LOCAL DIFFUSION CONTROL: A TOOL INTERFACE FOR VLM-GUIDED CORRECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image generators often produce visually plausible scenes but make local semantic errors, such as omitting an object, assigning an attribute to the wrong object, or introducing unwanted details. In this work, we investigate whether such errors can be corrected within the same sampling trajectory, after they become visible but before denoising is complete. We introduce Direct-CFG, a spatially and temporally bounded intervention interface that uses intermediate clean-image estimates to implement three operations: strengthening visible elements, suppressing unwanted content, and locally re-noising a region followed by its regeneration. We distinguish between two tasks: the effectiveness of the correction tools themselves and the ability of vision-language models (VLMs) to determine when, where, and how to apply them. We evaluated the approach using fixed human-specified interventions and automated VLM control, revealing the capabilities and limitations of local correction of generation trajectories.

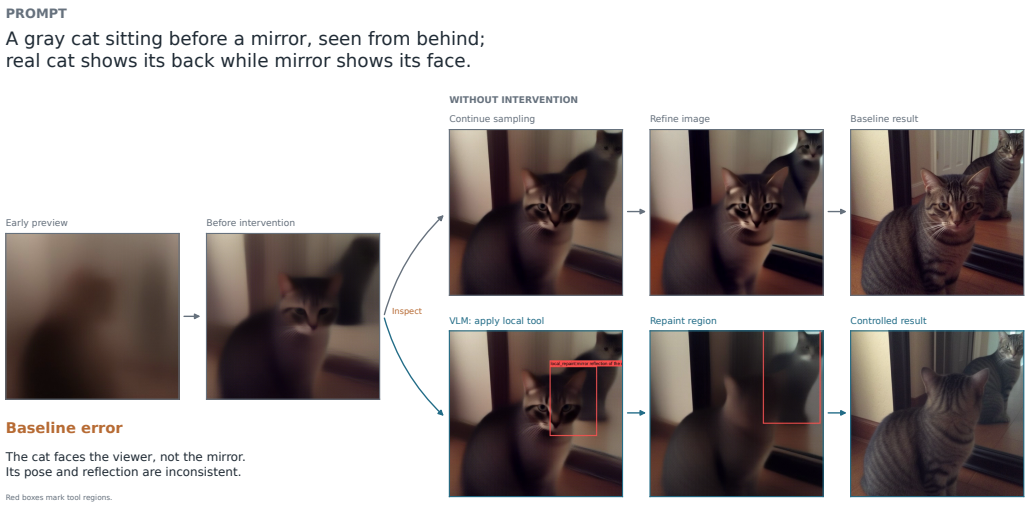


Figure 1: **Correcting a trajectory during generation.** The prompt asks for a cat seen from behind, with its face visible in the mirror. Without intervention (top), the cat faces the viewer and its reflection is inconsistent with its pose. VLM-guided tool calls (bottom) change the evolving image toward the requested arrangement. Red boxes show recorded tool regions. This selected example illustrates the mechanism, not an aggregate success rate or guaranteed preservation.

1 INTRODUCTION

Text-to-image models can produce a convincing scene while **missing a requested object**, assigning a color to the wrong instance, or adding an unwanted element. When the rest of the image is satisfactory, generating another sample may replace content the user wanted to keep. Editing the

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.

completed image offers another route. Here, correction is studied before sampling ends, **when an error is already visible** in an intermediate preview and the generator still has denoising steps remaining.

Direct-CFG is introduced as an interface for local interventions within an ongoing sampling trajectory. An **inspector** is the human or vision-language model (VLM) that examines an intermediate preview and chooses whether, where, and how to intervene. It specifies a target region, an edit instruction, and an active interval; the controller validates and executes the call, while the generator continues denoising. The interface supports **strengthening** visible content, **suppressing** unwanted content, and **locally re-noising** a region to regenerate it. The inspector can also choose to leave the trajectory unchanged. The original prompt, initial seed, generator, and sampling schedule are kept fixed. Each intervention changes the remaining denoising steps without **restarting the full image generation**.

Our central question is whether intermediate previews and explicit actions provide useful control over an unfinished image. We test **what the tools can change**, how feedback across attempts helps select actions, and how a ready-made VLM uses the same interface. Separating these questions matters: failure can arise from the preview, error recognition, localization, tool choice, or the generator's response. A final-image score alone cannot identify the cause.

The evaluation first tests individual tools with prescribed edit instructions and automatically constructed target regions. For each case, three versions are generated from the same prompt and seed: one without intervention, one using the tool designed for the requested change, and one using a different tool as a control. These versions are checked to determine whether the edit works, whether the rest of the image stays unchanged, and whether the edit creates visible defects. Next, interactive trajectories from joint human–agent correction with automatic VLM suggestions are analyzed, including failed edits and unchanged outputs. Finally, VLM-controlled trajectories are examined to identify errors in perception, localization, and action selection.

The paper makes three contributions:

- A local intervention interface with explicit spatial and temporal bounds for guidance, suppression, and regional re-noising, with the option to leave the trajectory unchanged.
- An evaluation of each tool that checks whether it makes the intended change without altering the rest of the image or creating visible defects.
- An analysis of saved interactive trajectories and VLM decisions that links requested actions to intermediate previews and final outcomes.

2 Related Work

Visual feedback and tool-based correction. PPAD (Lv et al., 2025) uses lookahead previews, corrective conditioning, and rollback/resampling. Adaptive negative prompting (Golan et al., 2026) updates conditioning from previews; GAZER (Chen et al., 2026) uses rollout and rewind in autoregressive generation. SLD (Wu et al., 2023) plans object-level repairs, and GenArtist (Wang et al., 2024) selects editing tools with verification and backtracking. Visual feedback and tool selection thus precede this work. We study region- and time-bounded actions within an ongoing denoising trajectory, separating tool capability from inspector choice.

Semantic guidance and regional regeneration. CFG (Ho & Salimans, 2022) combines conditioned predictions; SEGA (Brack et al., 2023) adds or subtracts semantic directions. Attend-and-Excite (Chefer et al., 2023) and BoxDiff (Xie et al., 2023) impose attention constraints, while Dynamic CFG (Papalampidi et al., 2026) adapts guidance scale. Our masked prediction differences follow this guidance principle; tools add text conditions while retaining the original prompt. SDEdit (Meng et al., 2021), RePaint (Lugmayr et al., 2022), Blended Diffusion (Avrahami et al., 2021), and Differential Diffusion (Levin & Fried, 2023) control editing through noise, masks, or blending. Restart sampling (Xu et al., 2023) revisits higher noise levels. Our repaint starts from the current preview, runs a regional branch, and rejoins the main schedule. DiffusionBrowser (Hong et al., 2025) likewise explores intermediate previews and interactive variations.

Evaluation. GenEval (Ghosh et al., 2023) and TIFA (Hu et al., 2023) measure text–image alignment. We use GenEval2's Soft-TIFA (Kamath et al., 2025) and assess preservation and defects separately. Retrospectively selected interactive finals describe search outcomes, not autonomous benchmark success.

3 Problem Formulation

We study corrections made while an image is being generated. The original prompt, initial seed, and generator remain fixed. At selected steps, an inspector sees a preview and requests an edit within a box. The generator then continues from its current state. We distinguish the inspector's requested action from the intervention accepted and executed by the controller.

Generation state and scheduler. Let z_j be the image latent before main step j, for $j = 0, \ldots, N-1$. A latent is the compressed representation on which the generator operates; decoding it produces an image. The scheduler supplies decreasing noise levels $\sigma_0 > \cdots > \sigma_N = 0$. Qwen-Image (`Qwen/Qwen-Image`, not Qwen-Image-Edit) predicts a flow velocity, denoted $v_\theta(z_j, \sigma_j; c)$ for text condition c. The scheduler uses this prediction to move to the next noise level:

$$z_{j+1} = z_j + (\sigma_{j+1} - \sigma_j) v_j. \tag{1}$$

Here v_j includes the original prompt guidance and any previously activated tool corrections. The noise-level difference sets the step size. The tools change the prediction supplied to the scheduler; push and suppress do not change the main sequence of noise levels.

What the inspector sees. We decode a predicted clean latent rather than the noisy latent itself:

$$\widehat{z}_j = z_j - \sigma_j v_j, \qquad I_j = D(\widehat{z}_j). \tag{2}$$

D is the generator's decoder. This preview is an estimate of the current scene, not a completed sample or a separate lookahead rollout. It includes the effect of tool guidance already active at step j.

Each request includes the original prompt, current RGB preview, remaining step fraction $p_j = 1 - j/(N-1)$, inspection schedule, tool schemas, and a short text history of active actions, recent calls, and findings. Earlier preview images are not included. Note that p_j differs from the scheduler noise σ_j.

The inspector returns findings and structured calls: normalized box, edit text, preservation clause, and active interval; push/suppress also specify strength. The controller validates calls and enforces configured limits. A new call is handled after the current main step: push/suppress affect later steps, while repaint rejoins at σ_{j+1}. Current repaint noise and guidance strengths are runtime settings, though older replayed calls may specify them. Full input and execution details are in Appendix C.

4 Local Trajectory Tools

Shared update for push and suppress. The controller converts a box to a soft latent mask M_k. A value of one applies the full correction, zero applies none, and intermediate values blend the boundary. Let $\mathcal{A}_j$ contain accepted actions whose intervals include step j and whose clean-tail cutoff has not been reached. Then

$$v_j = v_j^{\text{base}} + \sum_{k \in \mathcal{A}_j} \Delta v_{k,j}. \tag{3}$$

v_j^{base} is the usual prompt-conditioned Qwen prediction, including its CFG normalization, evaluated on the current corrected latent. It is not the prediction from a separately generated baseline image.

Both tools use two additional text-conditioned predictions:

$$\Delta v_{k,j} = \alpha_k g M_k \odot \left[v_\theta(z_j, \sigma_j; c_k^+) - v_\theta(z_j, \sigma_j; c_k^-) \right]. \tag{4}$$

g is the generator CFG scale, α_k is the tool strength, and $\odot$ means elementwise multiplication. The difference supplies a text-conditioned correction direction; the mask limits where it is added. The original prompt still supplies v_j^{base}.

For `cfg_push`, c_k^+ combines the desired content and preservation clause, while c_k^- describes an unwanted alternative or is blank. The tool guides the region without adding fresh noise. For `cfg_suppress`, c_k^- describes the unwanted content and c_k^+ is the preservation clause, defaulting to *clean realistic image region* when absent. Suppression uses the same difference operator with these conditions; it does not reverse the entire generator prediction. These actions request changes rather than guaranteeing that the generator will realize them.

Regional regeneration with `cfg_repaint`. Repaint gives a region additional denoising steps. Suppose inspection occurs at main step j. Set $q = j + 1$, the point already reached by the main scheduler, and retain the preview latent $\widehat{z}_j$. The tool starts a temporary branch at an earlier, noisier schedule index $r < q$. With fresh noise $\epsilon \sim \mathcal{N}(0, I)$, define a noised reference and initialize the branch:

$$R_i = (1 - \sigma_i)\widehat{z}_j + \sigma_i \epsilon, \qquad b_r = R_r. \tag{5}$$

The same noise sample is used for all reference levels within this branch. For each $i = r, \ldots, q - 1$, the generator predicts a velocity u_i using the repaint text conditions and normalized CFG. After its scheduler step, the branch restores the region outside the mask to the noised reference:

$$b_{i+1} = M_k \odot \left[b_i + (\sigma_{i+1} - \sigma_i)u_i\right] + (1 - M_k) \odot R_{i+1}. \tag{6}$$

At the rejoin point, only the masked branch result is inserted into the main latent:

$$z_q^{\text{edit}} = M_k \odot b_q + (1 - M_k) \odot z_q. \tag{7}$$

The branch reuses the main scheduler's already shifted noise levels. It does not construct a new full schedule or denoise to zero before rejoining. Generation resumes at the same main index, with additional masked guidance scheduled for the requested interval. The extra branch steps add computation beyond the main N steps.

An optional outside-region lock continues a pre-repaint reference trajectory and restores its unmasked latents at every remaining step, adding model computations. It does not guarantee identical decoded pixels (Appendix C).

Timing and no intervention. Push, suppress, and repaint guidance act only over their accepted intervals. The clean-tail cutoff disables these guidance corrections on the final main steps; it does not undo earlier edits or disable the optional outside lock. `no_op` adds no intervention. Generation and already active actions continue, so it does not mean that the latent stops changing.

The masks bound direct edits rather than every eventual image difference. Denoising and decoding can propagate changes beyond the target. We therefore evaluate the requested change, preservation of unrelated content, and new artifacts separately.

The equations above describe scheduler-restart repaint, not older one-shot latent-blend branches. Recovered settings and unresolved historical provenance are listed in Appendix A.

The guarded preset limits strength, duration, repaint area, and repeated edits. Experiment 2 also includes unrestricted diagnostic runs, as specified in Section 5.2.

5 Experimental Design

We separate **tool capability**, **interactive correction**, and **automatic control** (Table 1). Controlled runs use Qwen-Image at 512×512, 50 main steps, and CFG 4. Prompt and seed are paired within each case; repaint adds evaluations beyond the 50 main steps. The experiments used different code versions and runtime settings, so we report their results separately rather than combine them into a single comparison. **Beyond a single generator or inspector.** The FLUX implementation shows that the preview–action interface can be implemented **beyond Qwen-Image**, although correction quality has not been evaluated across generators. The Gemma-LoRA pilot demonstrates a route to **adapting a small, locally served VLM as an inspector**, potentially reducing reliance on remote models and inspection latency. A subsequent diagnostic comparison examines the **speed–quality trade-off** (Appendix D).

Table 1: Study design. Experiment 1A uses a separate fixed-setting confirmation pool; 1B separates development and held-out prompts. Experiments 2 and 3 use previously seen diagnostic prompts.

Study	Cases	Who chooses the action?	What is evaluated?
1A: tools	Development: 20 per task; confirmation: 80/76/80 for insertion/attribute/removal	Prescribed calls; detector or heuristic boxes	Baseline, intended tool, and alternative tool
1B: oracle	100 development + 100 held-out prompts	Rule-based planner given failed baseline atoms	Initial settings and a frozen retimed configuration
2: interactive	50 mixed + 50 complex GenEval2 prompts	Human–agent feedback across attempts	Historically selected final, including baseline fallback
3: automatic	100 general + 100 GenEval2 prompts	Astra, four inspections	One original corrected final per baseline

5.1 Experiment 1: prescribed and oracle-assisted calls

1A tests tools without an online inspector. Calls are prescribed; OWLv2 or a free-space heuristic supplies boxes. We compare repaint for insertion, push for attributes, and suppress for removal with alternative tools. Removal and half the attribute cases are constructed edits. Development uses four calibration and 20 test cases per task; test feedback informed tuning. Confirmation uses a separate 200-prompt pool with two seeds and frozen settings, leaving 80/76/80 cases after four already-correct attribute exclusions. For removal we report a later same-case test of a pre-existing background-repaint recipe; the originally planned empty-background control remains in the appendix.

1B supplies failed baseline atoms to a rule-based planner, not perfect localization. Timing is selected by GM on 100 development prompts (937 atoms), then tested once on 100 disjoint held-out prompts (924 atoms, balanced across 7–10 requirements). The frozen schedule is 0.78/0.70/0.62, span 0.45, one call per inspection, and cutoff 0.30. Separately, a background-repaint removal recipe is developed on 31 scenes and confirmed on 34 fresh cases with a fixed 16-step restart. Full settings and selection history are in Appendices A.1 and C.

5.2 Experiment 2: interactive correction

Humans and an agent inspect previews and finals, then revise boxes, tools, text, and strength with VLM suggestions. We evaluate 50 mixed and 50 complex GenEval2 prompts. Finals are **retrospectively selected**, with variable attempt budgets and baseline fallback, but frozen before blinded reevaluation. Individual decision authorship is incomplete. Some runs disable controller guards and permit near-full-frame edits; these outcomes do not measure purely local or autonomous correction. The GenEval2 prompts were seen in development.

5.3 Experiment 3: automatic inspection

`gpt-6-astra` inspects at 0.86/0.68/0.50/0.32 on 200 previously seen diagnostic prompts, producing 400 paired finals and 800 records. We evaluate original outputs, not retries. Separate method-blind Astra calls judge finals, with possible shared-model bias. Failed requests are missing decisions, not valid no-ops. Retrospective decision labels are unverified hypotheses; paired preservation ratings are unavailable.

5.4 Evaluation

For 1A, the composite requires **target success, preserved non-target requirements, and outside-box LPIPS** ≤ 0.05. Qwen3-VL rates the semantic checks. Confirmation lacks general artifact ratings but now includes removal patch checks; development additionally uses Claude crop ratings. Object substitution can pass the removal question. Experiments 2/3 use Gemini 3.1 Pro/Astra to judge images independently and method-blind; strict PASS requires prompt compliance and no judged artifacts. Gemini also reviews 186 branch pairs, without edit instructions, which can confound intended and unrelated changes.

Official **Soft-TIFA** computes geometric (GM) or arithmetic (AM) means of atomic scores per prompt, then averages over prompts, on a 0–100 scale. It measures alignment, not preservation. We score 50 GenEval2 pairs in Experiment 2 and 100 in Experiment 3. Paired prompt-level bootstrap intervals, exact McNemar tests, and Wilson intervals quantify uncertainty; 1A comparisons use Holm correction across tasks. They do not account for adaptive selection or evaluator error. Author inspection was informal; **independent human calibration is unavailable**. Detailed criteria, versions, and resampling settings are in Appendices A.1 and C.

6 RESULTS

6.1 CAN EACH TOOL MAKE ITS INTENDED LOCAL CHANGE?

Adding, changing, and removing objects. We first test fixed tool settings on new cases, without further tuning (Table 2). Repaint adds the requested object while preserving other prompt requirements and limiting changes outside the box in 22/80 cases, compared with 1/80 for push. For attribute changes, repaint also passes more often than push: 30/76 versus 4/76. For removal, the background-repaint recipe passes 37/80 cases versus 24/80 for suppress. This composite excludes artifacts: disappearance can mean object substitution, and repaint can leave a flat patch.

Table 2: Fixed-setting confirmation on new cases (the v2 set). A case must pass all three checks: the requested edit occurs, previously satisfied prompt requirements remain satisfied, and outside-box LPIPS is ≤ 0.05. Artifact freedom is not evaluated here. Four attribute cases are excluded because the baseline already satisfies the target. The removal recipe was tested later on the same cases, seeds, and boxes; its settings predate this set. The planned empty-background control passes 8/80. Neither disappearance nor this composite establishes a clean background.

Requested edit	Tool	Cases passing all three checks
Add an object	Repaint	**22/80**
	Push	1/80
Change an attribute	Push	4/76
	Repaint	**30/76**
Remove an object	Suppress	24/80
	Repaint, removal recipe	**37/80**

Wilson 95% intervals for these intended-tool proportions are 19–38%, 2–13%, and 21–41%, respectively.

Achieving the edit and preserving the scene. On the confirmation set, push changes the requested attribute in 24/76 cases, but only 14/76 outputs meet the outside-box similarity threshold. Repaint changes attributes in 37/76 cases and meets that threshold in all 76 (Table 16). General artifact ratings are unavailable; removal-specific patch checks and earlier crop ratings appear in the appendix.

Exact paired McNemar tests, Holm-adjusted over insertion, attributes, and removal with the recipe comparator, favor repaint for insertion ($p = 3 \times 10^{-5}$) and attributes ($p = 0.048$); their composites also favor repaint ($p \leq 3 \times 10^{-6}$). Removal is unresolved: target disappearance is 49/80 for suppress versus 44/80 for the recipe ($p = 0.51$), and composite success is 24/80 versus 37/80 ($p = 0.079$). Crop judgments label another object in 48/80 suppression outputs and only background in none; the recipe leaves a flat patch in 41/80. The strict clean-removal endpoint passes 0/80 and 2/80, respectively. These constructed tasks oppose the original prompt, which still requests the object; crop criteria and their limitations are in Appendix C.12.

Prescribed correction improves held-out alignment. After selecting the timing and span on development prompts, we test the frozen configuration on 100 held-out prompts. GM rises from 17.86 to 20.73, a gain of 2.87 (95% CI [0.50, 5.68]; Table 3). The run fixes 20 atoms and breaks 14; prompts passing every atom increase from 7 to 9. AM improves by 0.78, with an interval including zero $[-0.29, 1.85]$. The initial detector and heuristic arms have gain intervals that include zero.

Table 3: Oracle correction and its alternatives, 100 prompts per set (baseline GM 20.59 and 17.86). Inpainting applies an oracle's calls to the finished baseline; best of N keeps the seed Gemini 3.1 Pro rates highest. Cost includes the baseline run behind the oracle's labels.

Set	Method	GM	ΔGM	95% paired CI	Cost	LPIPS to base
Dev.	Oracle, detector, initial	21.94	+1.35	$[-1.58, 4.71]$	3.54×	–
Dev.	Oracle, heuristic, initial	21.14	+0.55	$[-1.46, 2.96]$	3.56×	–
Dev.	Oracle, detector, retimed	**23.86**	+3.27	$[-0.41, 7.51]$	3.70×	–
Held-out	Oracle, detector, retimed	20.73	+2.87	[0.50, 5.68]	3.74×	0.083
Held-out	Inpainting, same calls	21.97	+4.11	[1.52, 7.21]	3.22×	0.118
Held-out	Best of 2 seeds	20.94	+3.08	[1.28, 5.04]	2×	0.276
Held-out	Best of 4 seeds	**24.86**	+7.01	[3.05, 11.45]	4×	0.431

Table 4: Official Soft-TIFA metrics on the 50-prompt diagnostic GenEval2 subset. Scores and differences are on a 0–100 scale. Finals were fixed before evaluation; intervals condition on these pairs and do not remove adaptive selection bias.

Metric	Baseline	Selected final	Δ	95% paired CI
Soft-TIFA GM	18.76	**47.66**	+28.90	[18.56, 39.65]
Soft-TIFA AM	76.60	**82.74**	+6.14	[3.64, 8.86]

Editing afterwards and generating another sample. Applying the oracle's calls to the finished baseline by inpainting gains as much at lower cost (Table 3; AM $+0.72$ against $+0.78$), and so does a planner that follows Experiment 1 (+3.03 after against +3.02 during generation). Keeping the best of several seeds gains more at equal cost but replaces the image.

Removing an object also means filling its place. We therefore test repainting the unwanted object's region as background, rather than only suppressing the object. On 34 fresh excess-instance cases, the 16-step background-repaint recipe passes all four checks in 11 cases, compared with 2 for suppression. The recipe was fixed before the fresh set was drawn. It removes the extra instance in 19/34 cases versus 5/34 for suppression (exact McNemar $p = 0.0005$; $p = 0.012$ for the four-check composite). The checks require removal, no judged patch or seam, preservation of other prompt requirements, and limited outside-box change. Other variants and the development results are in Appendix B; comparisons across several tested recipes remain exploratory.

Takeaway: the tools provide usable control. The tests establish **three forms of control within an ongoing trajectory**: insertion, attribute changes, and removal. The comparisons identify which actions work better for a given edit and expose their preservation and artifact trade-offs. The held-out oracle gain confirms that prescribed interventions can improve alignment; selecting and executing those interventions remains consequential.

6.2 Interactive feedback improves selected outputs

1. Better agreement with the prompt. Interactive search improves the selected outputs on the 50 complex GenEval2 prompts: Soft-TIFA GM rises from 18.76 to 47.66 and AM from 76.60 to 82.74 (Table 4). Both paired confidence intervals for the gain exclude zero. GM improves in all four groups with 7–10 requirements, although the hardest prompts still score lowest. These are selected results after adaptive attempts, not a full-benchmark or autonomous-inspector score.

2. Correct content can still contain artifacts. Gemini checks prompt compliance and visible defects separately. Compliance alone rises from 44/50 to 48/50 on mixed diagnostic prompts and from 6/50 to 25/50 on complex GenEval2 prompts. Requiring both compliance and no judged artifacts reduces the selected-output counts to 38/50 and 13/50 (Table 5). The strict PASS gains are 4 percentage points (95% CI: 0–10; exact McNemar $p = 0.50$) and 20 points (10–32; $p = 0.00195$), respectively, conditional on the selected pairs.

Table 5: Experiment 2: method-blinded Gemini reevaluation. Strict PASS requires prompt compliance and no judged artifacts. The endpoint was selected before this reevaluation, with variable attempt budgets and baseline fallback. No ambiguous endpoint verdicts occurred. These are not official GenEval2 scores.

Cohort	Baseline PASS	Selected PASS	F→P	P→F
Mixed diagnostic (50)	36/50 (72%)	**38/50 (76%)**	2	0
Complex GenEval2 (50)	3/50 (6%)	**13/50 (26%)**	10	0

Table 6: Experiment 3: method-blind Astra judgments of original finals. Differences and paired bootstrap intervals are in percentage points; p is the two-sided exact McNemar value. Previously used diagnostic prompts, not held-out benchmark results. Retries are excluded.

Cohort	Base	Corrected	F→P	P→F	Δ [95% CI]	p
General (100)	**70/100**	**70/100**	5	5	$0.0\ [-6.0, 6.0]$	1.000
GenEval2 (100)	25/100	**30/100**	14	9	$5.0\ [-4.0, 14.0]$	0.405
All (200)	95/200	**100/200**	19	14	$2.5\ [-3.0, 8.0]$	0.487

3. The rest of the scene is not always preserved. Across reviewed branches, Gemini flags unrelated changes in 26/74 mixed-cohort pairs and 102/112 GenEval2 pairs, and new artifacts in 11/74 and 25/112, respectively. Flags can overlap. These counts include failed attempts and multiple branches per prompt. Because the judge sees no edit mask or action instruction, some intended changes may be flagged as unrelated; this error rate has not been measured. Moreover, 62 of 194 attempts request a box covering at least 90% of the image, and 45 have no recovered or known box, including no-op cases. The gains therefore cannot be attributed solely to small edits: the attempts' tool texts in the prompt, without any trajectory edit, gain +17.6 to +34.6 GM (Appendix C.11).

4. Recorded trajectories. The workflow records **previews, structured calls, and outcomes** across 100 cases, 154 sessions, and 298 branches, including failed attempts. These records make the sequence of edits available for analysis and candidate training examples. Mixed-tool branches do not isolate each call's effect; missing branch and frame records are documented in Appendix C.

Takeaway: correction and data collection. Experiment 2 demonstrates **correction through iterative feedback** and **collection of action–outcome records** in the same workflow. Its gains come mostly from the texts written for the tools and include the benefits of repeated attempts and final selection; baseline fallback can conceal unsuccessful edits. The records offer a concrete basis for studying and learning intervention choices, after provenance checks and filtering.

6.3 Automatic control: alignment gains and decision errors

When should the inspector look? A pilot on an earlier stack (FLUX.2-klein, 28 steps, GPT-5.5; Appendix C.6) bounds the useful window to remaining-step fractions 0.89–0.56: above 0.89 the inspector is at chance, below 0.56 edits no longer land. Inside it, placement beats count: three points at 0.852/0.741/0.630 raise the pass rate on 60 discriminative prompts from 0.392 to 0.492 (McNemar $p = 0.008$), the same three calls at 0.72/0.54/0.36 change nothing, and five or nine points score below three.

Table 6 reports the original 200 paired outcomes. Strict success changes from 95/200 to 100/200, with 19 fixes and 14 regressions. The net gain is 2.5 percentage points (95% paired CI $[-3.0, 8.0]$; McNemar $p = 0.487$). The inspector fixes some failures but also introduces regressions; neither cohort shows a statistically resolved binary gain.

Partial prompt alignment. On the 100-prompt GenEval2 subset, official Soft-TIFA GM rises from 19.90 to 24.36 and AM from 75.55 to 77.07 (Table 7). Both saved paired intervals exclude zero: the inspector improves satisfaction of individual requirements without a resolved gain in complete-image success. These are diagnostic-subset results. Experiment 2 uses different selection and attempt budgets, so it is not a matched operator baseline.

Table 7: Official Soft-TIFA on the diagnostic 100-prompt GenEval2 subset (937 atoms); scores are on a 0–100 scale, not full-benchmark results.

Metric	Baseline	Corrected	Δ	95% paired CI
Soft-TIFA GM	19.90	**24.36**	+4.45	[0.91, 8.39]
Soft-TIFA AM	75.55	**77.07**	+1.51	[0.09, 2.95]

Decision audit. A retrospective model audit labels 73 of 800 inspection records as decision errors, mainly wrong boxes (33) and missed errors (24). These are unverified hypotheses, not human-confirmed causes. Full counts and the diagnostic figure are in Appendix C.

Takeaway: a target for learning tool use. A ready-made VLM can use the interface to achieve **partial alignment gains**, but does not yet deliver reliable complete-image correction. The audit identifies candidate decisions to examine, including localization and error recognition. The records from Experiment 2 provide examples for testing trajectory-based supervision; the present evidence does not rank inspector errors against tool limitations as the dominant cause.

Compact-inspector pilot. A matched diagnostic comparison provides **an initial test of inspector adaptation**: on 100 GenEval2 prompts, Gemma-LoRA increases Soft-TIFA GM from 18.62 to 24.11 relative to Base Gemma, but strict success decreases from 28/100 to 21/100. Base Gemma frequently produces non-executable reports, so this is an end-to-end pipeline comparison, not an isolated test of decision quality. LoRA reduces inspector latency but increases total runtime through more tool activity (Appendix D).

7 Limitations

Control does not guarantee correction. An edit can change nearby objects or introduce visible defects. If applied too late, the remaining generation steps may not be enough to create a missing object. High-strength push often changes content outside the box.

Evaluation scope. We establish intervention feasibility under the tested settings, not reliable correction across prompts, generators, or inspectors. Experiment 2 reports retrospectively selected finals from variable numbers of attempts, with baseline fallback, and 62 of 194 attempts request a box covering at least 90% of the image; its gains show what feedback can reach, not what a single local edit delivers. Experiments 2 and 3 reuse development prompts, so neither is a held-out test. All judgments come from models without human calibration, and in Experiment 3 the same model (Astra) both inspects and judges, so their errors may coincide. In Experiment 1A, an edit counted as successful may still look damaged: on the confirmation set we did not rate the visual quality of the edited region, except for two removal failures (a substitute object and a flat patch).

Training data. The records are candidate training examples, not a validated dataset: decision provenance, filtering, and train/test separation need verification. The compact-inspector pilot improves partial alignment but not strict success; response-format failures and unverified training/evaluation separation prevent attributing the gain to better intervention decisions (Appendix D).

8 Conclusion

We demonstrate **control over an unfinished image**: explicit interventions can insert, alter, or remove content while generation continues. Controlled studies reveal operation-dependent trade-offs in preservation and visual quality. Interactive feedback improves diagnostic prompt alignment through repeated attempts and selection; tool texts explain much of that gain.

The workflow also records **candidate training trajectories** for learning an intervention policy with the generator frozen. A matched Gemma-LoRA pilot improves partial alignment, but not strict success or end-to-end speed. This motivates a valid-output-controlled, held-out training study rather than establishing reliable learned correction. Additional backend implementation and pilot details are in Appendix D.

AI USE STATEMENT

Generative AI tools were used to assist with literature discovery, research ideation, experimental design, software implementation and debugging, analysis of experimental results, and drafting and editing parts of this manuscript. Generative models were also used as components of the studied system, including as multimodal inspectors, as the target/preservation evaluator and crop-quality rater in Experiment 1, as the automatic image and preservation judge in Experiment 2, as the automatic inspector, image judge, and retrospective trajectory annotator in Experiment 3, and as sources of synthetic policy trajectories during preliminary development. The authors manually reviewed AI-assisted text, code, experimental configurations, and reported results. Literature references are checked against original sources. Quantitative results are traced to saved artifacts and evaluation scripts where available; exploratory aggregate-only results and incomplete provenance are identified in the text. The authors take responsibility for the final content, claims, code, data, and conclusions.

REPRODUCIBILITY STATEMENT

The paper specifies the tool updates, recovered generation settings, evaluation criteria, and statistical procedures. Saved experiment packages contain configurations, prompt and seed manifests, images, decision traces, and model ratings. Appendix A distinguishes recovered settings from missing historical provenance. The anonymized supplementary has passed integrity and CPU recomputation checks for saved results and includes linked image evidence; full GPU regeneration and training reproduction have not been verified. Independent human calibration is unavailable, and some historical records are incomplete.

REFERENCES

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. *arXiv preprint arXiv:2111.14818*, 2021. URL https://arxiv.org/abs/2111.14818.

Manuel Brack, Felix Friedrich, Dominik Hintersdorf, Lukas Struppek, Patrick Schramowski, and Kristian Kersting. SEGA: Instructing text-to-image models using semantic guidance. *arXiv preprint arXiv:2301.12247*, 2023. URL https://arxiv.org/abs/2301.12247.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics*, 42(4), 2023. URL https://arxiv.org/abs/2301.13826.

Junhao Chen, Chanyu Zhu, Zheqi Lv, Keting Yin, and Shengyu Zhang. Training-free semantic correction for autoregressive visual models. *arXiv preprint arXiv:2606.22550*, 2026. URL https://arxiv.org/abs/2606.22550.

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 Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2310.11513.

Shelly Golan, Yotam Nitzan, Zongze Wu, and Or Patashnik. Vlm-guided adaptive negative prompting for creative generation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=JzA6d2II4Q.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Susung Hong, Chongjian Ge, Zhifei Zhang, and Jui-Hsien Wang. DiffusionBrowser: Interactive diffusion previews via multi-branch decoders. *arXiv preprint arXiv:2512.13690*, 2025. URL https://arxiv.org/abs/2512.13690.

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*, 2023.

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. URL https://arxiv.org/abs/2512.16853.

Eran Levin and Ohad Fried. Differential diffusion: Giving each pixel its strength. *arXiv preprint arXiv:2306.00950*, 2023. URL https://arxiv.org/abs/2306.00950.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. *arXiv preprint arXiv:2201.09865*, 2022. URL https://arxiv.org/abs/2201.09865.

Zheqi Lv, Junhao Chen, Qi Tian, Keting Yin, Shengyu Zhang, and Fei Wu. Multimodal LLM-guided semantic correction in text-to-image diffusion. *arXiv preprint arXiv:2505.20053*, 2025. URL https://arxiv.org/abs/2505.20053.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. URL https://arxiv.org/abs/2108.01073.

Pinelopi Papalampidi, Olivia Wiles, Ira Ktena, Aleksandar Shtedritski, Emanuele Bugliarello, Ivana Kajic, Isabela Albuquerque, and Aida Nematzadeh. Dynamic classifier-free diffusion guidance via online feedback. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.16131.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. GenArtist: Multimodal LLM as an agent for unified image generation and editing. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2407.05600.

Tsung-Han Wu, Long Lian, Joseph E. Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting LLM-controlled diffusion models. *arXiv preprint arXiv:2311.16090*, 2023. URL https://arxiv.org/abs/2311.16090.

Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. BoxDiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2307.10816.

Yilun Xu, Mingyang Deng, Xiang Cheng, Yonglong Tian, Ziming Liu, and Tommi Jaakkola. Restart sampling for improving generative processes. *arXiv preprint arXiv:2306.14878*, 2023. URL https://arxiv.org/abs/2306.14878.

A Recovered Settings and Remaining Provenance

Generation settings. The recorded controlled runs use the original text-to-image checkpoint `Qwen/Qwen-Image` (not Qwen-Image-Edit), 512×512 images, 50 main steps, and CFG scale 4. Additional repaint and outside-reference evaluations are not included in the 50-step count. Experiment 2 contains historical runtime variants rather than one frozen implementation.

The archived GPT-5.5 paired preset pins `Qwen/Qwen-Image` to revision `75e0b4be04f60ec59a75f475837eced720f823b6`. This recovered revision is not assumed to identify every historical run.

For Experiment 1A, the saved runner uses scheduler-restart repaint with a 16-step restart budget (minimum 8), repaint guidance strength 1, mask blur 0.35, no mask dilation, and outside-region locking. Push and suppress use mask blur 1.5 and dilation 1 in the runner's mask units. The strength ceiling is 12. These values describe this runner, not all historical branches.

Table 8: Recovered settings. Fractions describe remaining main steps, not noise levels. Missing checkpoint and runtime revisions remain unresolved.

Study	Inspection fractions	Intervention settings
1A	Repaint: 0.82; push/suppress: 0.70	Span 0.45; cutoff 0.30; strengths 0.95/10/6 for repaint/push/suppress.
1B	Initial: 0.82, 0.66, 0.50; selected: 0.78, 0.70, 0.62	One call per inspection; span 0.18 (initial), 0.45 (selected and held-out); cutoff 0.30.
2, GenEval2	Approximately 0.857, 0.735, 0.633	No clean-tail cutoff or combined-delta cap.
3	0.86, 0.68, 0.50, 0.32	Four requested inspections; retries kept separate.

Interactive archive completeness. Of 298 saved branches, 289 are marked completed, seven cancelled, one queued, and one has a final image but no matching branch-status record. These counts are distinct from the 194 intervention attempts and 247 unique judged images. Missing records are not reconstructed as successful actions. Historical budget summaries use earlier unblinded labels, not the new blinded evaluation.

Historical checkpoint and code revisions are not established for every run. The archives contain session records, configurations, traces, and available source hashes, but current code cannot establish which implementation produced an older image. In particular, historical suppress and repaint-removal variants are kept distinct. Complete hardware, precision, and timing provenance is unavailable across all arms. The submitted supplementary is a separately sanitized package with verified file hashes, saved-score analysis, linked Experiment 2 images, selected Experiment 3 images and traces, and matched Gemma runtime/status records. It also includes the schedule and regeneration/text-control evidence. The full research archives are not released; complete generation and training reproduction and third-party rights review remain outside the verified scope.

A.1 Additional experimental protocol

Tool-study tuning and settings. The development manifests contain 72 distinct prompts with no overlap between calibration and test cases. After initial target success of 3/20 for attribute changes and 0/20 for suppression, strengths 3, 6, and 10 were tested on calibration cases before another test run. Repaint insertion used one calibration round. The test therefore informed development despite disjoint prompts. For 1A, repaint acts at 0.82 and push/suppress at 0.70, with span 0.45 and cutoff 0.30. Intended strengths are 0.95/10/6 for repaint/push/suppress; comparison push uses 0.8 and comparison repaint 1. High strengths exceed the guarded preset. Repaint uses a 16-step restart. Cases already satisfying the target at baseline are excluded. Claude crop ratings cover only the final v1 settings. Pixel-distance evaluations use original PNGs, not JPEG media copies.

Interactive sampling and provenance. The complex 50-prompt cohort has 13/12/13/12 prompts with 7/8/9/10 requirements, selected by alternating positions within strata of an existing 100-prompt pool. Recorded GenEval2 inspection fractions are approximately 0.857/0.735/0.633. The first cohort contains multiple historical runtimes; the cohorts are not pooled as one controlled comparison. Some cases exceed the initial two-attempt budget. The earlier audit was unblinded; new model ratings evaluate frozen historical selections. Inspector-unavailability no-ops are missing decisions. Saved parents, branch points, and timestamps support retrospective oracle best-k summaries under historical labels. Unknown order or verdicts require bounds; these summaries have not been recomputed with blinded ratings and do not estimate a precommitted attempt-limited policy. Training on the records requires provenance verification and separation from evaluation prompts.

Evaluator details. The oracle evaluator uses Qwen3-VL-8B-Instruct with Transformers 5.13.1 and PyTorch 2.11.0. Experiment 2 uses the unchanged official `soft_tifa` function with Qwen3-VL-8B-Instruct in bfloat16 and Transformers 4.57.0.[1] GM aggregates each prompt's atomic scores geometrically, AM arithmetically; both are then averaged over prompts. These evaluator stacks are

[1] https://github.com/facebookresearch/GenEval2

not assumed interchangeable. Gemini reevaluation uses the requested `gemini-3.1-pro` endpoint at high reasoning effort; the provider did not return a backend model identifier. Single-image requests contain only the target prompt and fixed rubric, with no method, prior verdict, trajectory, or selection history. Pair requests reveal reference/candidate roles but hide method and prior scores. No human calibration of the LPIPS threshold is available.

Table 9: Prompt-level resampling settings. Binary tests are two-sided exact McNemar tests. Confidence intervals are 95%.

Study	Paired percentile bootstrap	Resamples	Seed
1B	Soft-TIFA GM/AM	100,000	20260815
2	Soft-TIFA GM/AM	10,000	20260920
2	Strict PASS difference	20,000	20260920
3	Soft-TIFA GM/AM	10,000	20260815
3	Strict PASS difference	10,000	20260923

B Exploratory Tool Ablations

Earlier development-set checks. The test set informed the strength search in this study. Its target-success McNemar values are 0.03125 for insertion, 0.55 for removal with the recipe, and 1.0 for attributes; none survives Holm correction across the three tasks.

What changes in the removal recipe. Local-only branch conditioning describes the replacement background rather than repeating the full scene prompt, which may still name the unwanted object. Negative-only post-guidance uses a blank positive condition and the unwanted object as the negative condition in Equation 4. The original scene prompt still supplies the main prediction. These are options in the exploratory removal study, not settings shared by every repaint run in the paper.

Composite removal outcomes on development and fresh cases. Table 11 reports the conjunction of removal, absence of a judged patch, preservation, and locality. It differs from the partial criteria in the earlier ablation table below. The fill variants add a region-fill operation before a shorter restart; the labels identify these run variants and do not imply equivalence to changing the restart length alone.

All counts in Table 12 use the same exploratory set of 31 excess-instance scenes. A missing flat-patch count means it was not reported, not that the output was clean. Flat-patch absence is narrower than an assessment of all artifacts, and the rating provenance requires further verification.

The study reports outside-region locality in 31/31 cases with the lock versus 9/31 without it; this is not a claim of identical decoded pixels. Object-shaped masks performed worse than rectangular masks in this set. A plausible explanation is that the retained silhouette encourages another object of similar shape, but this mechanism is not established by a controlled causal test.

Transfer to oracle-assisted generation. On the 100 held-out prompts of Table 3, 225 atoms fail at baseline. Only 20 are excess-instance counts, occurring in 20 prompts. Changing removal has little effect on whole-run GM (Table 13). Global repaint-conditioning changes also affect non-removal calls; the per-call variant limits this interference. The suppress row is the frozen confirmation arm; the later removal variants are exploratory comparisons on the same held-out prompts.

Per-prompt scores, oracle verdicts, runtime records, and paired summaries for these arms are archived with the analysis scripts. They support the reported intervals; inclusion in the anonymous supplementary package requires a separate provenance and anonymization check.

Table 10: Separate checks on the earlier development set (v1), with 20 cases per task. Each column counts one check independently; the counts must not be added or read as joint success. Qwen3-VL assesses the edit and retention of previously satisfied prompt requirements. Outside-box similarity requires LPIPS ≤ 0.05; Claude rates visible artifacts in crops. These are model ratings, not human judgments. This is a different set from Table 2.

Requested edit	Tool	Edit achieved	Other targets kept	Outside box similar	No visible artifacts
Add an object	Repaint	**6/20**	**20/20**	**20/20**	18/20
	Push	0/20	**20/20**	**20/20**	**20/20**
Change an attribute	Push	12/20	15/20	2/20	**15/20**
	Repaint	**13/20**	**16/20**	**20/20**	14/20
Remove an object	Suppress	9/20	**19/20**	11/20	**13/20**
	Repaint, recipe	**12/20**	14/20	**20/20**	9/20

Table 11: Composite excess-instance removal. The fresh set is separate from the development set; selecting the best fresh-set variant would require another independent test. Any reported removal-variant p-values are nominal, without correction for multiple comparisons.

Variant	Development (31)	Fresh (34)
Suppress	3/31	2/34
Empty-background repaint, scene prompt	12/31	5/34
Background recipe, 16-step restart	14/31	11/34
Fill + 8-step restart	**15/31**	13/34
Fill + 4-step restart	**15/31**	14/34
Fill + 8-step restart + guidance away from the scene prompt	10/31	**15/34**

C Additional Protocol and Diagnostic Results

C.1 Retrospective inspector audit

Retrospective decision audit. We analyze the inspector's individual decisions rather than assigning a single failure label to an entire trajectory. The retrospective Astra audit assigns 73 of 800 inspection records an inspector decision-error label (Figure 2). Wrong bounding boxes (33) and missed errors (24) dominate this subset, followed by wrong tools (5), false diagnoses (5), bad parameters (1), and five other decision errors. These labels are diagnostic hypotheses, not recognition accuracy against human ground truth or verified causal explanations. Multiple records belong to the same trajectory and are not independent samples.

We separate 117 reasonable-but-ineffective action labels from inspector decision errors: an unsuccessful action alone does not establish that its choice was wrong. Another 42 records are labeled API failures, 132 successful correction, and 436 justified no-op. These labels are not final-image FAIL-to-PASS transitions.

The retrospective audit contains 200 trajectory-level attributions and 800 inspection-level labels. The figure uses the latter. These labels have not been independently checked against the full preview–call–execution–result sequence. Separate paired preservation ratings for Experiment 3 are unavailable.

C.2 Full study protocol

C.3 Experiment 1: tool capability and oracle-assisted diagnosis

1A: prescribed calls. We test repaint for insertion, push for attributes, and suppress for removal. Each case also uses an alternative: push for insertion and repaint for the other tasks. **No online inspector chooses the call.** OWLv2 or a free-space heuristic supplies the box, and the edit is specified before generation. Removal and half of the attribute cases are constructed edits rather than naturally occurring failures. The original removal control is empty-background repaint with the full

Table 12: Exploratory removal ablations. Variants were investigated on the same cases; the selected recipe is not independently validated by this table.

Removal setting	Excess removed	Removed, no flat patch
Suppress	12/31	–
Repaint, scene prompt in branch	14/31	–
Repaint, local-only empty background	22/31	–
Local background description	22/31	15/31
Negative-only post-guidance (recipe)	**24/31**	18/31
Recipe + outward feather (48 px)	22/31	15/31
12-step restart	20/31	15/31
8-step restart	13/31	12/31
Guidance ends at 0.50	21/31	16/31
No post-guidance	22/31	15/31
Later repaint at 0.50	14/31	10/31
No outside lock	**24/31**	**19/31**
Object-shaped SAM mask	9/31	8/31
Second repaint pass	21/31	15/31

Table 13: Removal variants on 100 held-out prompts. GM gains and paired intervals cover the whole run. Excess fixed counts failed one-too-many atoms that become passing. Other broken counts previously passing atoms that fail in the 20 prompts with an excess-instance error, not across all 100 prompts.

Oracle removal variant	ΔGM	95% paired CI	Excess fixed	Other broken
Suppress	**+2.87**	[0.50, 5.68]	3/20	**0**
Repaint, global local-prompt mode	+2.71	[−0.08, 5.65]	8/20	9
Global mode + area cap 0.25	**+2.87**	[0.36, 5.55]	**9/20**	9
Local-only mode on removal calls	+2.66	[0.16, 5.57]	6/20	1

scene prompt. A later test adds the pre-existing removal recipe on identical cases, seeds, and boxes (Appendix C.12).

The development study (v1) has four calibration and 20 test cases per task. Although its 72 prompts are distinct, the test informed further strength tuning. The confirmation study (v2) uses a separate 200-prompt pool and two seeds, with **settings frozen before generation**. Four already-correct attribute cases are excluded, leaving 80/76/80 cases. Both pools use 3–6 atomic requirements and have no documented overlap with each other or the oracle sets. Exact settings and tuning history are in Appendix A.1.

1B: supplied error labels. The planner receives failed atoms from the completed paired baseline, then chooses tools with OWLv2 or heuristic boxes. This isolates neither perfect localization nor autonomous error detection. On 100 development prompts (937 atoms), we compare timing and call-selection variants and select inspections 0.78/0.70/0.62 with span 0.45 by GM. We then draw **100 held-out prompts** (924 atoms; 25 per level of 7–10 requirements) and run the selected settings once, unchanged. Prompt text and source indices do not overlap with development. All arms allow one call per inspection and use cutoff 0.30; initial arms use 0.82/0.66/0.50 and span 0.18.

Dedicated removal recipe. We develop background-repaint recipes on 31 excess-instance scenes and test the frozen 16-step recipe against suppression on 34 fresh cases. These come from 200 previously unused prompts with 7–10 requirements and four seeds. Other recipe comparisons remain exploratory (Appendix B).

C.4 Experiment 2: interactive trajectory correction

Humans and an agent inspect previous previews and final images, then revise **boxes, tools, edit text, and strength**, with automatic VLM suggestions. The debugger records branches, calls, and

Table 14: The evaluated action basis. Each active call includes a normalized box, semantic condition, temporal window, and a preservation clause.

Action	Intended capability
`cfg_push`	Strengthen a visible object's attribute, material, texture, illumination, or local relation without resetting its structure.
`cfg_repaint`	Re-noise one bounded region and guide its regrowth to insert, replace, or reconstruct local content.
`cfg_suppress`	Weaken a localized unwanted object or attribute while preserving surrounding content.
`no_op`	Preserve the current trajectory when it is correct, ambiguous, too early to judge, or unsafe to modify.

Table 15: Evaluation criteria. Object substitution can pass a removal question; low LPIPS does not imply identical pixels. Soft-TIFA measures alignment, not scene preservation or artifact freedom.

Measure	Evaluator	Criterion and scope
1A target	Qwen3-VL-8B	Renormalized yes/no probability: ≥ 0.5 for insertion/attributes; < 0.5 for presence of the object to remove.
1A preservation	Qwen3-VL-8B	Every non-target requirement passing at baseline must remain passing.
1A locality	Outside-box LPIPS	Distance ≤ 0.05 after expanding the box by 0.03 and replacing its interior with baseline pixels.
1A composite	Checks above + crop rater	v2 requires target, preservation, and locality. v1 also requires a clean Claude crop rating; v2 has removal-specific patch checks, not general artifact ratings.
2/3 strict PASS	Gemini 3.1 Pro / Astra	Prompt compliance and no judged artifacts; each image judged separately without method identity.
2 preservation	Gemini pair review	Unrelated changes and new artifacts; no edit mask or action instruction is shown.
Soft-TIFA	Official GenEval2 evaluator	GM and AM over soft atomic scores, then averaged across prompts; reported on a 0–100 scale.

outcomes. Authorship of individual settings was not consistently recorded. The mixed cohort covers attributes, counts, contact, geometry, lighting, and unwanted content; the GenEval2 cohort has 7–10 requirements and was already used during development.

We report **retrospectively selected finals**, with variable attempt budgets and baseline fallback. Selections were frozen before the new blinded evaluation; its verdicts did not select the outputs. This measures interactive search, including feedback from earlier attempts. Some runs permit near-full-frame repaint and disable the guarded preset's cutoff and delta cap, so successful correction need not be spatially local. Historical runtime variants are kept separate. Sampling details and archive limits appear in Appendix A.1.

C.5 Experiment 3: automated VLM inspection

The `gpt-6-astra` inspector acts at remaining-step fractions 0.86/0.68/0.50/0.32. The 200 diagnostic prompts produce **400 paired finals and 800 inspection records**. We score original outputs only; archived retries do not replace them. Failed requests are missing decisions, not valid `no_op` calls.

Separate method-blind Astra calls judge the finals. Sharing a model family leaves possible inspector–judge bias. A retrospective Astra audit labels individual inspection decisions; these are **unverified diagnostic hypotheses**, not human-confirmed causes. Pairwise preservation ratings and human agreement are unavailable. Historical runtime provenance and exact serialized inspector inputs remain incomplete.

Table 16: Separate checks on the confirmation set, using the same cases as Table 2. Columns count each check independently, not joint success. Outside-box similarity requires LPIPS ≤ 0.05. These are model-based assessments; general artifact ratings are unavailable. Removal uses the later same-case recipe test.

Edit	Tool	Edit achieved	Other targets kept	Outside box similar
Insertion	Repaint	**23/80**	**77/80**	**80/80**
	Push	3/80	75/80	71/80
Attribute	Push	24/76	63/76	14/76
	Repaint	**37/76**	**64/76**	**76/76**
Removal	Suppress	**49/80**	**75/80**	47/80
	Repaint, recipe	44/80	70/80	**80/80**

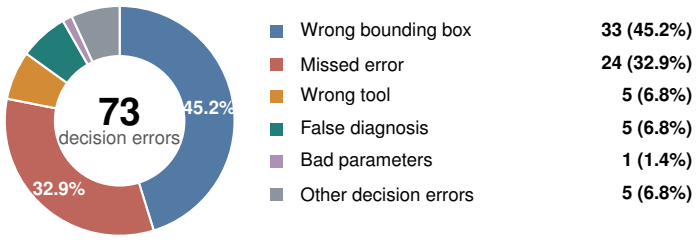


Figure 2: Inspector decision-error labels from the retrospective Astra audit. The denominator is 73 error-labeled inspection records out of 800, not 200 trajectories or failed final images. Other errors comprise wrong timing, ignored feedback, invalid action, wrong priority, and conflicting actions (one each). Reasonable but ineffective actions (117) and API failures (42) are excluded, as are success (132) and justified no-op (436) labels. Labels are model-generated hypotheses, not human-verified causal attribution.

C.6 Pilot schedule study: detectability, actuation, and sweep

The pilot of Section 6.3 ran on FLUX.2-klein-base-4B at 512×512 with 28 steps, a KIE-hosted GPT-5.5 inspector in direct tool-call mode, and detector-based GenEval scoring, so no LLM judge is involved. **Detectability:** 200 baseline generations, the $\hat{x}_0$ preview decoded at 15 steps, one neutral question derived from the GenEval metadata ("is this layout on track to satisfy the requirement?"), and the signed confidence scored by AUC against the detector's verdict on the finished image of the same run. At a fixed threshold the early points look different but are misleading: at 0.96 the inspector flags 95% of healthy runs, so it is refusing to commit rather than detecting early. **Actuation:** the same scripted call ("a large bright red apple in this box") injected at ten steps on 24 prompts with either tool, judged by the same detector; 0.04 is its false-positive rate on untouched images. The drop between 0.56 and 0.44 is the controller's late-repaint cutoff (0.45), below which a repaint is downgraded to a push. **Sweep:** eight schedules on the 60 discriminative GenEval prompts (45 contested, passing 1–3 of 4 baseline seeds; 15 hard, passing 0 of 4) with the two baseline seeds; the 322 always-passing prompts cannot separate schedules and were skipped. The baseline pass rate on this subset is 0.392.

For the best schedule, the per-category gains are position $+0.167$, two-object $+0.188$, color attribution $+0.056$, and counting 0.000. Reading the inspection logs against the baseline verdict of the same (prompt, seed) pair, the inspector acts on 0.27/0.32/0.35/0.32/0.35 of the runs that go on to fail at fractions 0.89/0.85/0.74/0.63/0.56, and on 0.13/0.18/0.10/0.06/0.00 of the runs that pass. A scripted always-fire probe puts the diffusion-side cost of a point at about 0.6 s (1.64 s per image with no points, 3.32 s with three late points, 5.19 s with six early points), against 20–26 s per inspector call. The pilot uses one generator, one inspector, and one benchmark subset on which the schedule was both tuned and validated; a held-out check on the Qwen-Image stack of Experiments 1–3 has not been run.

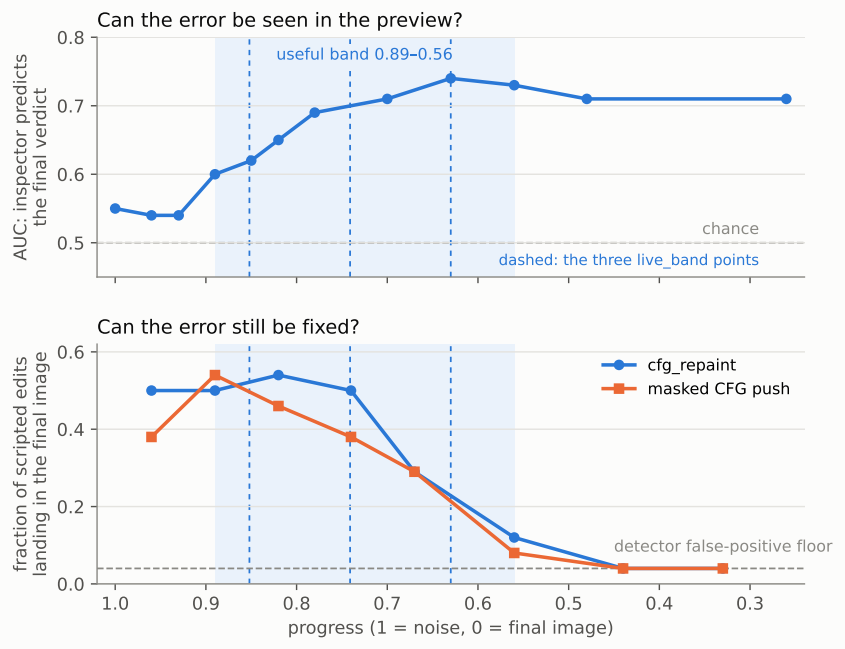


Figure 3: Pilot schedule study: the useful inspection window. Top: AUC with which the inspector's verdict on the intermediate preview predicts the detector's verdict on the finished image; above remaining-step fraction 0.89 it is at chance. Bottom: fraction of scripted edits (24 prompts, no VLM) that the detector finds in the final image; from 0.44 on it equals the detector's false-positive floor. Shaded: the 0.89–0.56 band. Dashed: the three points of the best schedule.

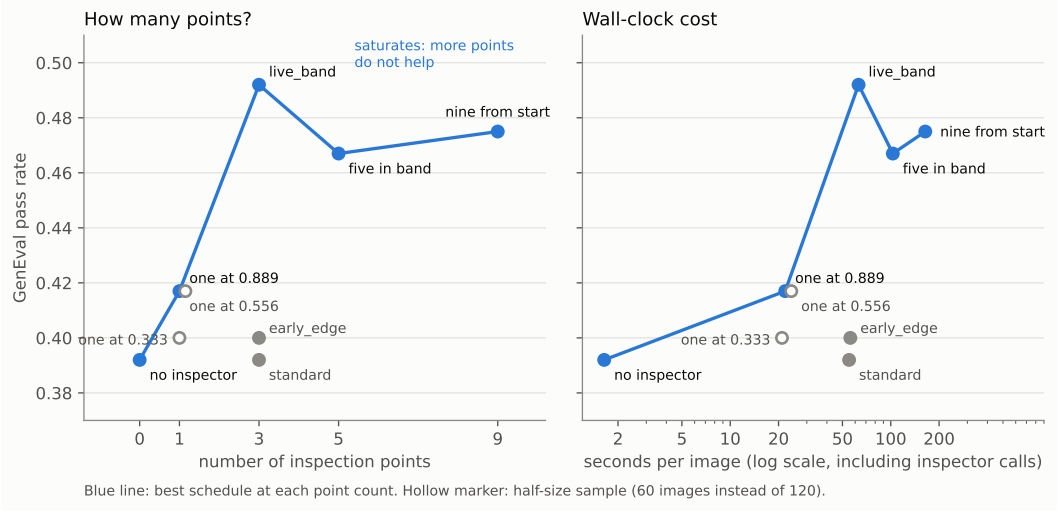


Figure 4: Pilot schedule study: pass rate against the number of inspection points (left) and against wall-clock per image including inspector calls (right). The line joins the best schedule at each point count. Hollow markers are half-size samples (60 images), which overstated single-point gains before settling at 120.

Table 17: Pilot schedule sweep on 60 discriminative GenEval prompts × 2 seeds (baseline pass rate 0.392). Δ is the paired difference in pass rate; p is the two-sided exact McNemar value; s/img includes inspector latency (20–26 s per call). Rows with $n = 60$ are half-size and not comparable in power.

Schedule	Points	n	Pass	Δ	F→P / P→F	p	s/img
live band	0.852 / 0.741 / 0.630	120	**0.492**	+0.100	15 / 3	0.008	63
nine from start	0.96 . . . 0.32	120	0.475	+0.083	13 / 3	0.021	164
five in band	0.889 . . . 0.593	120	0.467	+0.075	10 / 1	0.012	103
one at 0.556	0.556	60	0.417	+0.033	2 / 0	0.50	24
one at 0.889	0.889	120	0.417	+0.025	5 / 2	0.45	22
one at 0.333	0.333	60	0.400	+0.017	1 / 0	1.0	**21**
early edge	0.889 / 0.815 / 0.741	120	0.400	+0.008	5 / 4	1.0	56
standard	0.72 / 0.54 / 0.36	120	0.392	0.000	3 / 3	1.0	55

C.7 Evaluation

Table 15 separates edit success, artifacts, preservation, and prompt alignment. Model ratings supply the quantitative outcomes. Author inspection during development was informal and unblinded; **no independent human calibration** or inter-rater agreement is available.

For Experiment 2, Gemini reevaluates 247 unique images and 186 branch pairs; all responses pass schema validation and image hashes identify the inputs. Ambiguous image judgments count as non-PASS. Pair ratings may mistake intended changes for unrelated ones, and multiple branches can share a prompt. Official Soft-TIFA scores the 50 GenEval2 baseline–selected pairs (954 VQA scores), without new generation or reselection. Experiment 3 uses 100 GenEval2 pairs and 937 atoms.

We use **paired prompt-level bootstrap intervals** for score differences, exact McNemar tests for paired binary outcomes, and Wilson intervals for 1A proportions. Updated tool comparisons apply Holm correction across three tasks, using the removal recipe as comparator; original-control tests are archived. Resampling settings and evaluator versions are listed in Appendix A.1. Intervals condition on the observed data; they do not account for adaptive selection or evaluator error.

C.8 Inspector request and execution details

In the single-VLM implementation, each request contains the original prompt, one current RGB preview, the current step fraction, the inspection schedule, the available tool schemas, and a short text history. The history lists active interventions, up to eight recent requested calls, and up to four recent findings. It does not include a sequence of earlier preview images. The system instructions ask the inspector to check the visible scene against the prompt, identify a local error, and choose an edit or `no_op`.

The code reports the remaining step fraction $p_j = 1 - j/(N-1)$, which decreases from one at the beginning to zero at the last main step (note that p_j is distinct from the scheduler noise level σ_j). A requested inspection fraction is mapped to a main step, and traces record the actual fraction used. Because the main step is executed before a new report is handled, a new push or suppress call first affects a later main step, while a repaint starts a branch that rejoins at σ_{j+1}.

What the inspector returns. The response is a structured report containing findings and tool calls. An active call specifies a normalized box $[x_1, y_1, x_2, y_2]$, text describing the change, a preservation clause, and an active interval. Coordinates run from zero to one, with the origin at the top left. Push and suppress also expose a strength. In the current live schema, repaint noise and guidance strengths are runtime settings, not VLM decisions; older replayed calls can specify them explicitly. The controller validates calls and applies the configured timing, strength, and region restrictions before execution.

C.9 OUTSIDE-REGION REFERENCE LOCK

When `local_repaint_lock_outside` is enabled, the runner also continues a reference trajectory from the latent immediately before the first repaint. At each remaining main step it restores the outside region from that reference. For multiple repaint boxes, the lock uses their soft union. This option adds reference-model evaluations. It constrains latent values where the mask is zero, but does not guarantee identical pixels outside the box after decoding.

C.10 INTERACTIVE ARCHIVE COVERAGE

4. What the trajectories show. The archive contains 100 cases, 154 sessions, and 298 branches. Attempts can combine several tools, so a successful branch does not establish which call caused the improvement. The 39-example gallery was selected under an earlier unblinded audit; it illustrates trajectories rather than providing an additional success count. One final lacks a branch entry, and some repaint frames may have been overwritten. Videos are partial records, not proof of complete call coverage. Reported box sizes describe the largest requested region, not the actual pixel changes.

C.11 WHERE THE INTERACTIVE GAIN COMES FROM

The tool texts of the 113 GenEval2 attempts are full scene descriptions: the positive texts of an attempt add a median of 140 words (up to 281), every attempt also carries a negative list, and 54 attempts request a box covering at least 90% of the image. We therefore measure two controls on the same 50 prompts and seeds (Table 18). The first draws other seeds at the per-prompt budget of Experiment 2, counted as images (one plus the attempts) or as recorded generator compute, and keeps one image. The second regenerates each of the 112 attempts with applied tool calls from the case's seed, with the attempt's positive texts appended to the prompt and its negative texts as the negative prompt. It uses no preview, inspection, or in-trajectory edit, so the texts act on the whole image from the first step.

Table 18: Where the Experiment 2 gain comes from. ΔGM is official Soft-TIFA GM against each method's own seed-0 baseline: 18.76 for Experiment 2 and 20.60 for the controls, on the same prompts and seeds but from different evaluation runs. *Same texts, no edit*: each attempt regenerated from its seed with its tool texts in the prompt, without a trajectory edit. *Judge selects*: the highest official score is kept, in hindsight; Gemini sees only the prompt and the images. Seeds are capped at 8 per prompt, which favors Experiment 2. Paired bootstrap over prompts, 100,000 resamples.

Method (50 complex GenEval2 prompts)	Images/prompt	ΔGM	95% CI
Experiment 2, selected finals	3.26	$+28.90$	[18.56, 39.65]
Other seeds, same images, judge selects	3.26	$+10.81$	[6.72, 15.88]
Other seeds, same compute, judge selects	≤ 8	$+14.38$	[9.08, 20.47]
Other seeds, same images, Gemini selects	3.26	$+5.90$	[2.17, 10.89]
Same texts, no edit, Experiment 2's pick	3.24	$+17.57$	[7.55, 28.26]
Same texts, no edit, judge selects	3.24	$+34.58$	[24.08, 45.50]

Keeping the attempt that Experiment 2 kept favors Experiment 2, whose picks were made by looking at its own edited images; judge selection favors the control, because the judge is the metric. The interactive gain of +28.90 lies between the two, so the texts alone account for most of it, and these data cannot show whether applying them inside the trajectory adds to it. A single control image gains +24.45 GM over its prompt's baseline on average. Starting from the partly correct preview can keep what is already right: in one case the baseline shows the five requested rabbits (GM 64.3), regenerating with the attempt's texts yields three (GM 5.1), and Experiment 2 kept its edited branch over that baseline. The archive has no per-branch images or official scores, so this is not measured beyond single cases. Gains are not paired between Experiment 2 and the controls, because the baselines differ.

C.12 REMOVAL AGAINST THE ORIGINAL PROMPT

The 1A removal task asks for deletion even though the original prompt requests the object. A later same-case test adds a recipe fixed before the confirmation pool existed: local-only branch conditioning with a model-written surroundings description, the object as negative condition, negative-only post-guidance, a 16-step restart, and outside lock. It uses the same cases, seeds, boxes, trigger 0.70, and span 0.45 as suppression. This is a follow-up comparison, not the originally planned empty-background control.

Table 19: Removal on the same 80 constructed confirmation cases. Gone is the whole-image target judgment; Background and Other are crop-based labels. Composite is target success, semantic preservation, and outside-box LPIPS ≤ 0.05. Clean additionally requires background only and no flat patch. Counts are model judgments, not independent human assessments.

Arm	Gone	Background	Other	Patch	Composite	Clean
Suppress	**49**	0	48	10	24	0
Naive repaint	9	2	6	21	8	1
Removal recipe	44	**8**	34	41	**37**	2
Recipe + fill, 8 steps	26	5	21	10	23	**4**

The crop evaluator asks whether the named object is present and whether only empty background remains. Other means both answers are negative, not necessarily a verified substitution: legitimate scenery can also fail the strict background question. Whole-image and crop judgments can disagree. Model-written surroundings may name neighboring objects, potentially conditioning their appearance inside the edit. These limitations preclude treating the crop labels as exact error rates.

The recipe preserves outside-box similarity in 80/80 cases versus 47/80 for suppression, but its higher three-check composite is not statistically resolved (exact McNemar $p = 0.079$). Clean removal is 2 versus 0 ($p = 0.50$). Fill reduces patches but often retains the object. On the 20 development cases, the recipe achieves target/preservation/locality/crop-cleanliness counts 12/14/20/9; its four-check composite is 1/20, equal to suppression. The newer clean-removal endpoint additionally requires background-only and passes 0/20 for both. These endpoints must not be conflated.

For transparency, the original confirmation comparison against naive repaint remains 49 versus 9 target successes and 24 versus 8 three-check successes. The updated three-task Holm family instead compares suppression with the removal recipe; adjusted values therefore change for insertion and attributes as well. The excess-instance study separately tests deletion consistent with the prompt and retains its 19/34 versus 5/34 removal result.

D ADDITIONAL BACKENDS AND PRELIMINARY INSPECTOR DISTILLATION

FLUX implementation. The repository also contains a custom denoising runner built on `Flux2KleinPipeline`. It decodes intermediate clean-image previews, requests structured inspector actions, and applies interventions within the sampling loop. This is implementation evidence for a second generator backend, not a controlled cross-generator evaluation. The historical FLUX action schema and runtime differ from the Qwen implementation evaluated here; we do not claim that all current tools transfer unchanged or retain their measured quality.

Historical Gemma pilot. Saved configuration and training metrics document LoRA training of `unsloth/gemma-4-E4B-it` on teacher-generated inspector decisions. The configuration uses rank 16, alpha 16, zero dropout, language-layer linear modules with the vision layers frozen, bfloat16, learning rate 10^{-4}, and an effective batch size of 16. Training lasted three epochs (3,658 seconds). The configured sampler draws 2,048 examples per epoch, with 70/30 nine-/four-inspection examples and 40/60 action/no-op examples; these sampling counts do not establish the number of unique training examples.

A saved 40-case evaluation reports strict success of 18/40 for the baseline, 21/40 for Gemma-LoRA, and 25/40 for the GPT-5.5 teacher. Gemma used four inspections and Qwen CPU offload; the teacher used nine inspections. Finals were rated by separate GPT-5.5 calls. These are historical

diagnostic results, not a matched teacher–student comparison. There is no paired base-Gemma arm, and the historical holdout's independence from all development and training data has not been re-established. Thus, the pilot demonstrates that a trained inspector can execute the interface, not that fine-tuning improves correction. Training-data quality has not been isolated as a cause of its errors.

Recorded runtime. Table 20 summarizes saved elapsed-time fields, including inspection overhead, from two separate archives. The Gemma archive contains 400 paired traces; Astra uses the 200 original Experiment 3 pairs, excluding later retries. Both record Qwen-Image at 512×512, 50 main steps, CFG 4, bfloat16, no Qwen CPU offload, and four requested inspections at 0.86/0.68/0.50/0.32. Gemma 4 E4B-IT + LoRA is served locally through vLLM; GPT-6-Astra (requested model ID `gpt-6-astra`, reasoning effort `high`) uses a remote API. Prompt sets, hardware provenance, tool runtimes, and execution conditions are not matched. In particular, their baseline times differ substantially. These measurements must not be read as a controlled model-speed comparison.

Table 20: Mean recorded seconds per image in separate historical runs. The Gemma base model is `unsloth/gemma-4-E4B-it`; high denotes GPT-6-Astra reasoning effort. Inspector time is the recorded cumulative inspector latency within a corrected run. Ratio is mean corrected time divided by mean baseline time within each archive. It is not a Gemma-to-Astra speedup. Subtracting inspector latency does not yield pure GPU compute time.

Inspector / archive	Pairs	Baseline (s)	Corrected (s)	Inspector (s)	Ratio
Gemma 4 E4B-IT + LoRA	400	13.09	41.23	18.82	3.15×
GPT-6-Astra (high)	200	39.34	250.86	168.52	6.38×

Matched compact-inspector pilot. We subsequently compared base `unsloth/gemma-4-E4B-it` with its historical LoRA adapter on the same 200 diagnostic prompts (100 general, 100 GenEval2). Both use Qwen-Image, 512×512, 50 main steps, CFG 4, four inspections at 0.86/0.68/0.50/0.32, and local bfloat16 serving on A100 GPUs. Baseline image hashes match for all 200 pairs. Separate blind GPT-6-Astra (high) calls rate each image; the baseline verdicts are reused. Official GenEval2 evaluation uses Qwen3-VL-8B-Instruct and 937 atoms on the 100 GenEval2 prompts.

Table 21: Matched diagnostic pilot. Soft-TIFA applies only to the 100 GenEval2 prompts. These are reused diagnostic prompts, not an established held-out training evaluation.

Arm	General PASS	GenEval2 PASS	Soft-TIFA GM	Soft-TIFA AM
No intervention	**72/100**	25/100	19.49	75.45
Base Gemma	**72/100**	**28/100**	18.62	74.70
Gemma-LoRA	70/100	21/100	**24.11**	**75.67**

Partial alignment, not reliable correction. LoRA exceeds Base Gemma by 5.50 GM points (paired 95% bootstrap CI [0.20, 11.07]), but strict success falls from 100/200 to 91/200. There are 18 fixes and 27 regressions relative to Base Gemma ($p = 0.233$, exact McNemar). The corresponding strict difference is -4.5 percentage points (CI [-11.0, 2.0]); AM changes by 0.97 points (CI [-1.25, 3.06]). We use 10,000 prompt-level bootstrap resamples. The GM comparison is exploratory and unadjusted for multiple comparisons.

Response-format confound. Of 800 Base Gemma inspections, 279 fail JSON parsing and another 181 have no executable tool call; LoRA has two such empty reports. Empty reports are not explicit no-op decisions. Full raw responses were not retained, so the latter failures cannot be assigned a precise cause. This comparison measures format compliance, action selection, and tool execution together; it does not isolate a benefit in decision quality. Training/evaluation separation also remains unverified.

A cleaner follow-up should enforce the same valid-output protocol for both inspectors and evaluate prompt families excluded from training and development. Faster inspector responses alone do not imply faster corrected generation.

Table 22: Mean recorded runtime across 200 paired cases per arm. LoRA reduces cumulative inspector latency but increases total generation time through more tool activity. Extra conditioning evaluations count generator work beyond the ordinary baseline; subtracting inspector latency does not isolate pure GPU compute.

Inspector	Baseline (s)	Corrected (s)	Inspector (s)	Extra conditioning evals
Base Gemma	29.40	**114.63**	68.36	**20.23**
Gemma-LoRA	29.35	152.16	**39.71**	138.03