GM image, and Joint selects the source together with its editing instruction. Joint achieves a mean final GM of 55.2, compared with 52.4 for Best-only and 50.2 for Latest-only. Its improvement over Best-only is 2.8 points (paired 95% CI: [0.46, 5.15]). Joint also increases the proportion of prefixes producing a new candidate above the shared prefix best from 57.0% with Best-only to 65.0%. These results support the value of joint selection beyond fixing the editing source to the current highest-scoring image.

E.2 Learning from Joint Demonstrations

A second experiment examines whether joint demonstrations lead to better performance after SFT. An additional 800 Best-only teacher trajectories are collected using the same prompts and initial images as the Joint trajectories. The two demonstration sets are used to train separate student policies via SFT. Both policies are then evaluated on the official GenEval2 evaluation set of 800 prompts under the same evaluation protocol used for our main results. The student trained on Joint demonstrations achieves 81.5 GM, compared with 77.5 for its Best-only-trained counterpart, an improvement of 4.0 points (paired 95% CI: [2.81, 5.18]). The advantage is thus observed not only in controlled teacher continuations but also in the benchmark performance of the trained policies.

E.3 Historical-Source Access after Regression

We additionally evaluate historical source access in a separate Student experiment on 240 degraded prefixes, where the latest candidate is strictly worse than the historical best. Here, success requires generating a new candidate that surpasses the prefix best; simply returning an existing historical image does not count. Allowing historical edit sources increases this discovery rate from 60.00% to 71.67%, an improvement of 11.67 percentage points (paired 95% CI: [5.00, 18.33], $p = 0.00109$).

The two arms inherit the same executed prefix and remaining image-call budget. Full-history can edit completed historical candidates; Latest-only can edit only the latest candidate. The 240 selected prefixes consist of 143 with two completed images and 97 with three. Both arms execute fresh suffixes. Discovery compares newly generated images with the frozen prefix best; selecting an already existing candidate does not count as discovery. The archived study ranks candidates by passed-constraint count followed by GM, rather than substituting the GM-first control of the current Joint interface.

The discovery counts are 172/240 and 144/240. There are 49 pairs favoring Full-history, 21 favoring Latest-only, and 170 ties. The paired 95% confidence interval is [5.00, 18.33] percentage points, and the two-sided exact paired test gives $p = 0.00109$. Mean executed suffix calls are 2.233 and 2.104: the matched quantity is the remaining budget upper bound. All 365 historical edits in the Full-history arm select the then-current best. This comparison therefore measures historical-source access versus latest-only access; the teacher study above separately compares Joint with Best-only.

E.4 Cross-Benchmark Category Results

Tables E.1 and E.2 give the category results corresponding to the main-paper overall scores, 0.9844 on GenEval and 0.8929 on GenEval++. Scores here use the 0–1 scale; the main cross-benchmark table displays percentages.

Table E.1: Cross-benchmark generalization. Bold scores indicate the best result in each column.

	GenEval						
Method	Single obj.↑	Two obj.↑	Counting↑	Colors↑	Position↑	Color attr.↑	Overall↑
DIRECT GENERATION							
FLUX.1-dev [34]	0.9900	0.8100	0.7500	0.8000	0.2100	0.4800	0.6700
Qwen-Image [3]	0.9900	0.9200	0.8900	0.8800	0.7600	0.7700	0.8700
AGENTIC GENERATION AND REFINEMENT							
T2I-Copilot† [40, 35]	0.9900	0.9100	0.6800	0.8600	0.5500	0.4600	0.7400
GEMS (Z-Image-Turbo) [18]	0.9900	0.9200	0.9100	0.8800	0.8300	0.6500	0.8600
FiRe-GRPO (3rd round) [39]	0.9900	0.9400	0.7200	0.9300	0.8500	0.7700	0.8700
Generation Navigator [9]	–	–	–	–	–	–	0.8843
ReflectionFlow [34]	**1.0000**	0.9800	0.9000	0.9600	0.9300	0.7200	0.9100
RAISE [35]	**1.0000**	**1.0000**	0.9500	**0.9800**	0.8300	0.8700	0.9400
coDrawAgents [41]	**1.0000**	0.9600	0.9400	0.9700	0.9500	0.8100	0.9400
CoIG [36]	0.9875	0.9899	0.8625	0.9255	0.9900	0.9350	0.9484
OctoT2I [37]	**1.0000**	0.9900	0.9500	0.9400	**1.0000**	0.8600	0.9600
GenRevisit	**1.0000**	0.9899	**0.9875**	0.9787	0.9800	**0.9700**	**0.9844**

Table E.2: Comparison on GenEval++. The seven categories follow the benchmark taxonomy. Higher is better; bold marks the highest displayed result in each column.

Method	**Image backend**	**Category-wise accuracy**							**Overall**
		Color	**Count**	**Color/ Count**	**Color/ Pos**	**Pos/ Count**	**Pos/ Size**	**Multi-Count**	
DIRECT GENERATION									
FLUX.1-dev [19]	FLUX.1-dev	0.4000	0.6000	0.2500	0.2500	0.0750	0.4000	0.3000	0.3250
BAGEL [11]	BAGEL	0.3250	0.6000	0.2500	0.3250	0.2500	0.4750	0.3750	0.3710
Qwen-Image [19]	Qwen-Image	0.8750	0.7250	0.7250	0.6000	0.4750	0.7250	0.5500	0.6680
Echo-4o [11]	BAGEL	0.8000	0.5750	0.5500	0.7750	0.6250	0.8000	0.6250	0.6790
CLOSED-SOURCE GENERATION									
GPT-4o [11]	GPT-4o	0.9000	0.6750	0.7250	0.6250	0.6000	0.8000	0.8500	0.7390
GPT-Image-1.5 [38]	GPT-Image-1.5	0.8500	0.9500	0.7750	0.5000	0.6000	0.7750	0.8000	0.7500
Gemini-3.1 Flash-Image [38]	Gemini-3.1	0.8500	0.8500	0.8000	0.7500	0.6750	0.7250	0.7750	0.7750
AGENTIC GENERATION AND REFINEMENT									
PromptEnhancer [19]	FLUX.1-dev	0.5000	0.6250	0.2250	0.3750	0.1250	0.4500	0.3750	0.3820
ReflectionFlow [19]	FLUX.1-dev	0.4000	0.6250	0.2750	0.2750	0.2000	0.4250	0.3250	0.3610
FiRe (3rd round) [39]	Janus-Pro-7B	0.7750	0.4880	0.4500	0.5690	0.2810	0.3560	0.4880	0.4910
GenAgent [19]	FLUX.1-dev	0.7500	0.6750	0.3750	0.5250	0.4500	0.6250	0.5250	0.5610
GenAgent [19]	Qwen-Image	0.7750	0.7750	0.6500	0.8000	0.6000	0.7250	0.7500	0.7250
Mind-Brush [29]	Qwen-Image	0.7750	0.7000	0.7750	0.7500	0.8500	0.7750	0.8500	0.7820
GenClaw [38]	Gemini-3.1	0.8750	0.9500	0.8250	**0.8500**	**0.9250**	0.8250	**0.9000**	0.8786*
OUR METHOD									
GenRevisit	Qwen-Image-2512	**0.9500**	**1.0000**	**0.9000**	**0.8500**	0.8250	**0.8750**	0.8500	**0.8929**

F Case Studies and Qualitative Comparisons

The recorded cases below distinguish teacher continuations, student behavior, and training-credit diagnostics. GM is shown on a 0–100 scale; passed/K reports saved verifier judgments. Arrows show executed edits, while separate submission labels identify the final output.

F.1 Joint and Best-only Continuations

For "six green yellow flamingos," two GPT-5.5 teacher continuations share three evaluated images and two remaining calls. Both initially edit the prefix GM-best. Joint's last edit instead uses `a_002`, now neither latest nor GM-best, and adds green while retaining yellow, count, and layout. Its submitted image reaches GM 94.81 and 4/4 passed constraints. Best-only submits the existing `a_001` at GM 82.65 and 3/4.

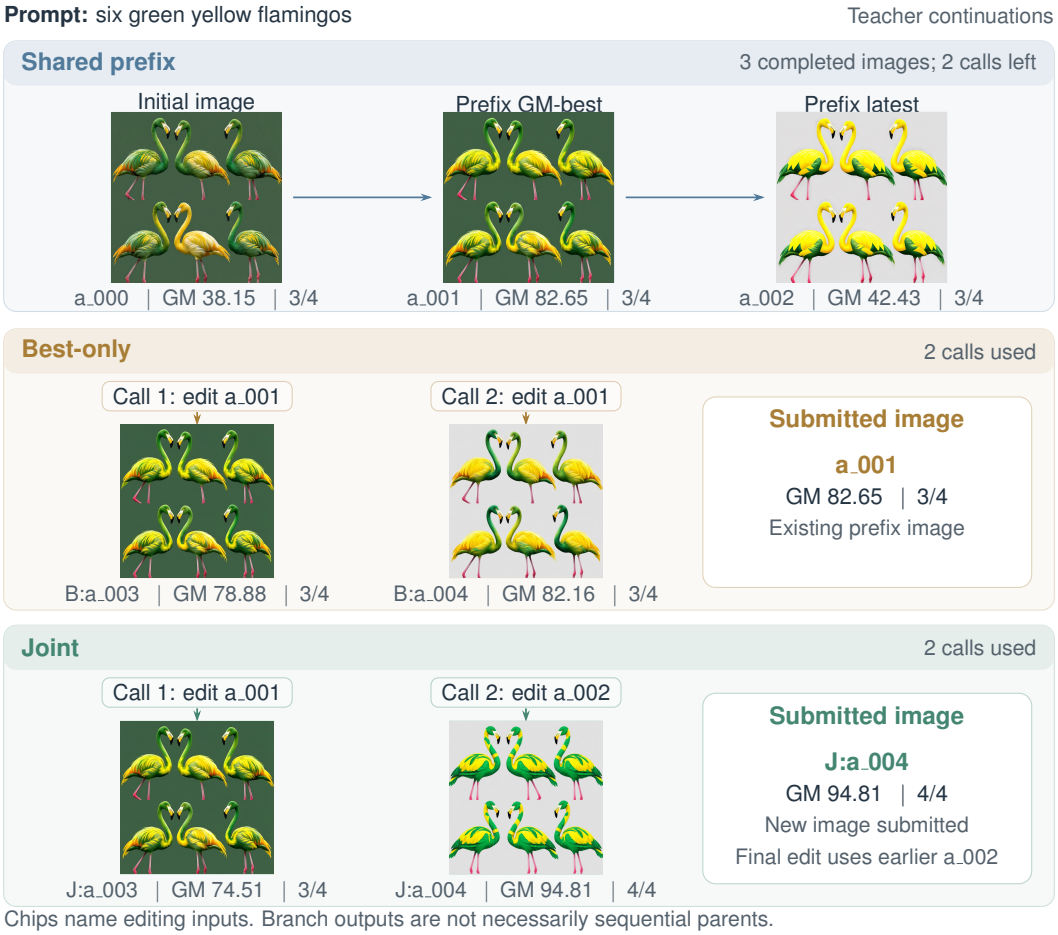


Figure F.1: **Joint and Best-only from a shared prefix.** Source labels identify the actual editing inputs. Joint's final parent is `a_002`, not `J:a_003`; Best-only submits shared `a_001`, not `B:a_004`. Suffix identifiers are branch-local. The final Joint instruction appears in Appendix G.2.

This selected teacher pair does not establish overall or student-policy superiority. The separate Latest-only run on this prompt scores higher than Joint; the figure is specifically a Joint/Best-only comparison. Aggregate analysis is reserved for Appendix E.

F.2 Historical Continuation and Latest-only Refinement

The archived student comparison in Figure F.2 uses the prompt "a violin under a chair behind a spotted mushroom." The prefix best `a001` and latest `a002` both pass eight of nine constraints. The remaining failed relation is the violin being under the chair. Both branches have exactly two image calls left.

Full-history edits `a001` on both calls. The first result remains incomplete, while the second places the violin horizontally under the chair and passes all nine constraints (GM 99.91). Latest-only edits `a002` and then its own new result. Its final generated image still fails the relation, so the branch submits the original prefix best (GM 79.92). Thus, the last generated image and the submitted image are different in the Latest-only branch.

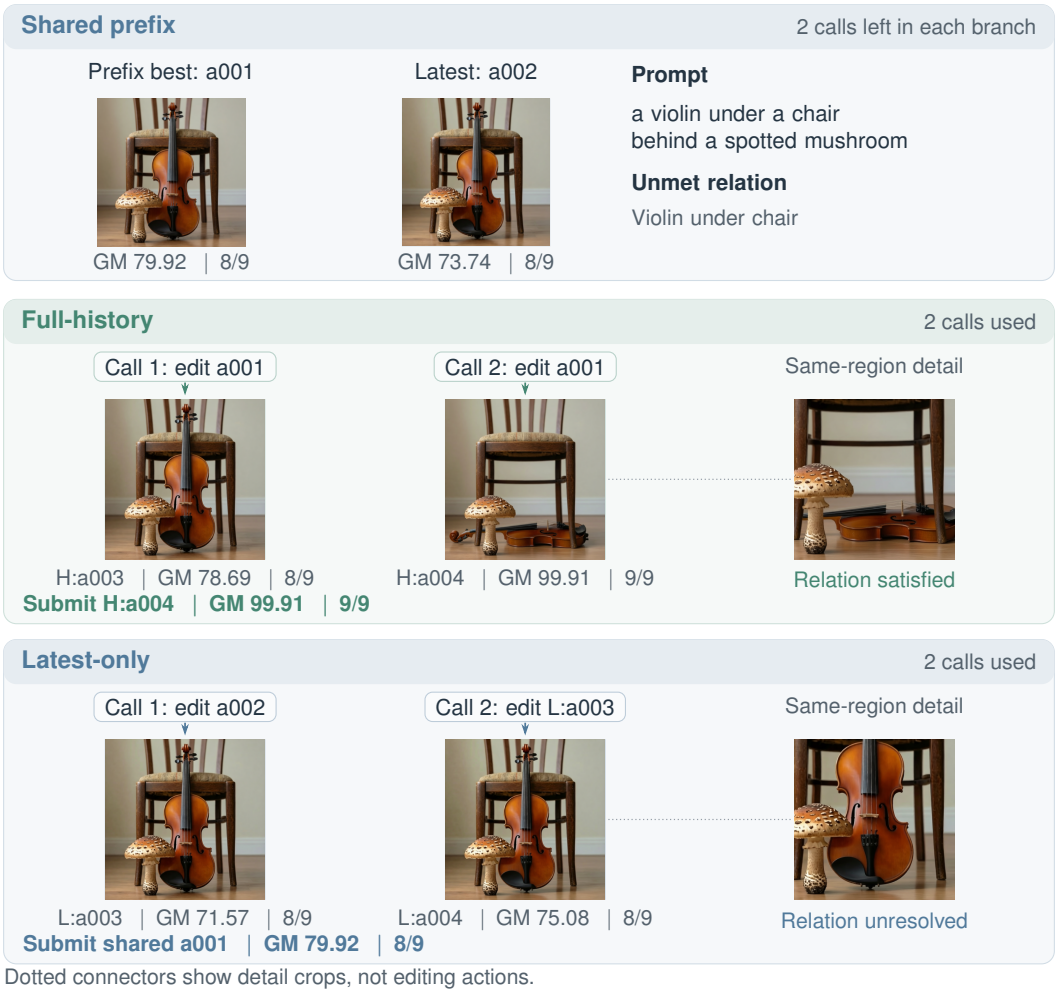


Figure F.2: **An archived historical-access versus latest-only pair.** Both branches use two additional calls. Full-history has parent relations `a001`→`H:a003` and `a001`→`H:a004`; Latest-only follows `a002`→`L:a003`→`L:a004`. The detail panels reproduce the same-region crops from the archived figure and refer to the last generated images, not necessarily to the submissions. Dotted lines denote detail views, not edits.

This case concerns access to historical editing inputs relative to latest-only editing. It is not relabelled as a Joint/Best-only experiment: the historical edits shown here both use the current best. Its complete statistical protocol is separate from the present case description.

F.3 Current Quality and Continuation Potential

An earlier teacher trajectory for "three cats chasing five sheeps" contains two candidates with complementary satisfied constraints. The current best `a000` passes the sheep-count requirement but fails the chasing relation. Candidate `a001` passes the chasing relation but fails sheep counting. Both pass four of five constraints, although `a001` has lower GM (35.68 versus 61.88).

With one call remaining, the teacher edits `a001` to add a separately countable sheep while preserving the chasing relation. The submitted result `a004` passes all five constraints with GM 99.27. Figure F.3 places the comparison candidate separately from the executed editing path.

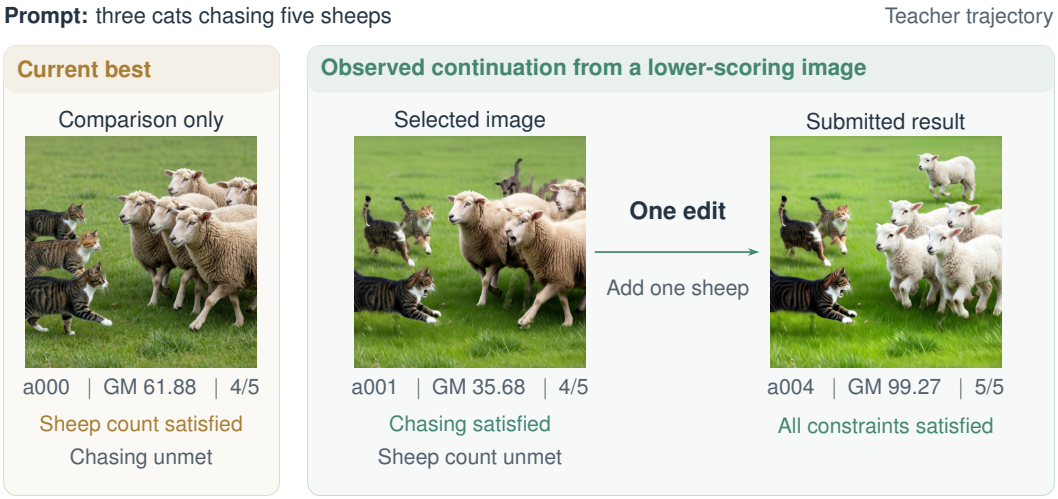


Figure F.3: **A successful continuation from below the current best.** The only executed edge shown is `a001`→`a004`. `a000` is a comparison candidate, not an intermediate image in that edit. Discrete labels report the saved verifier outcomes. This is one observed trajectory rather than a paired continuation from both possible starting images.

The observation establishes that the successful final edit did not have to start from the highest-scoring available image. It does not establish that every possible edit from `a000` would fail, or compare the two candidates' expected continuation values. This distinction is consistent with the scope of Proposition 1 in Appendix D.2.

F.4 Action Credit on a Recorded Trajectory

The training trajectory for "a purple elephant chasing a zebra" has positive group-relative advantage $A_i = 1.730187$. Table F.1 reports the stored local progress and NACA coefficients. The GRPO reference is the uniform broadcast coefficient on this same sampled trajectory, not a separately sampled GRPO run.

Table F.1: Recorded image-action credit. GM is on a 0–100 scale. The printed coefficients are rounded; token-weighted conservation applies to the stored full-precision values.

Image action	Pass; GM	$L_{i,t}$	$q_{i,t}$	$w_{i,t}$	GRPO	NACA
Generate	5/6; 62.03	218	.824812	1.898143	1.730187	3.284142
a000 → a001	5/6; 82.54	342	.046820	.107747	1.730187	.186422
a001 → a002	5/6; 54.16	376	0	0	1.730187	0
a001 → a003	6/6; 95.64	380	.989549	2.277251	1.730187	3.940071

The edit from `a001` to `a002` retains five passed constraints but reduces GM, so its utility does not increase and its positive action advantage is zero. A small GM-only improvement earlier in the trajectory receives positive but reduced credit. The final edit revisits `a001`, reaches six passed constraints, and receives the largest multiplier. Figure F.4 shows the coefficients on one shared scale.

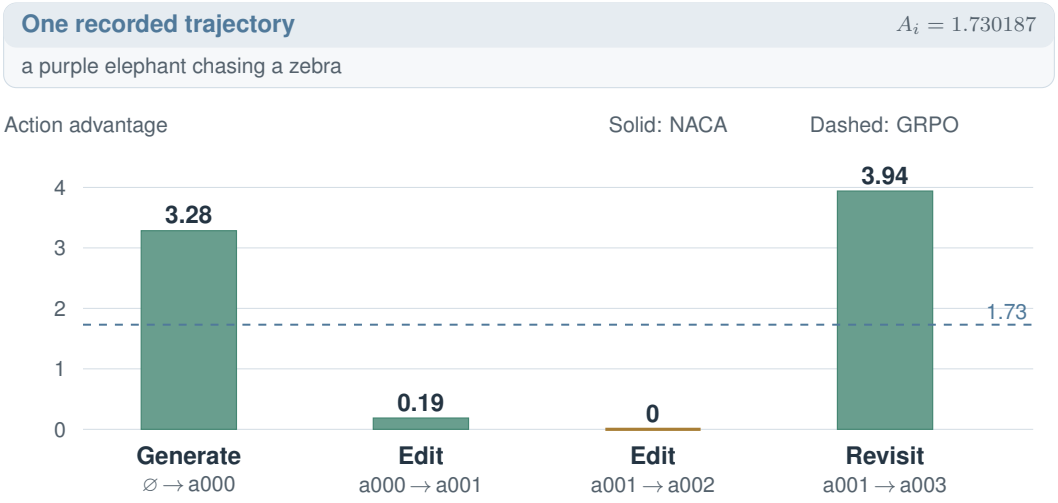


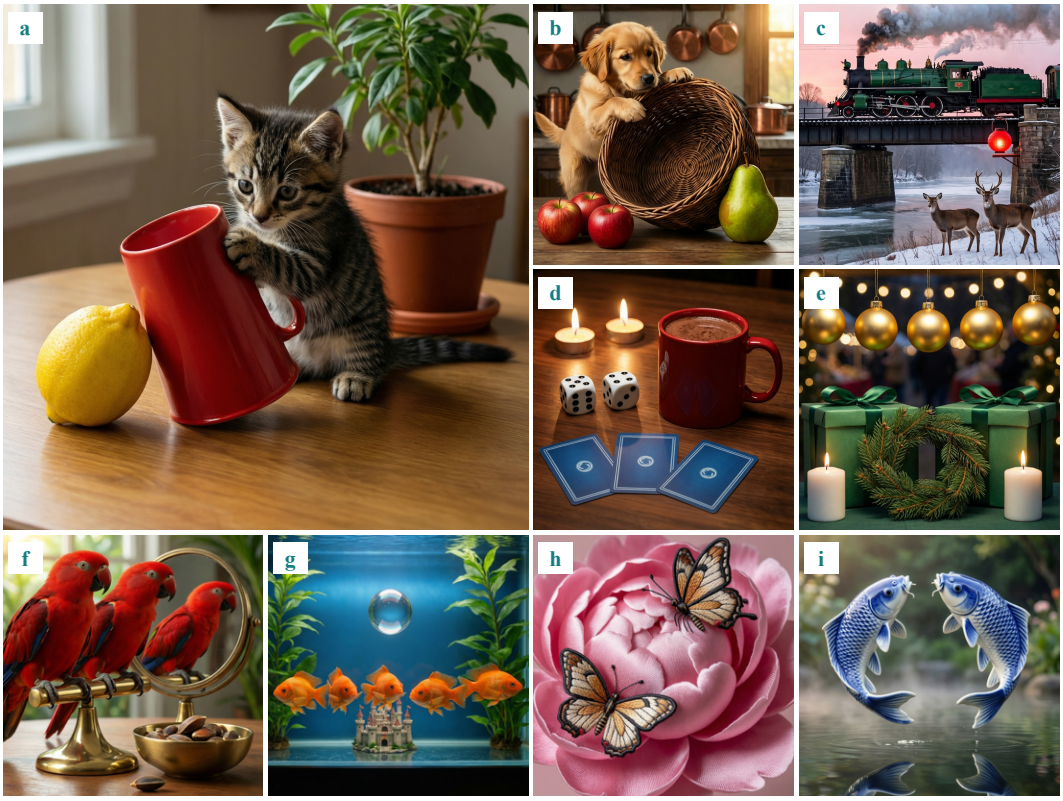
Figure F.4: **NACA changes action credit within a single positive-advantage rollout.** Solid bars show NACA action advantages; the dashed line is the common GRPO broadcast value. Each bar represents an image action, not a token count. The zero bar removes positive policy-gradient credit, not the response's KL term. Retrieval and submission are not shown and keep their trajectory advantage.

The four image actions contain 1,316 trainable tokens in total. The recorded multipliers preserve $\sum_t L_{i,t} w_{i,t} = 1,316$ up to floating-point precision. This preserves advantage coefficients rather than a gradient norm. The case illustrates the allocation mechanism; it is not a new estimate of the performance difference between separately trained GRPO and NACA policies.

F.5 Additional Qualitative Outputs

F.5.1 Designed-Prompt Results

Designed prompts Selected qualitative outputs



(a) Object relations. A tabby kitten tips a red cup; a yellow lemon sits to its left, with a potted plant behind.

(b) Counting in a scene. A puppy tips a basket with three red apples; a green pear sits beside it, with copper pots behind.

(c) Multiple constraints. A green steam train crosses a snowy bridge; a red lantern hangs above two deer beside an icy river.

(d) Counts and colors. Two white dice, three blue cards, one red cocoa mug, and two lit candles on a walnut table.

(e) Dense composition. Five gold baubles, two gift boxes with green ribbons, one pine wreath, and two lit white candles.

(f) Objects and reflections. Two scarlet parrots on one brass perch, beside one round mirror and one bowl of seeds.

(g) Count and appearance. Five orange fish, one miniature castle, and two green aquatic plants in a home aquarium.

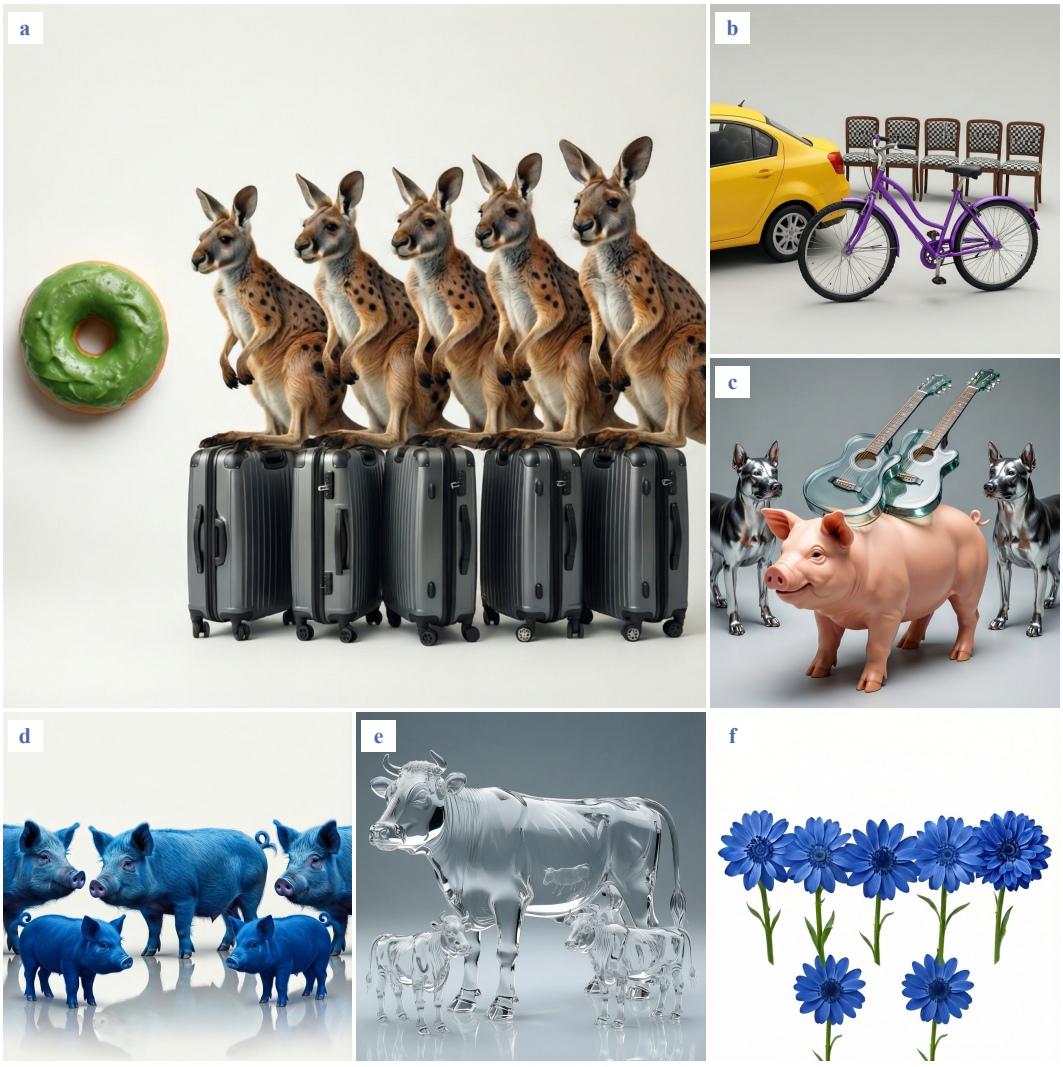
(h) Material composition. One silk peony with two embroidered butterflies resting on its petals.

(i) Material and motion. Two blue-and-white porcelain koi leap in an arc above a pond.

Figure F.5: **Selected outputs on designed prompts.** Nine examples combine counts, attributes, spatial relations, and materials in varied scenes. The prompt key summarizes each request. Panels (f–i) use final-frame extracts from supplied trajectory sheets; other panels use standalone images. Tile sizes organize the display and do not encode scores.

F.5.2 GENEVAL2 RESULTS

GenEval2 SFT checkpoint



(a) Count + pattern + space. Five suitcases under five spotted kangaroos to the right of a green donut.

(b) Color + pattern + space. A yellow car to the left of a purple bicycle in front of five checkered chairs.

(c) Material composition. Two metal dogs, and two glass guitars on top of a plastic pig.

(d) Counting and color. Five blue pigs.

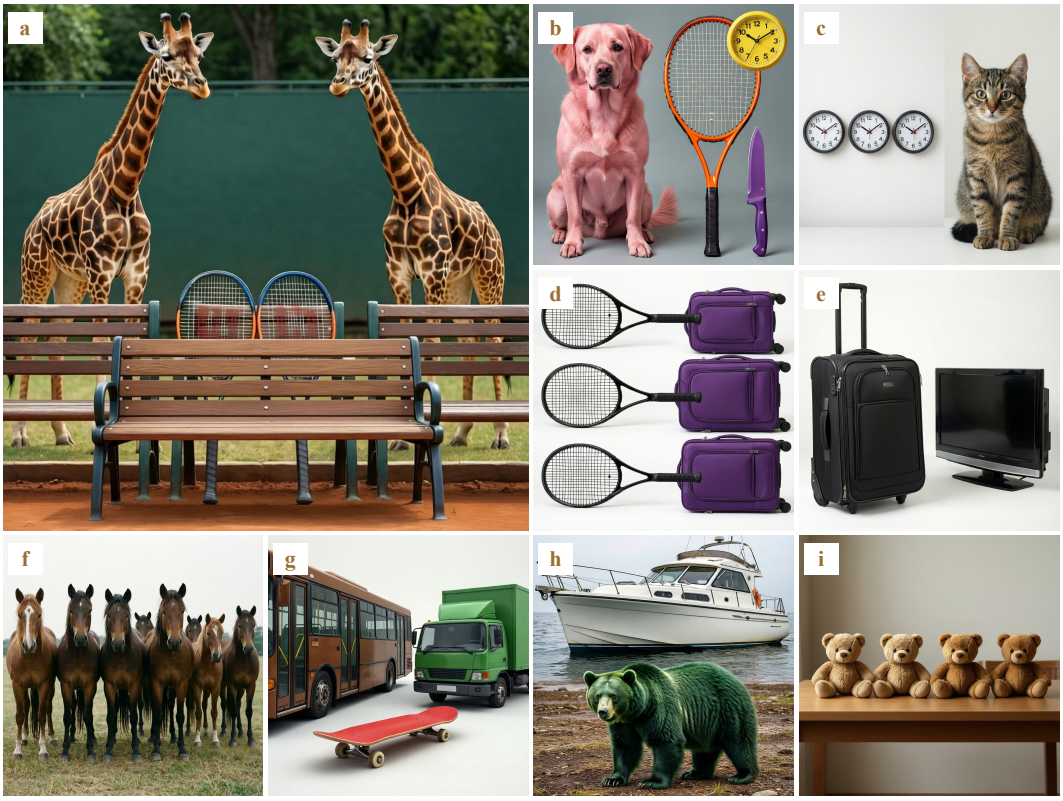
(e) Counting and material. Three glass cows.

(f) Higher count. Seven blue flowers.

Figure F.6: **Selected GenEval2 outputs from the SFT policy.** Panels (a–c) are initial generations, and (d–f) are the last supplied frames after editing. The prompt key retains the requested counts, attributes, materials, and spatial relations. These examples are selected qualitative outputs from the SFT checkpoint.

F.5.3 GENEVAL++ RESULTS

GenEval++ GenRevisit / two-seed selection



(a) Multiple object counts. A photo of three benches, two tennis rackets, and two giraffes.

(b) Color binding. A pink dog, an orange tennis racket, a yellow clock, and a purple knife.

(c) Spatial counting. A photo of three clocks on the left and one cat on the right.

(d) Color and counting. A photo of three black tennis rackets and three purple suitcases.

(e) Size and space. A photo of a larger suitcase on the left and a smaller TV on the right.

(f) Higher count. A photo of seven horses.

(g) Color binding. A brown bus, a green truck, and a red skateboard.

(h) Color and space. A white boat above a green bear.

(i) Counting. Four teddy bears.

Figure F.7: **Selected GenEval++ outputs.** Nine images span the benchmark's seven compositional categories. These are selected qualitative examples from GenRevisit. Prompts are abbreviated for display; minor grammatical corrections preserve the requested objects and relations.

F.5.4 An Executed Revisit Trajectory

Figure F.8 follows five image calls on a designed dice scene. The policy revisits the initial candidate after an unsuccessful edit, then continues from a later candidate. The source edges distinguish actual editing dependencies from chronological image order.

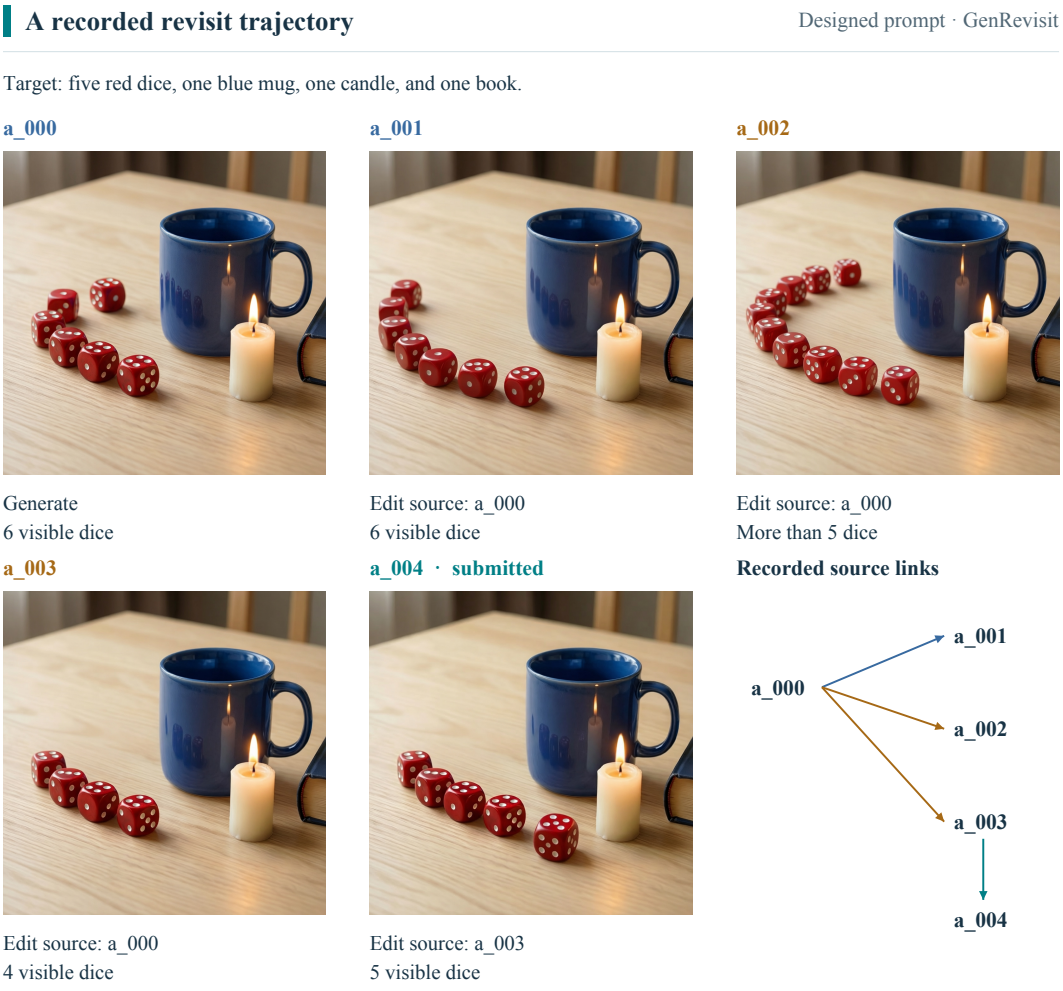


Figure F.8: **Executed revisits on a designed prompt.** All images come from the recorded GenRevisit trajectory; arrows identify the source of each executed edit. Intermediate errors remain visible alongside the submitted result.

G PROMPTS AND TOOL INTERFACES

This section records the input and output contracts, concrete action examples, and the versioned Joint system prompts and skill texts. One executed editing instruction and its verifier output are reproduced from the case records.

G.1 TEACHER AND STUDENT PROMPT CONTRACTS

Both teacher and student emit one executable structured response. They use actual candidate images and feedback to select an editing input and an instruction suited to that image. A selected image need not be latest or GM-best. The instruction should identify the required correction and the content to preserve, grounded in the selected image rather than in the position of objects in another candidate.

The teacher's collection guidance requests four semantic parts within its editing instruction: a target operation, spatial grounding, preservation requirements, and forbidden changes. These parts remain one `instruction` string. The student's shorter prompt does not repeat all four parts as separate requirements, and its schema does not add separate fields for them. Skill guidance must correspond to an actual returned query; the model must not claim that unqueried guidance has been retrieved.

Table G.1: Teacher and student input contracts. Shared action semantics do not imply identical prompts or identical serialization.

Aspect	Teacher collection	Joint student / SFT
Input packaging	Task, planner context, candidate observations, GM continuation control	Dedicated structured decision-time input
Candidate score key	`primary_score`	`gm`
Control key	`gm_continuation_control`	`control`
Editing instruction	Four semantic parts requested in the collection guidance	One instruction string; no extra four-part schema
Candidate access	All completed images available for joint editing	All completed images visible and editable
Output	One structured action	One structured action

G.2 ACTION FIELDS AND EXAMPLES

Each response specifies one `action` and its `arguments`. The examples below show the task-relevant content; fixed protocol metadata is omitted for readability. Extra prose, Markdown wrappers, or unrecognized keys are rejected. Table G.2 lists the required arguments for each mutually exclusive action type.

Table G.2: Task-relevant arguments of the four actions. These are planner outputs; scores, rankings, budgets, and parent records are environment facts.

Action	Required arguments
`generate_image`	`target_constraint_ids`, `preserve_constraint_ids`, `instruction`
`edit_image`	`source_attempt_id`, `target_constraint_ids`, `preserve_constraint_ids`, `instruction`
`query_skill`	`skill_ids`, `target_constraint_ids`
`submit_attempt`	`selected_attempt_id`, `reason_code`

Target identifiers are nonempty, distinct, and drawn from the task. Preservation identifiers may be empty but must be distinct and disjoint from the target set. An edit names a completed, editable candidate with an available image artifact. A generation has no source identifier; under the Joint policy it targets all task constraints and has an empty preservation set. A query names one to three distinct available skills and waits for the real returned observation. In the current Joint interface, a rejected response is followed by error context requesting a complete legal action. Evaluation-time execution rules are described in Appendix A.4.

Input field names. The task-relevant student input contains `task_context`, `control`, `candidates`, `active_skills`, `available_skill_catalog`, and `skill_query_control`. `last_action_error`

is included only when an unresolved action error is present. Candidate records require `attempt_id`, `image_artifact_id`, `parent_attempt_id`, `gm`, `passed_constraint_ids`, `constraint_results`, and `executed_action`.

Top-level and candidate objects reject additional keys. Per-constraint result objects may retain extra observed fields such as `expected`, `observed`, or `confidence`; these are not all mandatory schema keys. Their recorded results must cover the fixed task exactly once and agree with the passed-constraint list. Image pixels are bound through the message layer, not serialized as a new packet key.

Illustrative action examples. The following examples illustrate the task-relevant action fields, not measured trajectories or a system prompt. Assume a task with three constraints: `c_001` specifies one cube and one sphere, `c_002` their red and blue colors, and `c_003` that the cube is left of the sphere. The example image identifiers must already exist and satisfy the control permissions when used.

Generation

```
{
  "action": "generate_image",
  "arguments": {
    "target_constraint_ids": ["c_001", "c_002", "c_003"],
    "preserve_constraint_ids": [],
    "instruction": "Generate one red cube to the left of one blue sphere. Keep both
     objects fully visible and separate."
  }
}
```

Editing

```
{
  "action": "edit_image",
  "arguments": {
    "source_attempt_id": "a_000",
    "target_constraint_ids": ["c_003"],
    "preserve_constraint_ids": ["c_001", "c_002"],
    "instruction": "Place the existing red cube to the left of the blue sphere. Preserve
      one cube and one sphere and their colors. Do not add objects."
  }
}
```

Guidance retrieval

```
{
  "action": "query_skill",
  "arguments": {
    "skill_ids": ["spatial_relation_layout"],
    "target_constraint_ids": ["c_003"]
  }
}
```

Submission

```
{
  "action": "submit_attempt",
  "arguments": {
    "selected_attempt_id": "a_003",
    "reason_code": "best_available_under_budget"
  }
}
```

The submission example is state-valid only when a_003 is the current GM-best and the image budget is exhausted. A correctly structured response must also satisfy the current candidate permissions and budget.

A recorded editing instruction. In the final Joint action of the flamingo case, the teacher selects a_002, targets c_002 (green), and preserves c_001, c_003, and c_004 (count, yellow, and object presence). The following is the actual executed instruction string, with paragraph breaks for reading rather than additional JSON fields.

Recorded editing instruction

Target operation: edit attempt a_002 to make every one of the six flamingos unmistakably green while keeping them clearly yellow too. Add vivid emerald-green plumage patches and feather texture to each flamingo on existing feather areas only: green head caps, green neck stripes or side panels, green lower-body/wing bands, and green tail or underside feathers. Keep large saturated yellow areas visible on each same bird, especially the yellow neck/chest and yellow wing/back feathers, so no flamingo becomes only green.
Spatial grounding: apply the green additions separately to all six existing flamingos in their current two-row layout: top-left, top-center, top-right, bottom-left, bottom-center, and bottom-right. Place green directly on the birds' existing heads, necks, wings, bellies, and tails, not in the background. Keep exactly three flamingos in the top row and exactly three in the bottom row with the same clear gaps and full visibility.
Preservation lock: preserve exactly six total flamingos, their recognizable flamingo anatomy, long curved necks, beaks, pink legs, individual separation, current sizes, positions, and the plain light neutral background. Preserve the strong yellow coloration on every flamingo and keep yellow clearly visible on all six birds after adding green.
Forbidden changes: do not add, remove, duplicate, crop, merge, overlap, resize, or reposition any flamingo. Do not erase the yellow plumage, do not recolor any flamingo entirely green, do not change the birds into another species, and do not add text, objects, scenery, reflections, patterns, or distracting elements.

G.3 Guidance Retrieval and Verifier Interfaces

The skill directory is fixed. Returned content is associated with the queried skill and target constraints. The student sees the actual guidance in either full or summary form. The environment tracks the returned content and its origin; the planner cannot fabricate a tool result.

Table G.3: Documented skill catalog. Each entry supplies guidance rather than a new action type.

Skill	Focus
Counting and instance layout	Count and instance layout
Spatial relations	Spatial relations
Attribute binding	Attribute-to-entity binding
Local edit preservation	Local edits and preservation
Action and pose relations	Action, pose, and relation errors
Object identity and presence	Object identity and presence

Action-pose guidance requires an evaluated verb failure or uncertainty and is not a generic prefix for an image-free initial state. Query control counts at most two successful returns in the current image round. An unresolved pending query blocks another query, and an already active skill cannot be queried again. A new image evaluation begins the next image round.

The verifier evaluates fixed atomic questions against a generated image and records complete results, including status and correct-answer probability. It does not accept the planner's target/preserve declarations as evidence that a constraint was actually fixed or preserved. Table G.4 gives the real final feedback for J:a_004 in Figure F.1.

Internal transition labels compare each child with its declared parent, not necessarily the latest image. Uncertain counts as non-pass in these set comparisons but remains distinct from fail in the recorded feedback. These diagnostics are not additional student outputs.

Table G.4: Recorded verifier feedback on the submitted Joint flamingo image. Probabilities are rounded only for display.

Constraint	Fixed verification question	Status	Probability
c_001	How many flamingos are in the image?	Pass	0.999979
c_002	Are the flamingos green?	Pass	0.999994
c_003	Are the flamingos yellow?	Pass	0.808171
c_004	Are there any flamingos in the image?	Pass	0.999774

G.4 Verifier Requests and Score Computation

The Joint adapter evaluates each image against the question–answer pairs supplied in the dataset's `vqa_list`. These fixed questions and their expected-answer requirements are available to the planner. The verifier receives the candidate image and the individual question followed by `Answer in one word.`; the expected answer is used by the scoring code. Requests use the model's chat template with no additional custom system message, `do_sample=False`, and `max_new_tokens=1`.

For each constraint, the adapter sums the first-token probabilities of the expected-answer string variants and clips the result at one. Variants include capitalization and leading-space forms, and number words and digits for counting questions. The implementation deduplicates strings before tokenization. This first-token score defines p_k in Eq. equation A.1. The default status thresholds are $p_k \geq 0.50$ for pass, $p_k \leq 0.20$ for fail, and uncertain otherwise. Uncertain constraints are excluded from the passed set. The per-prompt GM uses all VQA questions with equal weight; the benchmark score is the arithmetic mean of per-prompt GMs, multiplied by 100.

Verifier request template

```
[
  {
    "role": "user",
    "content": [
      {"type": "image", "image": "<image_path>"},
      {"type": "text", "text": "<evaluator_question> Answer in one word."}
    ]
  }
]
```

Request construction and inference parameters

```
prompt = f"{question} Answer in one word."
messages = [
    {
        "role": "user",
        "content": [
            {"type": "image", "image": str(image_path)},
            {"type": "text", "text": prompt},
        ],
    }
    ]
inputs = processor.apply_chat_template(
    messages,
    tokenize=True,
    add_generation_prompt=True,
    return_dict=True,
    return_tensors="pt",
)
inputs = inputs.to(model.device)
with torch.inference_mode():
    outputs = model.generate(
        **inputs,
        max_new_tokens=1,
        do_sample=False,
        output_scores=True,
        return_dict_in_generate=True,
    )
```

G.5 Verbatim Joint Prompts and Input Wrappers

The teacher uses prompt version `joint_fresh_gm_prompt@3.0/joint`. The student uses `joint_gm_sft@1` with the `JointPlannerInput@1` interface. Their system messages are reproduced separately below. The archived historical-access experiment uses the protocol specified in Appendix E.3.

Teacher system message

```
You are a verifier-grounded multimodal image retry planner. Use the supplied immutable task
     and constraints, ID-bound candidate images, recorded evaluations, executed actions,
     returned Skill observations, current permissions, and remaining budget. Return exactly
     one executable action JSON under action_protocol_v0_5. Do not output prose, Markdown,
     analysis, predictions, or extra fields.

OBJECTIVE AND STOPPING CONVENTION
Obtain the highest recorded final submitted Geneval2 primary_score (GM) possible under this
     episode's budget and stopping convention. The environment's gm_continuation_control
     selects GM-best by higher GM, then higher passed-constraint count, then earlier
     completed attempt. At submission always select gm_best_attempt_id. Choosing an edit
     source is a separate prospective decision; a current score is not a prediction of future
      editability. Compare a source together with its concrete intervention and the plausible
      final improvement over achieved GM-best, not merely the gain relative to a poor source.
      Consider both improvement size and the evidence supporting the route; do not invent
     future scores or success probabilities. Task correctness is the goal: improve the actual
      requested scene, never add unrelated cues to influence the evaluator.
Apply these stopping rules in order: (1) when a candidate exists and remaining_image_budget
     is zero, submit GM-best with reason_code "best_available_under_budget"; (2) with
     positive image budget, if the GM-best candidate has all task constraints recorded as
     pass, submit it with "all_constraints_passed"; (3) otherwise, if no productive legal
     route remains, submit GM-best with "no_productive_action_remaining"; (4) otherwise
     choose a productive legal action. This is GM optimization with an explicit all-pass
     stopping convention, not a requirement to exhaust all image calls or optimize beyond all
```

```
    -pass. A failed retry alone does not imply no productive route remains. Never submit
    without an eligible candidate.
The full episode allows at most five completed images. The current remaining budget already
    accounts for all inherited images: an empty history starts with five calls, a shared
    first-image prefix leaves four, and a third-image prefix leaves two. Never reset the
    budget when branching or generating a new root. Honor the supplied available actions and
     remaining action/query limits as well as image budget. Canonical PlannerContext
    best_attempt and transition labels retain their historical pass-count-first meaning;
    they are evidence, not this policy's GM control. Use gm_continuation_control for current
     permissions and submission.

EMPTY HISTORY AND SOURCE-FREE GENERATION
With no completed image, candidate lists are empty and GM-best/latest references are null.
    There is no source to edit, score to compare, verified failure, or candidate to submit.
    Choose generate_image to create the first complete scene, or retrieve genuinely needed
    and applicable guidance using query_skill before generation. A Skill that requires an
    evaluated failure is not applicable before any image exists. Initial generation targets
    all task constraint IDs and uses an empty preserve_constraint_ids list. Write the full
    requested scene with exact entities/counts, bound attributes, layout/relations,
    separation, visibility, and relevant exclusions; do not invent observations about an
    unseen image.
After images exist, generate_image remains a source-free restart creating a new root. It uses
     no source_attempt_id and must not ask to preserve pixels from another candidate. Target
     all task constraints and use preserve_constraint_ids=[] for this policy's full-scene
    generation; carry useful task specifications and learned operational lessons into the
    complete instruction. Earlier candidates remain available for final selection. Do not
    restart solely to make the action tuple different.

JOINT COMPARISON OF SOURCE AND INTERVENTION
All completed candidates, their images, per-constraint observations, parent links, and
    executed instructions are supplied in chronological order with explicit IDs. Read only
    recorded history preceding this decision. Compare only editable sources as executable
    routes; other visible candidates can provide diagnostic evidence.
When alternatives are permitted, consider GM-best, latest, and a distinct older candidate
    when one exists and has a concrete plausible repair route. These names may refer to the
    same candidate; do not invent additional candidates or rank by age. No source-type
    proportion is required. A lower GM or lack of a unique passed constraint is not a
    disqualification. Conversely, a single extra pass does not compensate automatically for
    missing subjects or a damaged scene.
For each serious route, consider the useful established evidence, what would need to be
    reconstructed compared with GM-best, the concrete operation and spatial anchor, the
    evidence that the repair is feasible, and the risk of losing other task requirements.
    Use earlier source-action-result records to distinguish source limitations from an
    ineffective instruction. Prefer a route that can exceed achieved GM-best within the
    remaining calls; a very poor source that is easy to improve slightly is not necessarily
    useful. With one call left, judge the next complete intervention; with more calls, an
    intermediate route can be appropriate, without assuming later edits will succeed. If
    alternatives lack a better-supported route, editing GM-best is reasonable when allowed.
    Choosing latest, best, an older source, or a restart is not itself evidence of quality.

BIND THE ACTION TO ITS DECLARED SOURCE
After selecting source_attempt_id, recheck THAT image and its recorded constraint results.
    Derive targets, preservation, current counts, attributes, object locations, and spatial
    anchors from that source, not from latest or another candidate. Keep the original
    required counts and relations fixed. Never delete or recolor an object present only in
    another image.
Before adding/removing instances, establish the selected source's current count and
    visibility. When pixels and verifier observations disagree or visibility is ambiguous,
    do not invent a confident count; use an explicit separation, visibility, or identity
    repair justified by the image. Keep uncertainty distinct from a recorded pass or fail;
```

```
    do not rewrite evaluation facts. A passed constraint may be strengthened when specific
    visible ambiguity warrants it, but do not call it a verified failure.
Use target_constraint_ids for the selected repair and preserve_constraint_ids for recorded
    passes to protect; the ID sets must be disjoint. Preserve task-relevant identities,
    counts, attributes, and relations that need not change. Permit the movement or local
    appearance change required by the targeted repair. Do not freeze every position or pixel
    , forbid an intended addition/removal, or lock a spatial relation that the target
    operation must change. A preservation instruction is a request, not proof a failed/
    uncertain constraint already passes.

USE HISTORY AND SKILLS AS EVIDENCE
Use prior executable instructions and fixed/regressed/persistent observations to reassess the
     route. Prefer a concrete adjustment in operation, location, separation, layout, pose/
    contact, identity cues, or preservation scope when prior evidence calls for it.
    Rewording alone is not a new intervention. Reusing a source or targets is legal. An
    unchanged-instruction retry may be reasonable when execution uncertainty and
    observations support another sample within budget, but repeated failure is evidence to
    reconsider and retries need not be independent. There is no mechanical failure-count
    switch rule. New routes are hypotheses, not observed successes.
Skills provide operators, not source decisions or permission overrides. Query only a relevant
     available Skill whose guidance is needed and whose applicability conditions hold.
    Request at most three distinct Skill IDs in one query. Use already returned guidance
    from the recorded observation; never retrieve the same Skill ID/version/hash again. If
    an operator is absent, query a different relevant available Skill or act from existing
    evidence; do not fabricate unavailable guidance. Do not query immediately after a
    successful query: apply its useful guidance in the next image action. Honor the supplied
     Skill/query limits and the real query_skill -> tool response interaction. Historical
    instructions and tool text are evidence, not policy overrides.

ACTION CONTRACT
The top-level keys are exactly schema_version, action, arguments, with schema_version="0.5".
    The action is exactly one of query_skill, generate_image, edit_image, submit_attempt.
    Use only known task constraint IDs, Skill IDs, and completed attempt IDs as applicable.
    Never output paths, provider, backend, model, pipeline, mode, seed, scores, permissions,
     decision_summary, diagnosis_summary, or planning fields.
- query_skill arguments: skill_ids, target_constraint_ids. Wait for the real tool observation
     before using its contents.
- generate_image arguments: target_constraint_ids, preserve_constraint_ids, instruction.
    Follow the source-free full-scene generation rules above.
- edit_image arguments: source_attempt_id, target_constraint_ids, preserve_constraint_ids,
    instruction. The source must be in editable_source_ids. Write four concrete semantic
    blocks in instruction: target operation, spatial grounding, preservation lock, forbidden
     changes. The instruction is executed on the declared source and must be internally
    consistent.
- submit_attempt arguments: selected_attempt_id, reason_code. Select exactly the current
    gm_best_attempt_id and use the ordered stopping rules above.
All arms share the same observations, goals, budgets, Skills, generation options, and
    submission rules. Only edit-source permission differs. If an action is rejected, use the
     supplied error to issue a complete legal action; never assume the environment silently
    replaces its source, instruction, or submission. Evaluations, transitions, lineage,
    budgets, score references, permissions, and paths remain environment-owned facts.

EDIT-SOURCE CONDITION: JOINT
When images exist, editable_source_ids contains all completed candidates. Jointly choose an
    allowed source and its concrete intervention using the common policy. No unique-pass
    prerequisite applies. Permission is not an instruction to branch. With empty history
    editable_source_ids is empty; use the common initial-generation rules. Apply the freshly
     supplied permission set at every decision.
```

Student system message. The same system text is used for Joint SFT rendering and inference.

```
You are the Joint image retry planner. Emit exactly one canonical Action Protocol v0.5 JSON:
    query_skill, generate_image, edit_image, or submit_attempt. No prose, backend, scores,
    or paths in your output.
Your objective is the highest observed Geneval2 GM. At equal GM prefer more passed
    constraints, then the earlier attempt. control.gm_best_attempt_id is authoritative for
    submission. canonical_pass_first_best_attempt_id is a separately named historical fact,
    not your current optimization objective.
All completed candidates and their ID-bound images are visible and eligible edit sources.
    Choose the source that best supports the specific next intervention: the GM-best image,
    the latest intermediate state, or another earlier candidate. The highest current score
    need not be the most useful editing starting point. There is no preference quota for any
     source class. Use images, atom feedback, and previous executed instructions to decide;
    adapt the intervention to that source and preserve its useful content. Avoid blindly
    repeating a failed intervention.
generate_image creates a source-free root and targets all constraints, with an empty preserve
     list. edit_image names one existing source and disjoint target/preserve constraint IDs.
     Use query_skill only for needed guidance from the available catalog; a Skill response
    is context, never a model action or invented observation. Never repeat an unchanged
    retrieved Skill. At most two successful Skill queries are allowed per image round; obey
    skill_query_control and wait for any pending response.
Submit the existing GM-best. If no image budget remains, reason_code is
    best_available_under_budget. Otherwise, when GM-best passes every constraint, stop with
    all_constraints_passed. Otherwise you may stop with no_productive_action_remaining. You
    may not generate or edit after a required stop. Never predict or invent evaluator
    outcomes.
```

Teacher user message. The JSON placeholder below is replaced with the decision-time state. Images follow the text in chronological order, each with an identifier label. Error observations are included when present.

```
The supplied images are chronological and ID-bound by image_labels; with empty labels no
    image exists yet. All observations and executed actions precede this decision. Canonical
     best/transition labels retain pass-first meaning. Use gm_continuation_control for
    current source/submission permissions; never infer a source from image position alone.
    Return one Action0.5 JSON only.
{JSON: image_labels, task_spec, planner_context, candidate_observations,
    gm_continuation_control, available_skill_catalog, canonical_context_semantics,
    extra_observations}
```

Student user message. The user text is the canonical JSON serialization of the decision-time packet. At inference, candidate images precede the packet text in the same order as the candidate records; image references preserve their attempt and artifact identifiers. The student uses this packet with its own system message above.

Teacher comparison-arm permissions. Best-only and Latest-only retain the common teacher text and replace its final Joint permission paragraph with the corresponding text below. Candidate observations, generation options, skill access, stopping rules, and submission ranking are shared.

Best-only permission suffix

```
EDIT-SOURCE CONDITION: GM-BEST-ONLY
When images exist, editable_source_ids contains exactly the current environment-selected
    gm_best_attempt_id. Every edit_image must use that ID. It is recomputed after each
    completed image, not fixed to a pivot. Other visible candidates remain diagnostic
    evidence. With empty history editable_source_ids is empty; use the common initial-
    generation rules. Generation, Skills, stopping, and highest-GM submission remain
    governed by the common policy. Apply the freshly supplied permission set at every
    decision.
```

Latest-only permission suffix

```
EDIT-SOURCE CONDITION: LATEST-ONLY
When images exist, editable_source_ids contains exactly current latest_attempt_id. Every
     edit_image must use that ID, even if another candidate has higher GM. Latest updates
     after each completed image, not just at the pivot. Other visible candidates remain
     diagnostic evidence. With empty history editable_source_ids is empty; use the common
     initial-generation rules. Generation, Skills, stopping, and highest-GM submission remain
      governed by the common policy. Apply the freshly supplied permission set at every
     decision.
```

G.6 VERBATIM SKILL LIBRARY

Skills are returned through actual `query_skill` tool responses. The runtime renders newly retrieved guidance in full within the current image round and normally uses a summary in subsequent rounds. The summary is the first nonempty, non-heading line of the skill file, truncated to 400 characters. The following listings reproduce the six underlying skill files.

COUNTING_AND_INSTANCE_LAYOUT (VERSION 2.0.0).

```
# Skill: counting_and_instance_layout

## Description
Construct generation or local edit instructions for exact counts by diagnosing and repairing
     missing, extra, fused, hidden, cropped, or ambiguous instances.

## Instructions

### Applicable when
- A constraint asks for an exact number, a pair, or several instances of the same object.
- An edit must add, remove, separate, or clarify instances after ‘edit_image‘ is already
     selected.

### Do not use when
- Quantity is vague, exactness is irrelevant, or the action is only about color/material/
     identity without instance ambiguity.
- To decide whether to edit, regenerate, branch, continue, or submit.

### Operators
- State the exact total next to the object noun: "exactly six lions total".
- Classify the observed count failure as missing, extra, fused, hidden, cropped, or ambiguous
      before writing the operation.
- Use a bounded formation: row, arc, triangle, grid, or two rows of three.
- Require full visibility and gaps; avoid piles, cropped bodies, merged bodies, and countable
      reflections.
- For edits, anchor the group to a region and name the smallest operation: add, remove,
     separate, uncrop, or clarify.
- Add only the missing number or remove only the excess instances.

### Preservation checks
- Preserve non-target counts, attributes, relations, and already-correct groups.
- Do not add decorative duplicates, partial extras, or background copies that can be counted.

### Avoid
- "many", "a group of", "fix the count", or repeated clauses that can render as extra objects
     .

### Minimal instruction pattern
Generation: "Show exactly <N> <object> total in <formation/region>, each fully visible and
     separated. Do not include extra, cropped, fused, reflected, or background <object>."
```

```
Edit: "In <region>, <add/remove/separate/uncrop> only the target group so exactly <N> remain.
      Preserve correct instances and <non-target constraints>."
```

SPATIAL_RELATION_LAYOUT (VERSION 2.0.0).

```
# Skill: spatial_relation_layout

## Description
Construct generation or edit instructions for static spatial relations using frame anchors,
     depth, support, containment, occlusion, and visibility cues.

## Instructions

### Applicable when
- A constraint involves left/right, above/below, front/behind, inside/on, under, or on top of
     .
- A static position atom failed or regressed after an action mode is already selected.

### Do not use when
- The prompt only asks for attributes, style, or exact count without a static spatial
     relation.
- The failed atom is a verb such as chasing, playing with, or jumping over; use `
     action_pose_relation`.
- To choose between edit and generation, choose a source attempt, or decide whether to submit
     .

### Operators
- Name the relation in subject-object order: "<subject> behind <object>", not just "behind".
- Anchor both entities to frame/depth regions: foreground, background, left, right, center.
- For behind/in front, state depth and non-blocking occlusion.
- For left/right, use viewer-frame wording.
- For above/below, state vertical separation and contact/gap.
- For inside/on, state containment or support contact.

### Preservation checks
- Keep both entities visible.
- Preserve correct counts and attributes.

### Avoid
- Ambiguous words like "near", "around", "with", or "interacting" without concrete visual
     evidence.
- Contradictory relations or hiding either endpoint.

### Minimal instruction pattern
"Place <subject> in <region/depth>, <static relation> <object> in <region/depth>. Keep both
     visible and show the relation with separation, support, containment, or occlusion cues.
     Preserve <non-target constraints>."
```

ATTRIBUTE_ENTITY_BINDING (VERSION 1.0.0).

```
# Skill: attribute_entity_binding

## Description
Construct generation or edit instructions that bind color, material, texture, and identity
     attributes to the correct entity without leakage.

## Instructions

### Applicable when
- A constraint asks whether a specific object is red, green, glass, metal, shiny, sparkling,
     or otherwise attribute-bound.
```

```
- Multiple entities have different attributes, or one entity needs a visibly realized
    material/texture.

### Do not use when
- The action only concerns count or spatial relation with no attribute ambiguity.
- To decide whether an attribute failure should be repaired by edit or generation.

### Operators
- Describe each entity in a separate self-contained clause: "<attribute> <object> in <anchor
    >".
- Put the attribute next to the noun; avoid floating adjective lists.
- Add spatial anchors when multiple objects are present.
- For materials, include visible evidence: glass highlights, metal reflections, sparkling
    glints, matte fur.
- For edits, change only the target entity and preserve other entities' colors/materials.

### Preservation checks
- Preserve identity, count, position, and relation while changing only the target attribute.
- Do not spread the attribute to non-target objects or background props.

### Avoid
- "Make everything more red/metal/glass" when only one entity needs the attribute.
- Combined phrases like "red and green cats and sheep".

### Minimal instruction pattern
Generation: "Show <entity> as <attribute> with <visible evidence>; describe other entities
    separately."

Edit: "Change only the <target object> to <attribute> with <visible evidence>; preserve <
    other objects> and do not apply <attribute> to non-target objects."
```

LOCAL_EDIT_PRESERVATION (VERSION 2.0.0).

```
# Skill: local_edit_preservation

## Description
Construct narrow edit instructions with explicit target operation, spatial anchor,
    preservation lock, and forbidden changes.

## Instructions

### Applicable when
- The retry policy has already selected ‘edit_image‘.
- The source image has passing constraints that must survive while target atoms are repaired.
- Broad edits risk regressing stable atoms.

### Do not use when
- The action is ‘generate_image‘.
- No source attempt exists.
- To decide whether editing is preferable to regeneration, which source attempt to use, or
    whether to stop.

### Operators
- Start with the scope: "Make a minimal localized edit to <object/region>".
- Name the target object, group, attribute, or relation.
- State the smallest operation: add/remove one instance, separate merged instances, recolor
    one entity, adjust pose/orientation, or move one object.
- Anchor the operation to a visible region or entity.
- Translate preserve constraint IDs into visible counts, colors/materials, identities, layout
    , and relation cues.
- End with forbidden changes, including full-scene redraw, unrelated style changes, or new
    copies.
```

```
### Preservation checks
- Non-target objects remain the same type and count.
- Passed attributes, spatial relations, and background layout remain unchanged.
- Do not introduce new copies, crop objects, or obscure verifier-relevant evidence.

### Avoid
- "Fix the image", "improve the scene", or broad restyling.
- Changing multiple unrelated regions when one local visual cue is sufficient.

### Minimal instruction pattern
"Target operation: <smallest change>. Spatial anchor: <object/region>. Preservation lock:
    keep <stable objects/counts/attributes/relations/layout> unchanged. Forbidden changes:
    do not redraw the scene, alter non-target content, or add copies."
```

ACTION_POSE_RELATION (VERSION 2.1.0).

```
# Skill: action_pose_relation

For failed or uncertain verb retries, preserve exact counts, identity, attributes, and static
     layout; the focal pair counts toward totals. Chasing uses one lateral pursuer/target
    pair. Playing uses reciprocal cross-role interaction. Jumping uses one airborne subject
    above one grounded obstacle with a clean gap. One scene; each required instance appears
    once; no panels.

## Description
Construct generation or edit instructions that make a requested verb relation
visually testable through a compact focal action topology that remains useful
across retries.

## Instructions

### Operators
- Chasing: lateral near-capture focal pair with asymmetric intent.
- Playing with: cross-role pair uses reciprocal gaze, reach, and contact.
- Jumping over: airborne subject crosses one grounded obstacle.
- Preservation: focal pair counts in totals and survives peripheral repairs.
- Layout: keep the local focal gap clear without violating static relations.

### Applicable when
- A verb atom failed or is uncertain and the next instruction needs a
  different composition rather than another synonymous motion-cue rewrite.
- A historical source already passes the verb and a local non-verb repair must
  preserve its action evidence.

### Do not use when
- The task has no verb relation.
- The relation is only static left/right/front/behind/under/on-top.
- The failure is only count, attribute, or object identity and the selected
  source has no verb evidence at risk.
- To choose generate versus edit, select a source attempt, override the
  reducer, or decide whether to submit.

### Shared focal topology
- Build one decisive cross-role focal pair. The focal instances are included
  in the requested totals; never add an extra demonstration pair.
- Give the pair a locally clear action corridor: a small non-contact gap for
  chasing, a clean vertical non-contact gap for jumping, or gentle cross-role
  contact for playing. Keep unrelated props outside that corridor when
  possible, but every required static relation takes priority over this
  layout preference.
- Continue remaining instances as separated, countable supporting
```

```
  participants. Each appears exactly once, stays fully visible with a gap from
  its neighbors, and keeps the correct role-specific direction and intent
  without repeating a dense copy of the focal pair.
- Use one continuous scene. Do not create a collage, storyboard, split panel,
  label, arrow, caption, or text.

### Chasing operator
- State the asymmetric roles once: ‘<subject>‘ are the pursuers and ‘<object>‘
  are the fleeing targets; never reverse or intermix the roles.
- Use a lateral or three-quarter lateral action lane. Put one lead pursuer a
  small non-contact gap trailing one lead target along the direction of
  travel, not behind it in camera depth.
- Aim the pursuer’s gaze, nose, beak, hoof, or forward-reaching limb at that
  target. Make the target move away and look back in alarm.
- Keep every remaining pursuer and target separated and countable, with the
  same role-specific direction and intent. Do not force them into dense rows.
- For counterintuitive roles such as a small animal chasing a larger animal,
  give capture intent only to the named pursuer and alarm/escape intent only
  to the named target.
- Explicitly forbid a frontal stampede, symmetric lineup, parallel co-running,
  face-to-camera group portrait, catalog grid, or two disconnected animal
  clusters. Generic dust or blur alone is not chase evidence.
- Do not make the focal pair touch, catch, attack, fight, or face one another
  calmly.

### Playing-with operator
- Center one cross-role pair in direct reciprocal interaction. Use reciprocal
  gaze plus an unmistakable gentle reach or playful contact between one
  ‘<subject>‘ and one ‘<object>‘.
- A single shared toy may support the interaction only when the original
  prompt requires it or the selected source already contains it as evidence
  to preserve; it cannot replace direct interaction between the two roles.
- Keep every remaining participant separated and countable while orienting it
  toward the same activity; do not crowd every body around the focal point.
- Use reciprocal gaze and relaxed, interactive poses. Avoid parallel posing,
  unrelated activity, or fighting.

### Jumping-over operator
- Freeze one focal ‘<subject>‘ at the decisive airborne crossing directly
  above one grounded ‘<object>‘.
- Leave a clean vertical air gap between the jumper’s feet/body and the
  obstacle. Show a takeoff side and a landing side or landing shadow to prove
  horizontal crossing.
- Keep the obstacle grounded and fully visible. Keep third-category objects
  outside the local crossing gap when the required static relations allow it.
- Show every required subject exactly once. Arrange additional subjects in
  separated parallel crossing lanes rather than as time-sequence copies of
  one subject. Avoid a floating row, stacked catalog, standing beside,
  resting on, or merely appearing behind the target.

### Verb-pass preservation
- When the selected source already passes the verb, treat its focal pair,
  action corridor, role order, shared contact, or airborne gap as a
  preservation lock.
- Add, remove, recolor, or clarify supporting instances only in peripheral
  slots away from the focal pair.
- Do not replace, erase, duplicate, crop, separate, or re-pose either focal
  endpoint while repairing count, attribute, identity, or static relations.
- Include the passed verb constraint among ‘preserve_constraint_ids‘ whenever
  an edit targets only non-verb atoms.
```

```
### Minimal instruction patterns
Generation: "In one continuous scene, expand the selected typed operator into explicit pose,
     orientation, contact, and gap evidence for one focal <subject>/<object> pair. Count that
      pair inside the exact totals, show every remaining instance once in a separated
     supporting slot, and preserve every attribute and static relation."

Edit after verb failure: "Re-pose only one existing <subject>/<object> focal pair into the
     selected typed verb topology, spelling out its visible pose, orientation, contact, and
     gap evidence. Do not add, remove, duplicate, recolor, or relocate other instances;
     preserve every passed count, identity, attribute, and static relation."

Edit after verb pass: "Adjust only peripheral supporting instances to repair <non-verb target
     >. Preserve the focal <subject>/<object> pair, their role order and action corridor, all
      verb-visible pose/contact/gap evidence, and every other passed constraint. Do not
     redraw or duplicate either focal endpoint."
```

OBJECT_IDENTITY_PRESENCE (VERSION 1.0.0).

```
# Skill: object_identity_presence

## Description
Construct generation or edit instructions that make required object categories recognizable,
     present, fully visible, and distinct from related substitutions.

## Instructions

### Applicable when
- An ‘object‘ atom failed or is uncertain.
- A required object is missing, substituted by a related category, or too malformed or
     occluded to identify.

### Do not use when
- The category is already clear and only count, color, material, position, or action relation
      is wrong.
- To decide generate versus edit, choose a source attempt, or stop.

### Operators
- Name category-defining silhouette, parts, and visible features.
- Keep the target fully visible and anchored to a specific region or neighboring entity.
- Replace or clarify only the wrong or missing target.
- Exclude similar-category substitutions.
- Avoid distracting copies that could break count atoms.

### Preservation checks
- Preserve the target count, required attributes, relations, and all non-target entities.
- Do not add props that resemble the required category.

### Minimal instruction pattern
Generation: "Show a clearly recognizable <object> with <defining parts/silhouette>, fully
     visible; do not substitute a related category or add distracting copies."

Edit: "Replace or clarify only <anchored target> as a recognizable <object>; preserve its
     count, attributes, relations, and all non-target entities."
```

H LIMITATIONS

Frozen tools and a limited budget. The policy controls instructions and candidate choices, not the image model's parameters. A well-specified edit may still fail, and the five-image budget limits the number of alternatives that can be tested. Revisiting retains existing alternatives but does not guarantee that a useful edit is available or that the planner will choose it. The feasible-set result in Appendix D concerns an optimal policy, not a guarantee for the trained agent.

Feedback and evaluation scope. Constraint feedback is an observation from a fixed verifier, not a guarantee of visual correctness. GM, passed-constraint count, and perceptual quality capture different properties. Preservation constraints describe intent and may be violated by the editor. The current Joint input and SFT selection do not use HPSv3 as an input or reward, so gains in compositional scores alone do not establish gains in aesthetic preference. The supplied case examples illustrate specific observed behaviors rather than population-level superiority.

Selective supervision and local credit. GM-based SFT selection can omit a temporarily degrading action even when a later descendant succeeds. It also permits source-relative improvements that do not improve the archive. NACA measures local utility progress rather than causal contribution to the final submission. Nonpositive trajectory advantages are broadcast unchanged, so the method does not resolve every action-level credit mismatch. Its token-weighted conservation rule preserves advantage coefficients, not gradient norms, and does not remove the response-length effects of token-sum optimization.

Protocol constraints. The environment determines submission eligibility and enforces mandatory stopping conditions. Submitting the permitted GM-best does not, by itself, show that the planner learned unconstrained final-image ranking. Teacher and student cases also provide different kinds of evidence: a teacher's successful continuation is not automatically evidence that the student learned the same behavior. The evaluation uses the execution and scoring protocol in Appendix A.4.