ADAPTIVE SELF-CORRECTION FOR MULTI-TURN INTERLEAVED TEXT-IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-correction in interleaved text-image generation requires deciding when to refine an image, fall back to a new attempt, or stop, based on the evolving generation history. We introduce *Adaptive Self-Correction Policy Optimization* (ASPO), an on-policy reinforcement learning framework that learns a variable-length policy over reasoning, correction actions, and image synthesis. Cold-start supervision establishes the correction behaviors; reinforcement learning then jointly optimizes the language-action and image-generation branches using only final-image rewards. Each intermediate image provides context for subsequent decisions. For the rectified-flow image branch, ASPO replays one cached stochastic denoising transition per generated image, yielding a tractable transition-level policy surrogate for optimization from the final trajectory outcome. We also introduce *CorrectionBench*, comprising 350 instances that evaluate recovery from flawed interleaved contexts. ASPO improves overall generation performance on GENIUS and GenEval++ over supervised initialization. On CorrectionBench, it achieves gains of 7.9 points in automated evaluation and 7.53 points in blinded human evaluation over the same baseline. Behavioral analyses and component ablations further support the learned correction behavior and the contribution of image-policy optimization.

1 INTRODUCTION

Text-to-image generation often exposes errors only after an image has been produced. A missing object may call for local refinement, an incorrect visual premise may require a new attempt, and a satisfactory result should be returned before further edits damage it. These choices depend on the current image and previous attempts. Self-correction therefore couples action selection, image generation, and stopping: each generated image both determines what has been achieved and changes the context for the next decision.

Existing work provides foundations for optimizing multimodal generation. UniGRPO jointly optimizes reasoning and image synthesis in a single-round generation setting (Liu et al., 2026), while R3 studies a generate–understand–regenerate routine (Ye et al., 2026). We formulate self-correction as a policy that selects refinement, fallback, or termination from the evolving text–image history. Its decisions determine both the content of subsequent attempts and the length of the correction trajectory (Figure 1).

We introduce *Adaptive Self-Correction Policy Optimization* (ASPO) to train this variable-length interleaved policy. Cold-start supervised fine-tuning (SFT) establishes the correction behaviors and their action semantics. ASPO then samples complete trajectories on policy, allowing the model to learn how to use these behaviors on the intermediate states encountered during generation. A shared final-image advantage trains language-action decisions and image-producing steps across the trajectory. For the rectified-flow image branch, one cached stochastic denoising transition per generated image supplies a tractable policy surrogate. The final outcome thus provides shared feedback for correction selection, visual execution, and subsequent decisions.

To assess recovery explicitly, we introduce *CorrectionBench*, comprising 350 interleaved instances with flawed intermediate states, alongside generation evaluations on GENIUS and GenEval++. The experiments trace the contributions of supervision and RL: the complete correction-action vocabulary outperforms either single-action SFT variant, yet the SFT model invokes fallback on only 2.7% of Fallback Task instances. RL raises this rate from 2.7% to 46.7%, alongside a rise in Fallback

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.

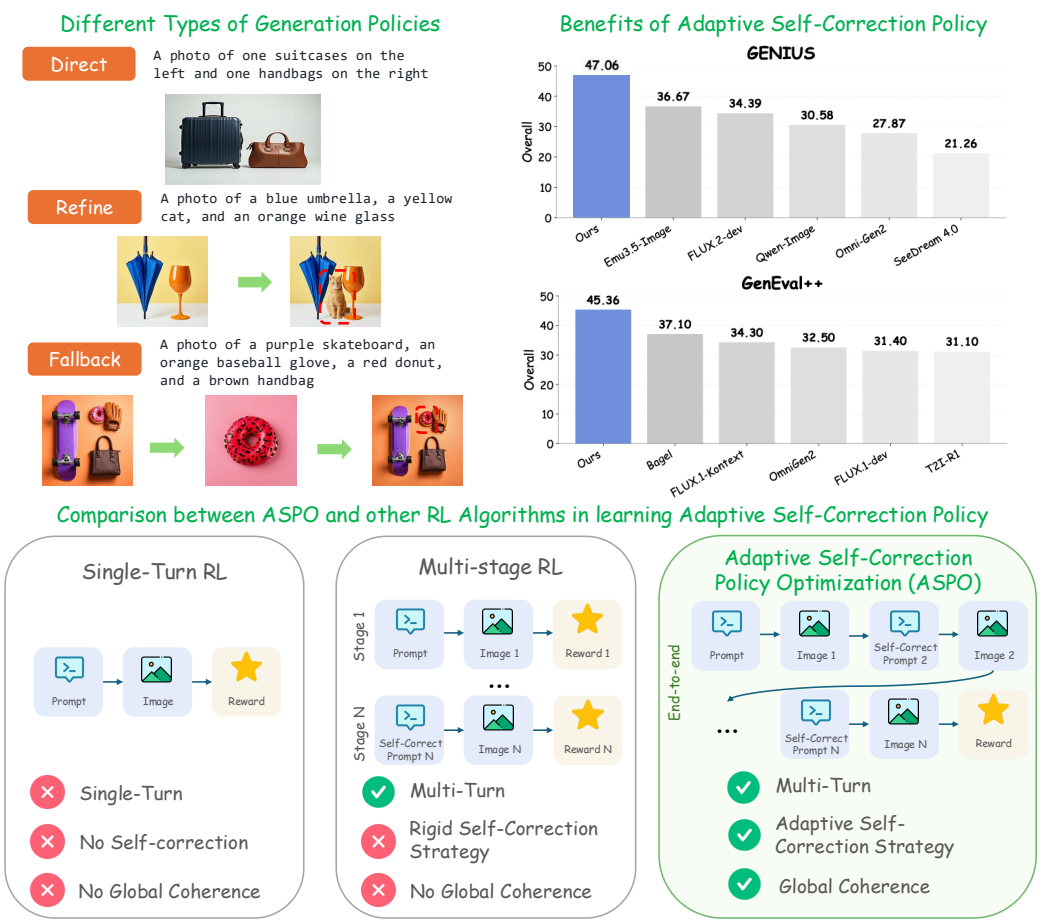

Figure 1: Different prompts require different self-correction strategies: return satisfactory images, refine local errors, or fallback from globally incorrect attempts. The charts show the final ASPO model (Ours) and selected baselines on GENIUS and GenEval++; GenEval++ scores use a 0–100 scale. The lower panel contrasts single-turn, fixed multi-stage, and adaptive trajectory-level optimization.

Follow-Plan score from 66.0 to 79.7. Image-policy and reward ablations further assess the joint training design, while blinded human evaluation corroborates the overall correction gain.

Our contributions are threefold: (1) a formulation of interleaved self-correction in which actions and stopping are selected from the evolving generation history; (2) ASPO, which jointly trains language-action and rectified-flow image policies over complete correction trajectories using stochastic-transition replay; and (3) CorrectionBench, together with complementary action, component, generation, and human evaluations that connect the learned behavior to improved error recovery.

2 Related Work

Unified multimodal understanding and generation. Unified frameworks combine visual understanding and synthesis (Zhao et al., 2025), as illustrated by Janus (Wu et al., 2025b) and Chameleon (Team, 2024). Emu3 (Wang et al., 2024) and Liquid (Wu et al., 2026) explore discrete visual representations, while VILA (Lin et al., 2024) studies visual-language pretraining. BLIP3-o (Chen et al., 2025) and UniWorld-V1 (Lin et al., 2025) use multimodal backbones to condition image decoders, while Show-o (Xie et al., 2025) and Transfusion (Zhou et al., 2024) combine autoregressive and diffusion objectives. BAGEL (Deng et al., 2025) and Mogao (Liao et al., 2025)

scale interleaved training for reasoning and generation. ASPO builds on a unified backbone and trains it to assess intermediate outputs and choose correction actions.

RL for diffusion and flow matching. Reward-based optimization treats sampling as a policy, but deterministic flow samplers lack the log probabilities required by standard policy gradients. FlowGRPO (Liu et al., 2025) and DanceGRPO (Xue et al., 2025) introduce stochastic sampling for group-based optimization; MixGRPO (Li et al., 2025) combines ODE and SDE sampling; and Flow-CPS (Wang & Yu, 2025) addresses timestep-dependent noise inconsistencies. TempFlowGRPO (He et al., 2025) and G2RPO (Zhou et al., 2025) study step-wise reward assignment. T2I-R1 (Jiang et al., 2025) and GoT-R1 (Duan et al., 2025) incorporate reasoning into reward-optimized generation. Our objective spans multiple image attempts and their correction decisions, with feedback assigned from the final returned image.

RL for unified multimodal generation. PromptRL (Wang et al., 2026) jointly trains prompt refinement and flow-based image generation, while ThinkGen uses SepGRPO (Jiao et al., 2025) to alternate reinforcement learning between its MLLM and DiT modules. R3 (Ye et al., 2026) uses a fixed generate–understand–regenerate routine, and UniGRPO (Liu et al., 2026) jointly optimizes single-turn prompt–reasoning–image trajectories. ASPO makes correction structure part of the policy: refine, fallback, and terminal are selected from the evolving history, and both branches are optimized over the resulting variable-length trajectories. Its focus is the formulation and training of sequential correction decisions within a unified generator.

3 METHOD

ASPO connects three elements: a history-conditioned correction policy, supervised initialization of its behaviors, and joint optimization from the final trajectory outcome (Figure 2). A generated image is both a policy output and part of the next decision state. Accordingly, the language-action branch selects a correction or termination, while the image branch executes image-producing actions using the retained history. SFT supplies demonstrations of these behaviors; RL trains their use on trajectories sampled from the current policy.

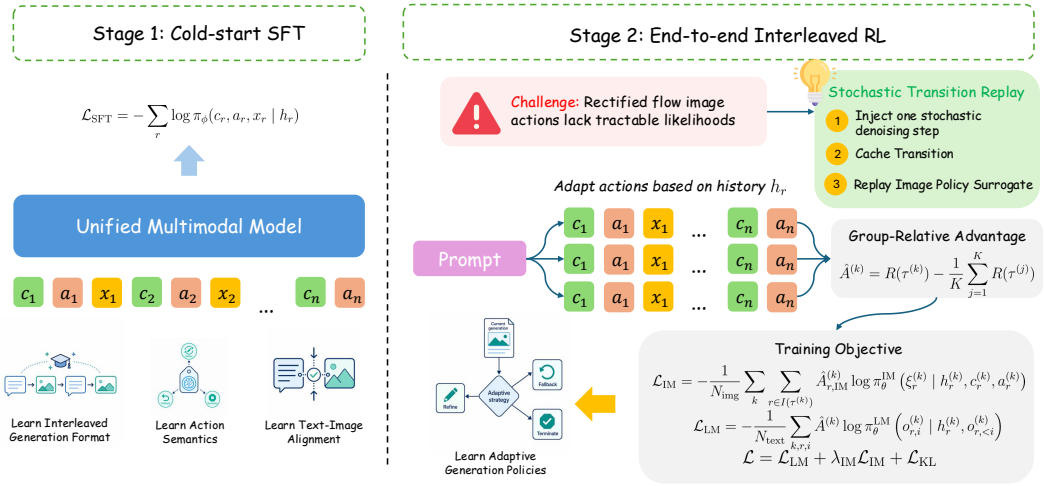

Figure 2: ASPO training. Cold-start SFT teaches interleaved generation and correction actions; erroneous intermediate images are context only, with image loss applied to the final correct target. On-policy RL then optimizes language-action and image-generation branches using final-image reward. Each emitted image contributes a cached stochastic denoising transition that is replayed as an image-policy surrogate.

3.1 MULTI-TURN INTERLEAVED TEXT-IMAGE GENERATION

We study multi-turn interleaved text-image generation, where a unified model alternates between chain-of-thought generation and image synthesis over multiple rounds. Given a prompt s, the model

produces a trajectory of at most R decision rounds, including a terminal round if selected. At round r, the language policy emits a chain-of-thought segment c_r and an action tag a_r.

The first round always creates the initial image and is constrained to

$$a_1 = \texttt{ins}.$$

In later rounds, the model chooses among refinement, fallback, and termination:

$$a_r \in \{\texttt{refine}, \texttt{fallback}, \texttt{terminal}\}, \qquad r \geq 2.$$

The actions `refine` and `fallback` are non-terminal image-emitting actions. They are textual control tokens that condition the same image generator through the full interleaved history; they do not invoke separate generators or discard context. The action `terminal` stops the rollout and returns the most recent image. Let

$$\mathcal{I}(\tau) = \{r \mid a_r \neq \texttt{terminal}\}$$

denote the image-emitting rounds in trajectory τ. Since $a_1 = \texttt{ins}$, every valid trajectory satisfies $1 \in \mathcal{I}(\tau)$. Let

$$h_r = (s, c_{<r}, x_{<r})$$

be the interleaved history before round r, where $x_{<r}$ contains all images emitted in previous non-terminal rounds. For each $r \in \mathcal{I}(\tau)$, the model samples an image x_r, appends it to the history, and conditions subsequent rounds on the updated text-image context. If no terminal action appears within R rounds, the final output is the most recent generated image.

We express the interleaved policy using the following factorization with an image-policy surrogate:

$$\pi_\theta(\tau \mid s) = \prod_{r=1}^{R_\tau} \pi_\theta^{\mathrm{LM}}(c_r, a_r \mid h_r) \prod_{r \in \mathcal{I}(\tau)} \pi_\theta^{\mathrm{IM}}(\xi_r \mid h_r, c_r, a_r), \tag{1}$$

where π_θ^{LM} is the autoregressive language-action policy and π_θ^{IM} is an image-policy density surrogate over the stored stochastic image transition record ξ_r. Thus, intermediate images are both policy outputs and conditioning inputs for later rounds.

3.2 Cold-start Supervised Fine-tuning

We construct 28,842 interleaved demonstrations by first analyzing model failures to distill error patterns, then creating and quality-filtering new generation targets and injecting controlled errors. Existing benchmark failures inform the abstract patterns; their prompts, reasoning, and images are not reused as training instances. Appendix A details the construction, behavior statistics, and exact-string overlap checks against evaluation prompts.

Each demonstration contains reasoning, action tags, image states, and final verification. Erroneous intermediate images receive image-loss mask 0; only the final correct target receives mask 1. Incorrect initial reasoning in wrong-premise fallback cases is also excluded from the text loss. SFT thus establishes correction behaviors without positively supervising erroneous images. During RL, the model must apply these behaviors to its own sampled intermediate states; the demonstrations initialize this process but are not replayed as RL trajectories.

3.3 End-to-end Interleaved Reinforcement Learning

Starting from the SFT model, we jointly optimize initialization, refinement, fallback, and termination over on-policy trajectories. The reward evaluates the image ultimately returned by the complete correction process. Each earlier action and generated image receives a training signal from that final outcome, including when an intermediate attempt is followed by further correction. This shared feedback connects behavior selection and image synthesis across rounds.

For each prompt s, we sample K candidate trajectories,

$$\tau^{(1)}, \ldots, \tau^{(K)} \sim \pi_\theta(\cdot \mid s),$$

and score each using only the final image,

$$R(\tau) = J(x_{\mathrm{final}}(\tau), s), \tag{2}$$

where $x_{\text{final}}(\tau)$ is the image returned at termination, or the last image if the rollout reaches R rounds. The judge J is Qwen3.6-27B, receiving only the original prompt and final image. It assesses prompt compliance for generation/editing and combines RC/VC/AQ scores, normalized to $[0, 1]$, for GENIUS-style prompts. Intermediate images receive no separate reward. Training and evaluation judges are distinguished in Appendix C.

We compute a prompt-wise group-relative advantage,

$$\hat{A}^{(k)} = R(\tau^{(k)}) - \frac{1}{K}\sum_{j=1}^{K} R(\tau^{(j)}). \tag{3}$$

This scalar advantage is assigned to all generated chain-of-thought tokens. Since action tags are generated by the language policy, they receive credit through their token log probabilities. For the image objective, the same trajectory-level advantage is applied only to image-emitting rounds:

$$\hat{A}^{(k)}_{r,\text{IM}} = \mathbf{1}\Big[r \in \mathcal{I}(\tau^{(k)})\Big]\, \hat{A}^{(k)}. \tag{4}$$

Thus, image-emitting rounds receive $\hat{A}^{(k)}$. Terminal decisions retain language-policy supervision but contribute no image-policy term.

The language objective is

$$\mathcal{L}_{\text{LM}} = -\frac{1}{N_{\text{text}}}\sum_{k,r,i} \hat{A}^{(k)} \log \pi_\theta^{\text{LM}}\left(o^{(k)}_{r,i} \mid h^{(k)}_r, o^{(k)}_{r,<i}\right), \tag{5}$$

where $o_{r,i}$ ranges over the chain-of-thought tokens and action tag generated at round r.

To propagate the final-image reward into the image generator, we require a tractable policy term for each image-emitting decision. Since the standard rectified-flow sampler is deterministic given the initial noise and conditioning history, it does not provide score-function log probabilities. We therefore make one denoising step stochastic for each emitted image, yielding an image-side policy event whose log density can be weighted by the trajectory advantage while leaving the rest of the sampler unchanged. Let

$$\xi_r = (z^{(r)}_i, z^{(r)}_{i+1}, t_i, \Delta t_i)$$

be the stored transition record for round r. During rollout, one selected denoising step is sampled as

$$z^{(r)}_{i+1} \sim q_\theta\left(z^{(r)}_{i+1} \mid z^{(r)}_i, h_r, c_r, a_r\right),$$

with

$$q_\theta = \mathcal{N}\left(z^{(r)}_{i+1}; \mu_\theta(z^{(r)}_i, t_i, \Delta t_i, h_r, c_r, a_r), \sigma_i^2 I\right). \tag{6}$$

Here, μ_θ is given by the rectified-flow update, and σ_i^2 is fixed by the sampler schedule.

During training, the stored transition is replayed as a fixed observation, and the current policy recomputes its transition log density:

$$\log \pi_\theta^{\text{IM}}(\xi_r \mid h_r, c_r, a_r) = \frac{1}{|z|}\sum_j \log q_\theta\left(z^{(r,j)}_{i+1} \mid z^{(r,j)}_i, h_r, c_r, a_r\right), \tag{7}$$

omitting Gaussian normalizers independent of θ. This is a transition-level policy surrogate, not the marginal likelihood of the decoded image or full denoising path. We sample the stochastic transition from the high-noise prefix of the denoising schedule because early steps more strongly influence semantic alignment and final image structure. The image objective is

$$\mathcal{L}_{\text{IM}} = -\frac{1}{N_{\text{img}}}\sum_k \sum_{r \in \mathcal{I}(\tau^{(k)})} \hat{A}^{(k)}_{r,\text{IM}} \log \pi_\theta^{\text{IM}}\left(\xi^{(k)}_r \mid h^{(k)}_r, c^{(k)}_r, a^{(k)}_r\right). \tag{8}$$

The overall objective is

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{IM}}\mathcal{L}_{\text{IM}} + \mathcal{L}_{\text{KL}}, \tag{9}$$

where λ_{IM} balances language and image updates, and $\mathcal{L}_{\text{KL}}$ optionally regularizes the language-action policy against the cold-start SFT reference policy. For intuition, suppressing loss normalizations, the

Table 1: GENIUS results across different methods.

Method	Overall	Implicit Pattern			Symbolic Constraint			Visual Constraint			Prior-Conflicting			Multi-Semantic		
		RC	VC	AQ	RC	VC	AQ	RC	VC	AQ	RC	VC	AQ	RC	VC	AQ
GPT-Image	47.15	58.14	41.92	93.60	58.82	32.82	93.79	49.17	62.50	92.50	43.50	33.33	90.00	28.64	–	85.45
Nano Banana	50.66	56.47	39.04	94.12	60.46	51.91	90.20	68.33	79.17	93.33	35.50	39.47	91.00	30.28	–	93.12
SeeDream 4.5	52.84	70.00	59.59	97.06	62.91	41.09	94.37	58.33	62.50	86.67	40.10	41.38	92.57	35.00	–	86.82
Nano Banana Pro	57.19	66.86	44.59	96.51	71.38	50.00	92.11	76.67	66.67	96.67	52.97	41.38	90.59	35.45	–	95.00
Bagel	26.74	26.74	27.03	84.30	29.61	16.03	76.32	22.50	12.50	49.17	22.28	17.24	74.75	33.49	–	53.67
Omni-Gen2	27.87	29.07	26.35	76.16	25.33	30.38	77.96	11.67	41.67	52.50	23.76	34.48	69.80	19.27	–	63.76
Qwen-Image	30.58	36.18	27.69	71.05	36.18	27.69	71.05	26.67	45.83	55.83	27.72	20.69	71.78	25.91	–	69.55
FLUX.2-dev	34.39	34.30	27.70	88.95	35.76	31.01	87.09	39.17	50.00	59.17	25.25	30.17	84.16	29.82	–	79.82
Emu3.5-Image	36.67	41.86	35.81	83.72	34.97	39.31	86.93	24.17	29.17	42.50	26.24	37.93	82.18	32.87	–	75.46
Our SFT	44.32	68.02	43.84	42.44	51.96	32.06	39.87	44.64	18.42	22.32	47.00	34.62	46.50	40.91	–	29.09
Our RL (ASPO)	47.06	70.35	36.30	55.81	55.23	27.10	41.18	63.39	18.18	38.39	58.50	34.62	40.00	44.55	–	35.00

Table 2: GenEval++ results across different methods.

Method	Color	Count	Color/ Count	Color/ Pos	Pos/ Count	Pos/ Size	Multi- Count	Overall
BLIP3-o 4B	0.125	0.225	0.1	0.45	0.125	0.55	0.225	0.257
SD3-medium	0.55	0.5	0.125	0.35	0.175	0.15	0.225	0.296
BLIP3-o 8B	0.25	0.25	0.125	0.6	0.125	0.575	0.225	0.307
T2I-R1	0.675	0.325	0.2	0.35	0.075	0.25	0.3	0.311
FLUX.1-dev	0.35	0.625	0.15	0.275	0.2	0.375	0.225	0.314
OmniGen2	0.55	0.425	0.2	0.275	0.125	0.25	0.45	0.325
FLUX.1-Kontext	0.425	0.5	0.2	0.25	0.3	0.4	0.325	0.343
Bagel	0.325	0.6	0.25	0.325	0.25	0.475	0.375	0.371
Our SFT	0.35	0.325	0.475	0.3	0.275	0.65	0.475	0.4071
Our RL (ASPO)	0.525	0.65	0.4	0.375	0.475	0.5	0.25	0.4536

image weight, and the optional KL term, the shared-advantage structure can be written schematically as

$$\nabla_\theta J \approx \sum_k \hat{A}^{(k)} \nabla_\theta \left[\sum_r \log \pi_\theta^{\mathrm{LM}}(c_r^{(k)}, a_r^{(k)} \mid h_r^{(k)}) + \sum_{r \in \mathcal{I}(\tau^{(k)})} \log \pi_\theta^{\mathrm{IM}}(\xi_r^{(k)} \mid h_r^{(k)}, c_r^{(k)}, a_r^{(k)}) \right]. \tag{10}$$

3.4 CorrectionBench

CorrectionBench evaluates recovery from flawed interleaved contexts. We analyze failures of models including Qwen-Image (Wu et al., 2025a) and BAGEL (Deng et al., 2025), and reasoning patterns represented in GENIUS (An et al., 2026) and WISE (Niu et al., 2025). GPT-4o (Hurst et al., 2024) audits reasoning errors; Gemini-3.1-Pro expands their scenarios; and Gemini-3.0-Flash-Image synthesizes flawed images. Detail artifacts are manually curated from open-source generation outputs.

The benchmark contains 350 instances: 150 logical errors, 150 quality degradations, and 50 detail errors. A separate input classification gives 120 single-image and 230 multi-image correction instances. Each model receives the request, prior text–image history, generation plan, and flawed latest image; construction-time corrective continuations and target reference images are withheld. Refine cases call for local repair, while fallback cases require regeneration. Following RealUnify (Shi et al., 2025), Gemini-3.1-Pro evaluates Follow Plan (FP) and Aesthetic Quality (AQ), weighted 0.9/0.1. Appendix B provides the task definitions, illustrative traces, evaluation prompt, and human validation.

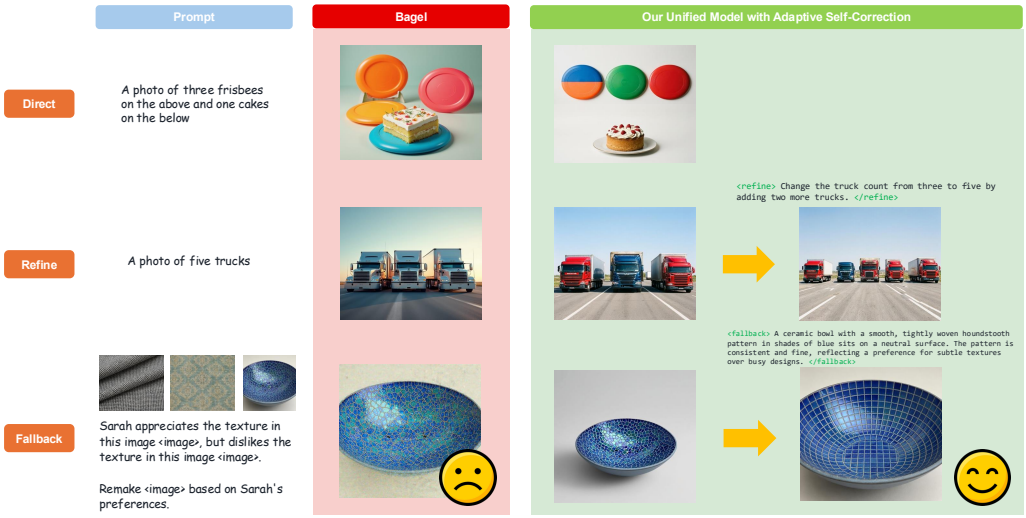

Figure 3: Qualitative comparison with BAGEL. From top to bottom: direct generation, refinement of an object-count error, and fallback after an unsuccessful edit. ASPO conditions each correction on the preceding text–image history.

4 Main Results

Experimental setup. All SFT and ASPO experiments use BAGEL-7B-MoT with its native rectified-flow image branch. We update language-action and image-generation branches while freezing ViT/VAE. SFT uses 28,842 trajectories and 8,000 steps. RL uses 8,871 training and 1,161 validation prompts, $K = 8$, $R = 3$, and 64 prompts (512 trajectories) per optimizer step for 100 steps. Appendix C consolidates hyperparameters and compute; Section 5 and Appendix D report controlled ablations and sensitivity experiments at 32 prompts per step.

Quantitative results. Tables 1 and 2 report generation performance. SFT raises GENIUS from BAGEL's 26.74 to 44.32 and GenEval++ from 0.371 to 0.4071; ASPO further improves these scores to 47.06 and 0.4536. Relative to SFT, Count improves from 0.325 to 0.650 and Pos/Count from 0.275 to 0.475, supporting stronger single-category and position-conditioned counting. Multi-Count, however, decreases from 0.475 to 0.250. Pure color binding remains challenging. On CorrectionBench, ASPO raises the automatic score from 67.9 to 75.8 (+7.9 points). Blinded human scores over all 350 instances likewise rise from 67.61 to 75.14 (+7.53 points; Appendix B.4).

Qualitative results. Figure 3 illustrates direct generation, refinement, and fallback. The examples show improved initial composition, correction of an object-count error through refinement, and recovery when an edit fails to produce the desired visual attribute. They illustrate how different correction actions can address different erroneous states.

5 Ablation Study

We examine how the correction behaviors are established, how RL changes their use, and how image-policy optimization and reward assignment contribute to the final outcome. Tables 3–6 provide complementary evidence linking these stages.

5.1 Role of Cold-Start Supervision

The supervised variants in Table 3 examine terminal verification and the correction-action vocabulary. Structured generation without terminal verification scores 56.7 on CorrectionBench, compared with BAGEL's 49.7. Adding verification in the *ins*-only variant raises the score to 59.4, indicating a benefit from learning to assess and terminate an attempt even before introducing correction actions.

Table 3: Ablation results on *CorrectionBench*. **FP** (Follow Plan) and **AQ** (Aesthetic Quality) are sub-metrics weighted 0.9 and 0.1 respectively. **Overall** is the mean score across both sub-tasks. All values are on a 0–100 scale.

		Refine Task		Fallback Task	
Method	Overall	FP	AQ	FP	AQ
BAGEL (Deng et al., 2025)	49.7	51.3	92.7	39.7	82.3
SFT w/o *terminal*	56.7	55.7	79.0	52.7	79.7
SFT (*ins* only)	59.4	58.3	80.3	56.0	78.3
SFT + *refine*	63.3	64.3	75.0	61.3	60.3
SFT + *fallback*	64.0	61.0	80.7	62.3	89.0
Our SFT	67.9	67.0	76.7	66.0	84.7
Our RL (ASPO)	**75.8**	**69.0**	**86.0**	**79.7**	**91.3**

Adding *refine* or *fallback* supervision increases the overall score to 63.3 and 64.0, respectively. The full SFT model reaches 67.9, exceeding both single-correction-action variants. In this comparison, jointly supervising local refinement, fallback, and verification yields the highest SFT score.

The sub-task scores reveal complementary strengths: refine-only supervision reaches Fallback Task FP/AQ scores of 61.3/60.3, while fallback-only supervision reaches 62.3/89.0. Together with the higher overall score of full SFT, these results motivate teaching verification and both correction behaviors. This initialization makes the behaviors available to the policy; Section 5.2 examines how RL changes their use on the erroneous states presented at evaluation.

5.2 Reinforcement Learning for Adaptive Action Selection

The full SFT model already receives explicit fallback demonstrations (Appendix A), yet it invokes *fallback* on only 2.7% of Fallback Task instances. Table 4 shows that ASPO raises this rate to 46.7%. On Refine Task instances, *refine* use increases from 35.3% to 51.3%. RL thus changes how the model uses behaviors learned during supervised initialization, with the largest shift in fallback use.

Table 4: Action usage rates of Our SFT and ASPO on *CorrectionBench*. *Refine rate* is the fraction of Refine Task samples in which the model invoked *refine*; *Fallback rate* is the fraction of Fallback Task samples in which the model invoked *fallback*. These rates describe action invocation; corresponding task performance is reported in Table 3.

	Refine Task	Fallback Task
Method	*refine* rate (%)	*fallback* rate (%)
Our SFT	35.3	2.7
Our RL (ASPO)	**51.3**	**46.7**

The corresponding task scores connect this behavioral change to error recovery (Table 3). Fallback FP increases from 66.0 to 79.7, while Refine FP improves from 67.0 to 69.0; overall performance rises from 67.9 to 75.8. Thus, the larger change in fallback use accompanies the larger FP improvement on the matching task. Invocation rates describe behavior, while FP measures its resulting output quality. Taken together, the action and performance evidence supports RL improving the use of learned correction behaviors on flawed intermediate states.

5.3 Image-Policy Optimization and Reward Assignment

The controlled comparisons in Tables 5–6 use 32 prompts per optimizer step with $K = 8$ and $R = 3$, compared with 64 prompts in the main RL run. We assess the component effects by comparing variants within this shared 32-prompt setting.

Image-policy optimization. An intermediate image supplies context for subsequent correction decisions, motivating its inclusion in trajectory-level optimization. Table 5 tests the image-policy term: both RL variants improve over SFT, while including the surrogate further raises GenEval++ from 0.418 to 0.429 and CorrectionBench from 69.72 to 71.55. The gains on both benchmarks support training image-producing actions from the final trajectory outcome alongside the language-action branch.

Table 5: Image-policy surrogate ablation at 32 prompts per step.

Variant	GenEval++	CorrectionBench
SFT	0.407	67.90
ASPO w/o image-policy surrogate	0.418	69.72
ASPO	**0.429**	**71.55**

Reward assignment. Table 6 compares final-image-only feedback with dense per-round rewards. Dense feedback raises GenEval++ from 0.429 to 0.436, but reduces CorrectionBench from 71.55 to 70.94. It also requires judge calls for intermediate rounds. Dense feedback therefore trades a higher GenEval++ score for a lower CorrectionBench score in this comparison. Final-image-only reward supports correction gains while evaluating the completed trajectory, and we retain it to train the joint policy without additional intermediate-round judge calls. Appendix D reports sensitivity to K, R, and the training reward judge.

Table 6: Reward assignment ablation at 32 prompts per step.

Reward design	GenEval++	CorrectionBench
Final-image-only	0.429	**71.55**
Dense per-round	**0.436**	70.94

6 Conclusion

ASPO formulates interleaved self-correction as a variable-length policy over reasoning, correction actions, and image generation. SFT establishes the correction behaviors, and on-policy RL jointly trains their deployment and visual execution from final-image feedback. Stochastic transition replay enables this objective to train image-producing steps alongside language-action decisions. Action-vocabulary ablations, changes in task-specific action use, and image-policy ablations connect the training design to improved error recovery on CorrectionBench, corroborated by human evaluation and gains on generation benchmarks.

AI USE STATEMENT

Generative AI tools were used for data construction and automated evaluation. As described in Section 3 and Appendix A, Gemini 3.1 Pro was used to distill error patterns and support the construction of synthetic interleaved training trajectories. For CorrectionBench, GPT-4o was used to audit generation errors, Gemini-3.1-Pro to expand textual scenarios and evaluate model outputs, and Gemini-3.0-Flash-Image to synthesize erroneous images. Qwen3.6-27B provided the RL training reward; Appendix D.3 additionally evaluates Qwen2.5-VL-72B as a training judge. The benchmark evaluation prompt is provided in Appendix B.3. The authors are responsible for the final content of this work, including AI-assisted text, data, and other artifacts.

ETHICS STATEMENT

ASPO studies adaptive self-correction for interleaved text-image generation. Although improved correction can support creative applications, it may also facilitate the generation of misleading or harmful imagery. The system may inherit biases from its underlying models, synthetic training data, and automated reward and evaluation models. Improvements on the reported benchmarks should therefore not be interpreted as evidence of safety or fairness. The construction of the synthetic training trajectories and CorrectionBench is described in Appendices A and B. Human evaluation on CorrectionBench involved three independent annotators who were informed about the evaluation and agreed to participate. None was involved in benchmark construction or model training. Annotators were not informed of the model identities or automated evaluation scores, as detailed in Appendix B.4.

REPRODUCIBILITY STATEMENT

Section 3 describes the trajectory formulation, supervised initialization, and reinforcement learning objectives. Appendix A describes SFT construction, supervision, behavior statistics, overlap checks, and illustrative text traces. Appendix B gives the CorrectionBench composition, model inputs, evaluation prompt, and human validation. Appendix C consolidates training configurations and compute. Section 5 reports component ablations, and Appendix D reports hyperparameter and training-judge sensitivity with their experimental settings.

REFERENCES

Ruichuan An, Sihan Yang, Ziyu Guo, Wei Dai, Zijun Shen, Haodong Li, Renrui Zhang, Xinyu Wei, Guopeng Li, Wenshan Wu, et al. Genius: Generative fluid intelligence evaluation suite. *arXiv preprint arXiv:2602.11144*, 2026.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

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

Chengqi Duan, Rongyao Fang, Yuqing Wang, Kun Wang, Linjiang Huang, Xingyu Zeng, Hongsheng Li, and Xihui Liu. Got-r1: Unleashing reasoning capability of mllm for visual generation with reinforcement learning. *arXiv preprint arXiv:2505.17022*, 2025.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. *arXiv preprint arXiv:2508.04324*, 2025.

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

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

Siyu Jiao, Yiheng Lin, Yujie Zhong, Qi She, Wei Zhou, Xiaohan Lan, Zilong Huang, Fei Yu, Yingchen Yu, Yunqing Zhao, et al. Thinkgen: Generalized thinking for visual generation. *arXiv preprint arXiv:2512.23568*, 2025.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Yiming Cheng, Miles Yang, Zhao Zhong, and Liefeng Bo. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025.

Chao Liao, Liyang Liu, Xun Wang, Zhengxiong Luo, Xinyu Zhang, Wenliang Zhao, Jie Wu, Liang Li, Zhi Tian, and Weilin Huang. Mogao: An omni foundation model for interleaved multi-modal generation. *arXiv preprint arXiv:2505.05472*, 2025.

Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld-v1: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *CVPR*, 2024.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

Jie Liu, Zilyu Ye, Linxiao Yuan, Shenhan Zhu, Yu Gao, Jie Wu, Kunchang Li, Xionghui Wang, Xiaonan Nie, Weilin Huang, et al. Unigrpo: Unified policy optimization for reasoning-driven visual generation. *arXiv preprint arXiv:2603.23500*, 2026.

Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Kunpeng Ning, Bin Zhu, et al. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025.

Yang Shi, Yuhao Dong, Yue Ding, Yuran Wang, Xuanyu Zhu, Sheng Zhou, Wenting Liu, Haochen Tian, Rundong Wang, Huanqian Wang, et al. Realunify: Do unified models truly benefit from unification? a comprehensive benchmark. *arXiv preprint arXiv:2509.24897*, 2025.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Feng Wang and Zihao Yu. Coefficients-preserving sampling for reinforcement learning with flow matching. *arXiv preprint arXiv:2509.05952*, 2025.

Fu-Yun Wang, Han Zhang, Michael Gharbi, Hongsheng Li, and Taesung Park. Promptrl: Prompt matters in rl for flow-based image generation. *arXiv preprint arXiv:2602.01382*, 2026.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

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

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *CVPR*, 2025b.

Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. Liquid: Language models are scalable and unified multi-modal generators. *IJCV*, 2026.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *ICLR*, 2025.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Sen Ye, Mengde Xu, Shuyang Gu, Di He, Liwei Wang, and Han Hu. Understanding vs. generation: Navigating optimization dilemma in multimodal models. In *ICLR*, 2026.

Shanshan Zhao, Xinjie Zhang, Jintao Guo, Jiakui Hu, Lunhao Duan, Minghao Fu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, et al. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*, 2025.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.

Yujie Zhou, Pengyang Ling, Jiazi Bu, Yibin Wang, Yuhang Zang, Jiaqi Wang, Li Niu, and Guangtao Zhai. Fine-grained grpo for precise preference alignment in flow models. *arXiv preprint arXiv:2510.01982*, 2025.

A SFT Data Construction and Text Traces

A.1 Construction and supervision

The cold-start dataset contains 28,842 newly constructed interleaved trajectories. Construction separates the analysis of model weaknesses from the creation of training instances. We first collect erroneous trajectories produced by existing models across benchmarks and use Gemini 3.1 Pro to distill abstract error patterns. We combine these patterns with sampled domain tags to guide new generation objectives. The analyzed benchmark trajectories themselves are not copied into the SFT dataset.

For each new objective, we construct a prompt, generation plan, and correct target image, retaining the instance after quality filtering. We inject zero, one, or two controlled erroneous intermediate states. Local errors motivate refinement; incorrect premises or fundamentally wrong outputs motivate fallback. Diagnosis text and the corresponding action are followed by a corrected image and final verification. Subsequent steps can refer to earlier images through the retained interleaved context. The format is:

> Task/context → reasoning and action → image state → diagnosis and corrective action, when needed → correct target → final verification.

`<ins>` initializes generation, `<refine>` requests local correction, and `<fallback>` requests a new attempt conditioned on the retained history. Final verification ends with `Generation is complete.` and corresponds to the terminal decision; it does not emit another image. Error-free initial outputs require verification but no corrective image generation.

Erroneous intermediate images receive image-loss mask 0, and only the final correct target receives mask 1. Incorrect initial reasoning in wrong-premise fallback examples is also excluded from the text loss. These masks concern supervised initialization; RL instead samples its own trajectories and optimizes image-emitting actions using trajectory-level advantages.

A.2 Behavior statistics

Table 7 reports trajectory lengths and action coverage. Lengths here count *image-producing turns*, whereas the RL limit R counts decision rounds, including terminal if selected. Action-coverage categories can overlap because a trajectory may contain both refinement and fallback. Every trajectory includes terminal verification.

Table 7: Cold-start SFT trajectory statistics. Percentages are rounded.

Statistic	Value
Total trajectories	28,842
Image-producing turns: mean / median / maximum	1.50 / 1 / 3
One-image trajectories	17,280 (59.9%)
Two-image trajectories	8,640 (30.0%)
Three-image trajectories	2,922 (10.1%)
Contains `<refine>`	6,268 (21.7%)
Contains `<fallback>`	7,242 (25.1%)
Contains terminal verification	28,842 (100%)

Table 8 lists image-producing action patterns; terminal verification follows each pattern and is omitted from the sequences. All 28,842 complete assistant trajectories are distinct, as are 72,115 of the 72,168 CoT blocks.

A.3 Data provenance and overlap audit

Existing failures guide error-pattern analysis, while SFT prompts, reasoning, erroneous images, action annotations, and final targets are newly constructed. An exact-string comparison between final SFT

Table 8: SFT correction patterns before final verification.

Action sequence	Count	Proportion
`ins`	17,280	59.9%
`ins` → `refine`	4,320	15.0%
`ins` → `fallback`	4,320	15.0%
`ins` → `fallback` → `refine`	1,948	6.8%
`ins` → `fallback` → `fallback`	974	3.4%

prompts and evaluation prompts finds no matches in the three checked sets (Table 9). These results describe prompt-string overlap.

Table 9: Exact-string overlap between SFT and evaluation prompts.

Evaluation set	Prompts checked	Exact matches
WISE	1,000	0
GENIUS	401	0
GenEval++	280	0

A.4 Illustrative text traces

These textual summaries illustrate representative examples and their supervision. Image states and diagnoses are described in words; the summaries are not verbatim model transcripts.

Refinement: correcting local bird attributes. **Task and initial state:** Generate an African grey parrot. The intermediate image has a long blue tail and a straight beak.
Diagnosis and action: Identify the incorrect attributes and emit `<refine>` to request a short red tail and a hooked black beak.
Target and verification: The corrected target satisfies these attributes, followed by verification ending with `Generation is complete.`
Supervision: The erroneous image has image-loss mask 0; the final correct target has mask 1.

Fallback: correcting an incorrect premise. **Task and initial state:** A request for Angola's national animal produces a rhinoceros.
Diagnosis and action: Correct the mistaken animal identity and emit `<fallback>` to generate a giant sable antelope, retaining the prior context.
Target and verification: The target depicts the requested animal, followed by final verification.
Supervision: The erroneous image and incorrect initial reasoning are masked from their respective losses; the final correct image receives image supervision.

Termination: accepting a satisfactory result. **Task and current state:** Pink scissors, a grey keyboard, and a yellow backpack are generated with the requested object–color bindings.
Diagnosis and action: Verify the bindings and end with `Generation is complete.`
Target and supervision: The existing correct image is the final target and receives image-loss mask 1. Verification terminates the trace without another image-producing turn.

B CorrectionBench and Evaluation

B.1 Construction, composition, and model inputs

CorrectionBench contains 350 manually constructed and curated interleaved correction instances. Logical errors concern multimodal or text-grounded reasoning; detail errors include incorrect text, missing or inconsistent objects, and anatomical artifacts; and quality degradation concerns visibly degraded generation. Logical and quality examples are constructed by analyzing failures, expanding

their scenarios, and synthesizing erroneous images as described in Section 3.4. Detail artifacts are manually curated from generated daily-life and narrative examples.

Error type and input structure are separate classifications of the same 350 instances (Table 10). Single-image correction focuses on the latest erroneous output. Multi-image correction additionally involves prior, reference, or contextual images. There are no pure text-to-image instances starting without an erroneous image.

Table 10: Two classifications of the same 350 CorrectionBench instances.

Classification	Category	Instances
Error type	Logical reasoning error	150
	Generation quality degradation	150
	Detail generation error	50
Input structure	Pure text-to-image generation	0
	Single-image-conditioned correction	120
	Multi-image preference/context correction	230

Each evaluated model receives the original request, previous text–image history, generation plan, and deliberately flawed latest image. Corrective continuations and target reference images retained during construction are not exposed to the model. These hidden targets are distinct from any prior or reference images included as part of the instance's input context.

Refine cases contain localized errors that can be repaired while preserving valid content. Fallback cases contain severe failures requiring a new generation attempt. Input image count does not itself determine which action is appropriate.

B.2 Illustrative text traces

These explanatory traces describe instance states and expected behavior, rather than reproducing model outputs. The desired correction is described here for illustration; the model does not receive the construction-time corrective continuation or target reference image.

Refine: correcting spatial arrangement. **Request and flawed state:** Five objects should follow a specified horizontal order, but the image stacks them vertically.
Required diagnosis: The arrangement violates the requested spatial relation.
Expected continuation: Refine the image to preserve the objects and place them in the required horizontal order.

Refine: correcting grouped counting. **Request and flawed state:** Fifteen toy cars should form three rows of five, but they are unevenly clustered.
Required diagnosis: The grouping does not satisfy the requested row structure.
Expected continuation: Refine the arrangement into three rows of five cars each.

Fallback: replacing an unrelated subject. **Request and flawed state:** The request specifies a bald eagle, while the image depicts a peacock.
Required diagnosis: The generated subject does not match the request.
Expected continuation: Invoke fallback to generate the requested bald eagle from the available context.

B.3 Automatic evaluation

Gemini-3.1-Pro evaluates Follow Plan (FP) and Aesthetic Quality (AQ) on a 0–2 scale, combined with weights 0.9 and 0.1. Benchmark result tables report scores on a 0–100 scale. The judge receives a textual Hint describing the correct target and the image to assess. The complete evaluation prompt is retained below.

Evaluation Prompt

You are an expert evaluator for AI-generated images.
You will be given:

1. A **Hint** -- a one-sentence description of what the correct generated image should look like.
2. The **generated image** to evaluate.

SCORING DIMENSIONS

DIMENSION 1: FOLLOW PLAN (WEIGHT 0.9)

Does the generated image match the Hint AND correct any known errors (if applicable)?

- **2 (Fully follows):** The image matches the Hint completely. All key visual elements, attributes, and details are correctly present. If there were errors to correct, they are fully resolved.
- **1 (Partially follows):** The image matches some parts of the Hint but misses or incorrectly renders important elements. OR errors are only partially corrected.
- **0 (Does not follow):** The image does not follow the Hint. Key elements are missing or wrong, and errors (if any) are not corrected.

DIMENSION 2: AESTHETIC QUALITY (WEIGHT 0.1)

How visually appealing and technically well-rendered is the image? (Independent of Hint content.)

- **2 (High quality):** Sharp, well-composed, natural lighting, no artifacts.
- **1 (Medium quality):** Acceptable but has minor issues (slight blur, minor artifacts, awkward composition).
- **0 (Low quality):** Obviously degraded, heavily distorted, or visually unpleasant.

FINAL SCORE

final_score = 0.9 × follow_plan + 0.1 × aesthetic

Evaluation Prompt (continued)

OUTPUT FORMAT

Return ONLY a JSON object (no markdown fences):

```
{
  "follow_plan": 0 or 1 or 2,
  "follow_plan_reason": "One sentence explaining match with
  Hint (max 35 words)",
  "aesthetic": 0 or 1 or 2,
  "aesthetic_reason": "One sentence on visual quality (max 15
  words)",
  "score": <computed: 0.9*follow_plan + 0.1*aesthetic>,
  "reason": "One sentence overall summary (max 35 words)"
}
```

B.4 BLINDED HUMAN EVALUATION

Three annotators who were not involved in benchmark construction or model training evaluated outputs for all 350 CorrectionBench instances using the same FP/AQ rubric. Annotators were not informed of the model identities or automated evaluation scores. Table 11 compares human and automatic evaluation. Human scores improve from 67.61 for SFT to 75.14 for ASPO, a 7.53-point gain with a reported 95% confidence interval of $[6.21, 8.89]$. Automatic main-table scores improve from 67.9 to 75.8, a 7.9-point gain.

Table 11: Human and automatic evaluation on CorrectionBench (0–100 scale).

Evaluation	SFT	ASPO	Improvement
Human	67.61	75.14	+7.53
Automatic	67.9	75.8	+7.9

Agreement statistics are reported in Table 12. These measurements support agreement between the automatic judge and human assessment, while the human comparison corroborates the observed correction gain.

Table 12: Reported agreement statistics for CorrectionBench evaluation.

Statistic	Value
Gemini–human Spearman's ρ	0.68
Gemini–human MAE (0–2 scale)	0.31
Gemini–human quadratic-weighted κ, FP	0.60
Gemini–human quadratic-weighted κ, AQ	0.55
Human mean pairwise weighted κ, FP	0.64
Human mean pairwise weighted κ, AQ	0.60

C TRAINING AND IMPLEMENTATION DETAILS

C.1 BASE MODEL AND TRAINABLE COMPONENTS

All SFT and ASPO experiments use BAGEL-7B-MoT and its native rectified-flow image branch. Language-action and image-generation branches are updated during SFT and RL, while ViT and VAE remain frozen. RL updates include both MoT language experts, CoT/action-token parameters, and the image branch. Initialization, refinement, and fallback condition the same image generator on the full interleaved history.

C.2 TRAINING CONFIGURATION AND COMPUTE

Tables 13 and 14 consolidate the settings. RL uses 8,871 training and 1,161 validation prompts, with trajectories sampled on policy. The main run uses $K = 8$ rollouts for each of 64 prompts per step, yielding 512 trajectories. The maximum $R = 3$ counts decision rounds and allows early termination. For example, `ins` $\rightarrow$ `refine` $\rightarrow$ `terminal` uses three decision rounds and two images; reaching the cap without terminal returns the most recent image.

C.3 TRAINING REWARD AND IMAGE-POLICY REPLAY

The training reward judge is Qwen3.6-27B, an open-weight vision-language model. It receives the original prompt and only the final returned image. It evaluates prompt compliance for text-to-image and editing prompts; for GENIUS-style prompts it combines RC, VC, and AQ through a weighted score normalized to $[0, 1]$. Intermediate images are not scored separately. Prompt-group rewards are centered to obtain the trajectory advantages in Eq. 3.

Language-action updates apply these advantages to generated reasoning and action tokens. The image branch replays one stored stochastic denoising transition per emitted image, recomputes its log

Table 13: Cold-start SFT configuration.

Setting	Value
Training trajectories	28,842
Training steps	8,000
Learning rate	10^{-5}
Maximum sequence length	32,768
GPUs	16

Table 14: Main ASPO RL configuration. Controlled ablations in Section 5 and sensitivity experiments in Appendix D use 32 prompts per step.

Setting	Value
Training / validation prompts	8,871 / 1,161
Rollouts per prompt, K	8
Maximum decision rounds, R	3
Prompts / trajectories per optimizer step	64 / 512
Optimizer	AdamW
Training steps	100
Learning rate	2×10^{-6}
Warmup steps	50
Gradient clipping	0.5
Precision / distributed training	BF16 / FSDP
Image-loss coefficient, λ_{IM}	2.0
Training reward judge	Qwen3.6-27B
GPUs	128 $\times$ 80GB
Approximate time per optimizer step	27 minutes
Reward judge deployment	Separate server

density, and weights it by the same trajectory advantage. The Gaussian transition, replay surrogate, and joint objective remain in Eqs. 6–9. Terminal rounds contribute no image-policy term. The optional language-policy KL regularizer is defined at the method level in Section 3.3.

Qwen3.6-27B supplies RL training feedback, whereas Gemini-3.1-Pro evaluates CorrectionBench. The benchmark's FP/AQ weights of 0.9/0.1 are evaluation weights and are not substituted for the RL reward's aggregation rule.

D Hyperparameter and Reward-Judge Sensitivity

D.1 Experimental scope

The controlled RL ablations in Section 5 and the sensitivity experiments below use **32 prompts per optimizer step**, compared with **64 prompts and 512 trajectories** in the main RL run. Unless varied explicitly, remaining settings follow Appendix C, including $K = 8$ and $R = 3$. Variants within each comparison share the stated setting. Their absolute scores should not be treated as a direct ablation of the 64-prompt main-run results. GenEval++ uses the 0–1 scale, and CorrectionBench uses the 0–100 scale.

D.2 Rollout count and decision-round limit

Table 15 varies K with $R = 3$. More rollouts improve both benchmarks but increase the number of sampled trajectories; $K = 8$ provides a practical performance–cost trade-off. Table 16 varies R with $K = 8$. Moving from two to three decision rounds improves both scores, while a fourth round gives no additional GenEval++ gain and a smaller CorrectionBench improvement. The main configuration therefore uses $R = 3$.

Table 15: Rollout-count sensitivity with $R = 3$ and 32 prompts per step.

K	GenEval++	CorrectionBench
4	0.414	69.72
8 (default)	0.429	71.55
16	0.436	72.76

Table 16: Decision-round sensitivity with $K = 8$ and 32 prompts per step.

R	GenEval++	CorrectionBench
2	0.421	70.33
3 (default)	0.429	71.55
4	0.429	72.15

D.3 Training reward judge

Replacing Qwen3.6-27B with Qwen2.5-VL-72B still improves both benchmarks over SFT (Table 17). Qwen3.6-27B achieves higher scores in this comparison, but gains persist under both tested training judges.

Table 17: Training-judge sensitivity at 32 prompts per step.

Training setting	GenEval++	CorrectionBench
SFT baseline	0.407	67.90
Qwen3.6-27B (default)	0.429	71.55
Qwen2.5-VL-72B	0.425	69.22