VISUALREFINEBENCH: CAN UNIFIED MULTIMODAL MODELS RELIABLY REFINE THEIR OWN IMAGES?

Anonymous authors
Paper under double-blind review

ABSTRACT

Unified multimodal models (UMMs) can inspect and revise their own images at test time, but their gains in final image quality remain unclear. We introduce VISUALREFINEBENCH, combining end-to-end evaluation with three diagnostic tasks across 800 examples. Refine-E2E measures initial-to-delivered improvement and records the best image score during refinement. Under fixed inputs, Reflection tests error diagnosis, Decision tests action choice, and Editing measures net gains from instructed edits. With a four-image-call budget, all six evaluated UMMs deliver lower mean quality than their corresponding Oracle best-of-N baselines. Diagnostics reveal missed errors, poor action choices, and damage to correct content that often offsets repair gains. Replacing the Controller with a model stronger in Reflection improves mean final quality by about 17 points across four generator–editor combinations. These gains primarily reflect higher-quality candidates produced during refinement, while reliable final-image selection remains a challenge. These findings identify opportunities to improve UMM self-refinement and show how diagnostic evaluation can guide component choices that improve end-to-end outcomes.

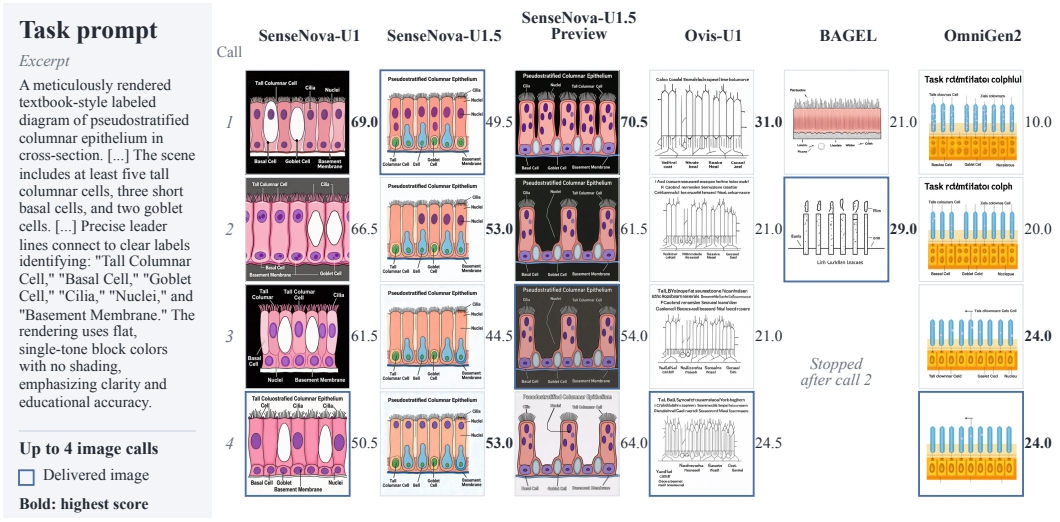


Figure 1: Examples of generated images from self-refinement runs on the same prompt. Each model occupies one column, with calls 1–4 ordered from top to bottom. Scores appear beside each image. Blue outlines mark the final image determined by the unified model; bold scores mark the trajectory maximum. Scores use the recorded rubric ($\times 100$). BAGEL stops after call 2.

1 INTRODUCTION

Modern text-to-image models follow increasingly open-ended visual instructions. Refining images to meet these instructions closely links image understanding, generation, and editing. Unified multimodal models (UMMs) natively support all three capabilities within a single language–image ar-

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.

chitecture (Wu et al., 2026; Deng et al., 2025). Two common approaches use additional test-time computation to improve image quality. Best-of-N samples several independent images, and an external verifier or human reviewer selects one (Ma et al., 2025; Jain et al., 2026). The other approach, *image self-refinement*, lets a model inspect and revise its images over multiple steps (Madaan et al., 2023; Yang et al., 2024; Zhuo et al., 2025). Each step uses earlier images to guide the next generation or edit. These integrated capabilities allow a UMM to carry out self-refinement and choose which image to deliver. How much this process improves final image quality remains unclear.

Prior benchmarks evaluate reflection, rectification, and verification in image generation and study their effects on image quality (Wang et al., 2026; Zhang et al., 2026). Yet no single protocol answers two questions central to autonomous image self-refinement.

> ***RQ1:*** *How much can current UMMs* ***improve the images they deliver*** *through self-refinement under a fixed image-operation budget?*
> ***RQ2:*** *How well do current models perform the* ***error diagnosis, action choice, and instructed editing*** *needed for image self-refinement?*

We introduce VISUALREFINEBENCH to answer these questions. Its four tasks contain 200 examples each. REFINE-E2E evaluates the complete SELFREFINE process under a fixed image-operation budget. It measures how much the submitted image and the best image produced during refinement improve over the initial image. ORACLE-BON provides a matched-budget reference, while the external Judge remains outside each trajectory. We further assess this process through three diagnostic tasks with fixed inputs. Reflection asks a model to identify and describe errors in an image relative to its prompt. Decision tests action choice among stopping, following a given repair plan, and returning to an earlier image, using fixed image history, error clues, and the remaining budget. Editing provides a repair instruction and measures the net improvement from one edit, accounting for both corrected errors and damage to previously correct content.

We evaluate six UMMs across all four tasks. In REFINE-E2E with a limit of four image calls, three models show clear mean gains from their initial to their delivered images, while the other three show little change. All six deliver lower mean scores than ORACLE-BON. Some models deliver a lower-scoring image than one they produced earlier in the same run. We also evaluate 18 additional vision-language models (VLMs) on Reflection and Decision and 12 additional image generation and editing models on Editing. Reflection shows that models often miss image errors. In Decision, some models repeatedly choose to continue or stop when another action is appropriate. In Editing, damage to correct content often cancels out repair gains. We then run component replacement experiments to test whether models with better diagnostic scores improve final image quality. In these experiments, the VLM diagnoses errors, plans repairs, and chooses actions. Replacing either the VLM or the editor improves final image quality, with the largest gains from the VLM among the tested changes. These results support using diagnostic tasks to improve end-to-end self-refinement.

2 VISUALREFINEBENCH

This section presents the four tasks in VISUALREFINEBENCH and their protocols (Figure 2). We first describe dataset construction (Section 2.1) and define the inputs and scoring criteria for each task (Section 2.2). Section 2.3 introduces REFINE-E2E to evaluate autonomous image self-refinement. Section 2.4 presents three diagnostic tasks for selected stages of this process under fixed inputs: Reflection for error diagnosis, Decision for action selection, and Editing for instructed image editing.

2.1 DATASET CONSTRUCTION

We use DeepSeek-V4-Pro (DeepSeek-AI, 2026) to expand domain-organized seed material into requests covering scientific charts, textbook and technical diagrams, illustrations, and visual design. GPT-5.5 (OpenAI, 2026a) generates a list of scoring requirements (a rubric) for each request. Human reviewers check these requirements. Reflection and Editing use GPT-Image 2 (OpenAI, 2026b) images with local errors, usually one or two unmet requirements. GPT-5.5, the Codex harness (Bolin, 2026), and human review verify errors and produce annotations and repair instructions, used as evaluation references in Reflection and model-visible instructions in Editing.

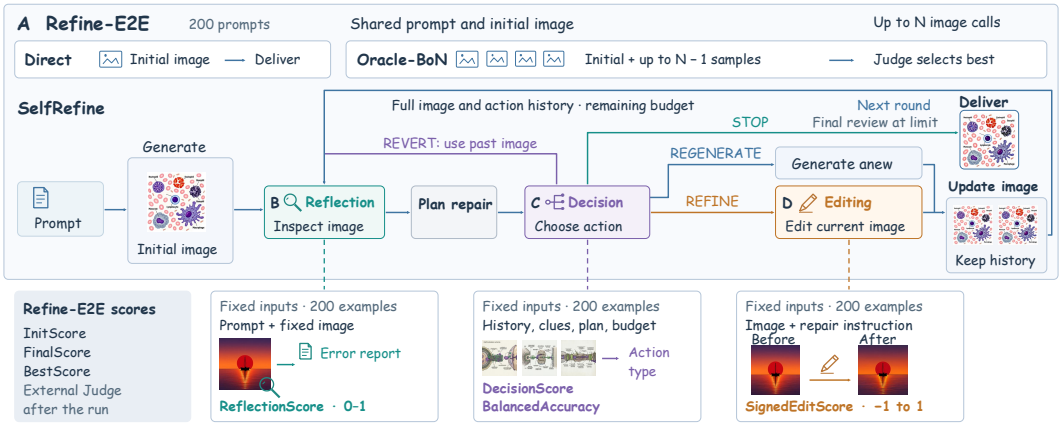


Figure 2: VISUALREFINEBENCH overview. (A) Direct, Oracle-BoN, and SelfRefine share an initial image generated without candidate selection. SelfRefine inspects, plans, chooses an action, and updates its image state across rounds. (B–D) Dashed links connect three stages to their independent diagnostic tasks. These tasks use fixed inputs and the original prompt. Archived images illustrate the interfaces; report symbols are schematic.

Decision uses multi-round trajectories from Qwen3.5-27B (Qwen Team, 2026a), Qwen-Image, and Qwen-Image-Edit (Wu et al., 2025), with each image scored by GPT-5.5. From these trajectories, we construct Decision examples covering substantial task completion, edits that damage correct content, and cases where further editing remains reasonable despite limited progress. Reviewers assign each example a set of acceptable actions, possibly including multiple actions. Each of the four subsets contains 200 task records, totaling 800; requests or images may be reused across subsets. Appendix B.1 gives construction and sampling details.

2.2 TASK DEFINITION AND SCORING

All four tasks receive the original request. Table 1 summarizes their additional inputs, scores, and evaluation methods. Rubrics, scoring references, and evaluation scores are withheld from evaluated models.

For example i, the rubric contains m_i scoring requirements. We write its j-th requirement as r_{ij} and its weight as $w_{ij} > 0$, for $j = 1, \ldots, m_i$. The rubric is $\mathcal{R}_i = \{(r_{ij}, w_{ij})\}_{j=1}^{m_i}$, and W_i is its total weight. For a completed evaluation of image I, $y_{ij}(I)$ is 1 if the evaluation marks requirement r_{ij} as met and 0 otherwise. The image score $S_i(I)$ is the weight of met requirements divided by W_i:

$$S_i(I) = \frac{1}{W_i} \sum_{j=1}^{m_i} w_{ij} y_{ij}(I), \qquad W_i = \sum_{j=1}^{m_i} w_{ij}. \tag{1}$$

All images for example i share the same rubric. This score $S_i(I) \in [0, 1]$ underlies REFINE-E2E and Editing; Editing keeps the Before labels fixed and judges only the After labels.

REFINE-E2E, Reflection, and Editing average scored repeats per source task, then weight tasks equally; Decision weights request groups equally. Task rules handle confirmed failures; missing evaluations and unknown outcomes are not set to zero. Bootstrap confidence intervals (Efron, 1992) resample source tasks or request groups (Appendix B.4). Section 3.1 gives Judge settings.

2.3 REFINE-E2E

REFINE-E2E tests whether autonomous refinement improves the image ultimately delivered under a fixed image-operation budget. For one run on example i, the Generator uses the original request to produce an initial image I_i^0. Let $\mathcal{T}_i$ be the set of all image states created in this run, including the initial image and images from branches abandoned after reverting. The delivered image is

Table 1: Task inputs and scoring. All tasks also receive the original request.

Task	Additional model inputs	Scores and evaluation
REFINE-E2E	Image/action history; current state; budget	**Init, Best, Final**: Judge scores $S_i(I)$ of initial, best, and delivered images.
Reflection	Fixed image	**ReflectionScore**: Judge rating $\in \{0, 0.5, 1\}$ using rubrics and reviewed error/repair references.
Decision	Image history; current image; confirmed errors; repair plan; budget	**DecisionScore**: 1 for an acceptable action, 0 otherwise; no Judge call.
Editing	Before image; repair instruction; preservation reminder	**SignedEditScore**: (repaired − lost weight) / initially unmet weight, clipped to $[-1, 1]$; fixed Before labels, judged After labels.

$I_i^{\text{out}} \in \mathcal{T}_i$. Using the full image/action history and remaining budget, the Controller plans corrections and selects REFINE (edit the current image with the Editor), REGENERATE (generate anew with the Generator), REVERT (return to a historical image and continue), or STOP (deliver the current image). The image-operation cap is a positive integer N, counting initial generation and each successful edit or regeneration; other actions consume no image budget. At budget exhaustion, the Controller can still select the current or a historical image for delivery. Equal image budgets do not imply equal total inference compute.

For the same generation configuration, task, and repetition, Direct, ORACLE-BON, and SELFREFINE (SR) share an initial image generated without candidate selection and its score. Direct delivers I_i^0. Following Best-of-N sampling (Ma et al., 2025), ORACLE-BON independently generates up to $N - 1$ additional images and uses external Judge scores to select the best, including the initial image. SR autonomously decides how to use the remaining budget and which image to deliver, without human selection or Judge scores. Thus ORACLE-BON measures quality with external selection, whereas SR measures autonomous delivery.

For each run, Init_i, Best_i, and Final_i denote the initial, highest, and delivered image scores:

$$\text{Init}_i = S_i(I_i^0), \qquad \text{Best}_i = \max_{I \in \mathcal{T}_i} S_i(I), \qquad \text{Final}_i = S_i(I_i^{\text{out}}). \tag{2}$$

$\text{Final}_i - \text{Init}_i$ measures delivered gain and $\text{Best}_i - \text{Final}_i$ the delivery gap. Both use all scored SR trajectories, including initial-only runs. SR fixes I_i^{out} before external scoring; Best is retrospective. Appendix B.2 gives execution details, and Appendix B.4 specifies the multi-image-only diagnostics.

2.4 REFLECTION, DECISION, AND EDITING

To help locate bottlenecks in autonomous refinement, we evaluate error diagnosis, action selection, and instructed editing through separate diagnostic tasks. Fixed inputs limit the influence of upstream model errors on each evaluated stage, and outputs are scored independently. Repair planning remains part of the REFINE-E2E evaluation.

Reflection: error diagnosis. Accurate error diagnosis is needed to target repairs without prompting unnecessary changes to correct content. Given the original request and a fixed image, the model reports image errors. For repetition r on example i, the Judge uses the references in Table 1 to assign a raw score $z_{ir} \in \{0, 1, 2\}$: 0 when no annotated error is correctly identified; 1 when at least one is identified but coverage or precision is insufficient, or relevant false claims remain; and 2 when all annotated errors are sufficiently identified without relevant false claims. The report's ReflectionScore is $z_{ir}/2$. Confirmed failures to deliver a valid report score zero; nonempty truncated reports are scored as delivered. We also report validity and scores conditional on scored valid reports (Appendix C.3).

Decision: action selection. Supplying confirmed errors and a repair plan lets us test action selection without requiring the model to diagnose errors or devise repairs. Given fixed history, budget, confirmed errors, and a candidate repair plan (Table 1), the model selects STOP (deliver current),

CONTINUE (execute the plan once), or REVERT (deliver a historical image and terminate). Only the action type is required; no image operation is run. For example i, let $\mathcal{A}_i^\star$ be the set of acceptable actions and a_{ir} the output from repetition r. The output scores $\mathbf{1}[a_{ir} \in \mathcal{A}_i^\star]$, where $\mathbf{1}[\cdot]$ is 1 when its condition holds and 0 otherwise. Invalid outputs score zero. No Judge call is needed. DecisionScore averages attempts within examples, examples within request groups, and groups equally. Balanced Accuracy (BA; Brodersen et al., 2010) equally averages the three group-weighted class recalls on examples with one acceptable action. Primary metrics require all planned outcomes to be known (Appendix B.3.2).

Editing: repair and content preservation. Editing tests whether a specified repair produces a net improvement after accounting for damage to correct content. Using a reviewed repair instruction and preservation reminder, the Editor turns the fixed Before image I_i^b into an After image I_i^a for example i. Superscripts b and a denote Before and After. Given the ordered pair, instruction, and $\mathcal{R}_i$, the Judge assigns the After label y_{ij}^a to each requirement r_{ij}; the Before label y_{ij}^b stays fixed. Both use 1 for met and 0 otherwise. Let F_i be the total weight of initially unmet requirements, G_i the weight repaired, and L_i the previously met weight lost after editing. Every Editing example has $F_i > 0$. We compute

$$F_i = \sum_{j=1}^{m_i} w_{ij}(1 - y_{ij}^b), \quad G_i = \sum_{j=1}^{m_i} w_{ij}(1 - y_{ij}^b)y_{ij}^a, \quad L_i = \sum_{j=1}^{m_i} w_{ij}y_{ij}^b(1 - y_{ij}^a). \tag{3}$$

The primary score divides net repaired weight by initial unmet weight, then clips it to $[-1, 1]$:

$$\text{SignedEditScore}_i = \text{clip}\left(\frac{G_i - L_i}{F_i}, -1, 1\right). \tag{4}$$

The clip operation changes values below -1 to -1 and values above 1 to 1; it leaves other values unchanged. The change in image score, ΔQ_i, uses total weight: $\Delta Q_i = S_i(I_i^a) - S_i(I_i^b) = (G_i - L_i)/W_i$. Each edit independently starts from the same Before image, and its score is clipped before aggregation. Appendix B.3.3 gives additional metrics and repetition rules.

3 Experiments

3.1 Experimental Setup

We evaluate 6 UMMs for all four tasks, 18 additional VLMs, and 12 image editing models for tasks designed for their roles in Section 2. The UMMs are SenseNova-U1 (Diao et al., 2026b), SenseNova-U1.5 (Diao et al., 2026a), its Preview checkpoint (SenseNova, 2026), Ovis-U1 (Wang et al., 2025a), BAGEL (Deng et al., 2025), and OmniGen2 (Wu et al., 2026). Refine-E2E uses an image-operation cap of $N = 4$. Reflection and Decision each have three planned outputs per record; Editing has three planned edits from the same Before image.

Model-based evaluation uses Qwen3.8-Flash-Next (Qiu et al., 2026) at temperature 0 to reduce sampling randomness, using fixed prompts and settings. Within a run, judgments are reused only for identical evaluation objects and full contexts. Table means use each method's scored tasks; SR gains compare Final and Init within trajectories, and paired comparisons use requests scored by both methods. Appendix C gives configurations, score coverage, and statistical methods.

3.2 Delivered Gains from Autonomous Refinement

Only three of the six UMMs show clear mean refinement gains. SenseNova-U1 and U1.5 improve mean delivered quality over Init by 4.32 and 4.65 points, while BAGEL gains 1.44 points; all three 95% confidence intervals lie above zero (Table 2). The intervals for Preview, Ovis, and OmniGen2 include zero. Integrating generation, understanding, and editing in one model does not by itself ensure effective autonomous refinement.

Improvements produced along a trajectory do not fully translate into delivery. Best exceeds Init by over 11 points on average for U1 and U1.5, whereas delivered gains are only 4–5 points,

Table 2: Autonomous refinement by six UMMs (scores $\times 100$; differences in quality points). Means use each method's scored tasks; SR gains are Final−Init within trajectories, with 95% CIs. Bold marks column maxima without implying significant differences. Appendix C gives score coverage and paired checks.

Model	SR Init	SR Final	Final − Init [95% CI]	SR Best	ORACLE-BON
SenseNova-U1	52.41	56.73	+4.32 [2.37, 6.40]	63.64	63.43
SenseNova-U1.5	**64.27**	**68.92**	**+4.65** [2.51, 6.82]	**75.38**	**74.47**
SenseNova-U1.5-Preview	58.05	58.16	+0.11 [−1.43, 1.64]	63.60	68.46
Ovis-U1	24.12	24.30	+0.18 [−0.37, 0.81]	25.02	30.67
BAGEL	21.22	22.66	+1.44 [0.24, 2.70]	25.65	28.35
OmniGen2	22.43	22.60	+0.17 [−0.93, 1.30]	27.16	31.21

leaving Best−Final gaps of 6.91 and 6.46 points (Figure 3(a)). All six UMMs deliver lower mean quality than their ORACLE-BON references, by 5.55–10.30 points; common-task paired comparisons support the same direction (Appendix C.2.2). U1 and U1.5 reach mean Best scores close to their ORACLE-BON references, so their trajectories contain gains lost at delivery.

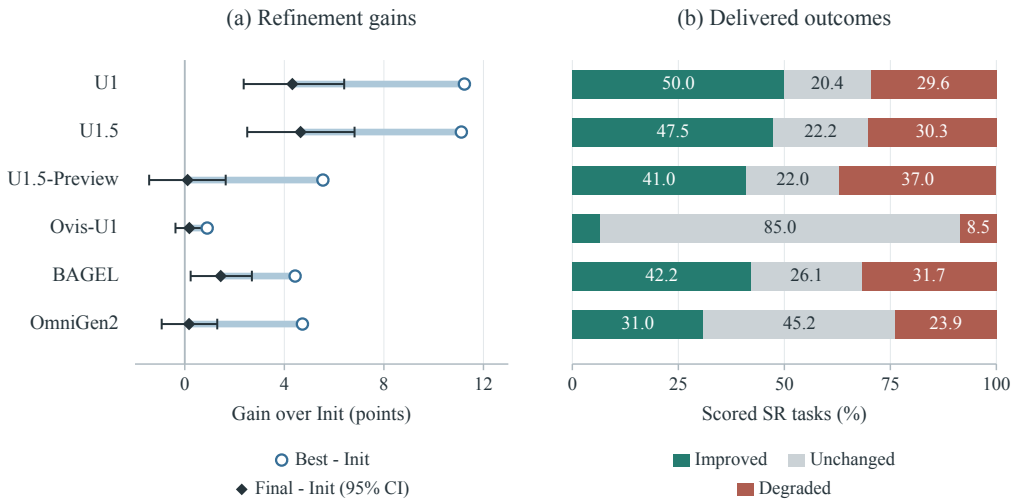


Figure 3: Autonomous refinement outcomes. (a) Mean Best−Init (circles), Final−Init (diamonds; 95% CIs), and Best−Final (segments), in quality points. (b) Improved, unchanged, and degraded task shares. Both panels use all scored SR trajectories, including initial-only runs; unchanged scores need not imply unchanged images. U1, U1.5, and U1.5-Preview denote SenseNova variants.

Similar mean gains correspond to different task-level outcomes. U1 and U1.5 improve only about half of requests, while roughly three in ten end below their initial quality (Figure 3(b)). Models with near-zero mean gains also differ: Preview improves 41.0% of tasks and degrades 37.0%, whereas Ovis leaves scores unchanged on 85.0%. Positive and negative gains largely cancel for Preview; Ovis mostly shows no score change.

Individual trajectories reveal distinct failure modes. On an illustrative epithelial-diagram task, U1 regenerates three times after its initial image, with scores declining from 69.0 through 66.5 and 61.5 to the delivered 50.5. Preview produces a last image scoring 64.0 but reverts to an earlier state scoring 54.0, below both that candidate and its initial 70.5. The first trajectory progressively loses initial quality; the second additionally discards a better available candidate at delivery. This example shows the complementary roles of preserving image quality and selecting among these states.

3.3 Capability Gaps in the Refinement Process

Table 3 and Figure 4 compare refinement capabilities. Specialist references are Qwen3.8-27B (Qwen Team, 2026c) and Molmo2-8B (Clark et al., 2026) for Reflection and Decision, and FLUX.1-Kontext-dev (Black Forest Labs et al., 2025) for Editing. Appendix C.3–C.5 gives full results.

Table 3: Reflection, Decision, and Editing (scores ×100; report rates in %). All six UMMs have Decision balanced accuracy at or below the fixed-action reference (33.33); the best model reaches 37.33 (95% CI: 31.15–43.07). UMMs are ordered by overall Reflection score; outputs without valid reports score zero, and valid-report scores retain judged zeros. Bold marks column-best model scores, not significance; † marks leading specialist references. Appendix C gives aggregation and intervals.

Model	Reflection			Decision		Editing
	Overall	Valid-report score	Valid reports (%)	Score	Balanced accuracy	Signed EditScore
Qwen3.8-27B †	**46.50**	**46.58**	99.83	35.51	35.93	—
Molmo2-8B †	17.92	18.67	96.83	**45.45**	**37.33**	—
Fixed action (reference)	—	—	—	—	33.33	—
SenseNova-U1	13.50	25.96	52.00	26.70	29.53	-7.16
Ovis-U1	12.00	12.00	99.67	1.70	2.13	-59.59
SenseNova-U1.5	11.00	25.88	42.50	27.84	32.32	**28.02**
OmniGen2	8.67	8.75	99.00	30.40	33.33	-64.87
BAGEL	5.00	5.43	91.83	26.42	31.47	-37.43
U1.5-Preview	3.75	4.44	84.50	27.56	32.36	-14.13
FLUX.1-Kontext-dev †	—	—	—	—	—	-2.55

UMMs lag behind the leading VLM in error diagnosis. Qwen3.8 scores 46.50 overall on Reflection, versus 3.75–13.50 for the six UMMs. For each UMM, more than half of scored valid reports fail to identify any annotated error correctly (Figure 4(a)), showing that report content also limits diagnosis.

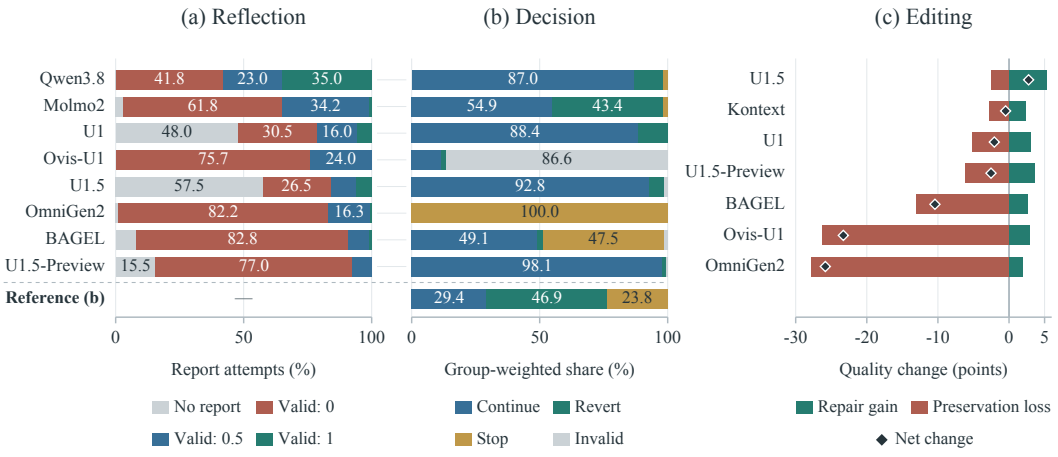


Figure 4: Atomic capability diagnostics. (a) No-report outcomes and valid Reflection scores (0/0.5/1). (b) Action shares on the single-reference subset, with equal prompt-group weights and invalid outputs retained. Panels (a,b) share model rows; the bottom reference applies only to (b). (c) Repair gains (right) and preservation losses (left), normalized by total rubric weight; diamonds show net quality changes in points.

Stronger error diagnosis can coexist with pronounced action bias. Reflection leader Qwen3.8 has Continue, Revert, and Stop recalls of 87.59%, 17.56%, and 2.63%. Molmo2 achieves the highest

BA, 37.33 (95% CI: 31.15–43.07), whose interval includes the constant-action reference of 33.33. Preview almost always selects Continue, whereas OmniGen2 always selects Stop (Figure 4(b)). Even with supplied diagnoses and repair plans, judgments remain uneven across the three actions.

Preservation losses offset repair gains for most evaluated editors. Before images average 89.47/100, with one or two unmet requirements in 183/200 images. Under this local-repair setting, only U1.5 has a positive mean SignedEditScore among 18 editors: 28.02 (95% CI: 19.27–36.57) after multiplying scores by 100. On the image-quality scale, it repairs 5.24 points and loses 2.49; Preview repairs 3.65 but loses 6.18 (Figure 4(c)). An unchanged-image control gives zero change scores for 193/200 sources; U1.5 retains a positive paired net gain over it (Appendix C.5.4).

Positive repair gains vary across repeated edits. U1.5 produces a positive SignedEditScore at least once for 127/200 images, but on all three attempts for only 74/200. The same Before image and repair instruction can yield both positive and non-positive edits (Appendix Figure C1).

These fixed-input results help locate distinct obstacles to autonomous refinement. Decision supplies confirmed errors and a repair plan, so action bias persists with this assistance. Editing supplies the repair instruction, yet target corrections can damage content that was already correct. The strongest diagnosis, action selection, and editing results come from different models. Because each track evaluates its own fixed task set, the component experiments below test whether these local advantages transfer to complete trajectories and final delivery.

3.4 Component Replacements and Delivered Gains

Baseline B uses U1.5-Preview for all three components. We choose U1.5-Preview for its relatively strong initial generation, intermediate editing performance among UMMs, and substantial room for improvement in autonomous refinement. C replaces the Controller with Qwen3.8-27B (Qwen Team, 2026c); G and E replace the Generator and Editor with released SenseNova-U1.5-8B-MoT (Diao et al., 2026a). Combined codes denote joint replacements (Table 4). All eight share the 200 requests and four-operation cap; comparisons pair common scored requests (Appendix C.6).

Table 4: Component replacements and autonomous refinement (scores $\times 100$; differences in quality points). Replacements are relative to B. Means use each configuration's scored tasks; comparisons pair common tasks. Bold marks the highest Best, Final, and Final−Init point estimates without implying significance. Oracle-BoN uses the corresponding Generator.

System configuration	Init	Best	Final	Final − Init	Oracle-BoN
B: Baseline	58.05	63.60	58.16	+0.11	68.46
C: +Controller	58.74	80.14	75.15	+16.41	68.44
G: +Generator	64.17	68.71	61.50	-2.67	73.73
E: +Editor	58.77	66.93	62.65	+3.88	68.83
CG: +Controller +Generator	63.10	82.90	78.99	+15.89	74.29
CE: +Controller +Editor	58.23	83.11	79.87	**+21.63**	68.38
GE: +Generator +Editor	64.26	69.17	65.04	+0.78	74.84
CGE: +Controller +Generator +Editor	63.89	**86.01**	**81.80**	+17.92	74.75

Controller replacement yields the largest delivered gains. In paired comparisons with B, replacing only the Controller raises Final by 17.02 points (95% CI: 14.50–19.80), compared with 4.49 for the Editor and 3.34 for the Generator. Across all four Generator–Editor combinations, Controller replacement improves Final by 16.82–17.51 points. All four configurations using Qwen3.8 also exceed their corresponding Oracle-BoN references, with paired advantages of 4.85–11.66 points (Appendix C.6).

Stronger Reflection accompanies better refinement candidates. Qwen3.8's clearest measured advantage over Preview is Reflection: 46.50 versus 3.75 overall (46.58 versus 4.44 on valid reports). Most Decision BA values cluster near the constant-action reference of 33.33; no model's 95% interval lies wholly above it. Qwen3.8 and Preview score 35.93 and 32.36 (Appendix C.4). C raises Best−Init by 15.89 points over B, accounting for most of its 17.02-point Final gain (Figure 5). This

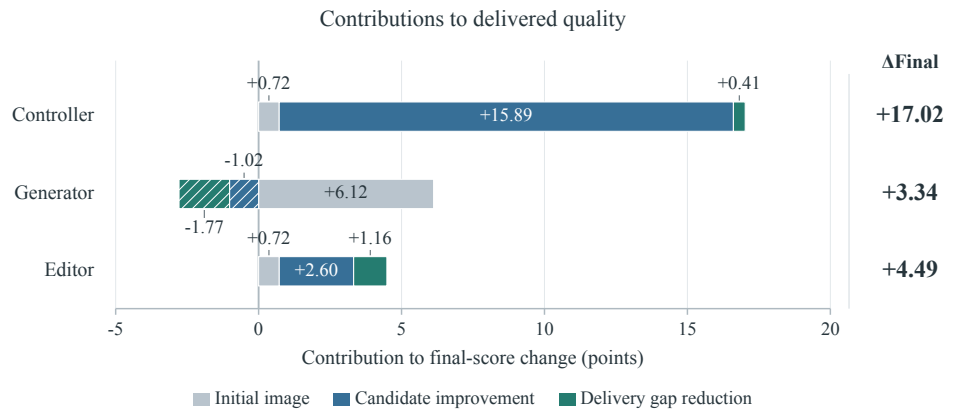


Figure 5: Paired contributions to delivered quality relative to B. Here Δ means the replacement value minus the baseline value for the same task. The contributions ΔInit, $\Delta(\text{Best} - \text{Init})$, and $-\Delta(\text{Best} - \text{Final})$ sum to ΔFinal. Negative contributions extend left and are hatched. Each comparison uses the same paired tasks and unrounded scores, in quality points. Controller and Editor replacements share baseline initial images; their Init differences reflect cross-run judgments. Full results and intervals are in Appendix C.6.

supports better diagnosis and corrective feedback, while joint changes to planning, instructions, and actions leave Reflection's causal contribution unresolved.

Stronger initial generation does not guarantee positive refinement gains. G increases Init over B by 6.12 points but Final by only 3.34 points (Figure 5). Its delivery is also 2.67 points below its own Init (95% CI: -4.32 to -1.10). Later operations and image selection lose part of the starting advantage, failing to retain the stronger initial images reliably.

Replacing the Editor improves delivered quality. In fixed-input Editing, released U1.5 repairs 1.60 more quality points and loses 3.69 fewer than Preview. Replacing the Editor raises Final over B by 4.49 points (95% CI: 2.98–5.91), and by another 4.97 points when added to C. The trajectory decomposition also shows greater improvement over Init for E (Figure 5). Better local repair and content preservation thus accompany higher-quality autonomous delivery.

4 Conclusion

VisualRefineBench connects end-to-end image refinement with three diagnostic tasks. For **RQ1**, under four image calls, only three of the six evaluated UMMs achieve clear mean gains, and all six deliver lower mean quality than their corresponding Oracle-BoN references. Their trajectories also contain higher-scoring images that are not delivered. For **RQ2**, models struggle with error diagnosis, action selection, and repair without damaging correct content.

Component replacements show that substantial improvements are already possible with existing models. A Controller stronger in Reflection improves Final by about 17 points across four Generator–Editor configurations, each exceeding its corresponding Oracle-BoN reference. These gains primarily reflect better candidates produced during refinement, although the contributions of feedback, repair instructions, and action choice remain unresolved. Editor replacement also improves delivery, supporting repair and preservation as practical improvement targets. Reliable selection of the final image remains a further challenge.

Iterative refinement offers a complementary direction for test-time scaling alongside broader best-of-N sampling. As diagnosis, repair, and selection improve, deeper refinement may raise attainable image quality under larger budgets. How to combine broader candidate exploration with deeper refinement, and allocate computation between them, remains a promising direction for future work.

AI USE STATEMENT

We used generative AI tools to assist with manuscript organization, English writing, and software development. The authors reviewed the AI-assisted text and code and take full responsibility for the final content of this work, including the accuracy of all references, numerical results, and claims.

ETHICS STATEMENT

VISUALREFINEBENCH is intended for research on model capabilities using generated and edited images. Its results should be interpreted within the scope of the evaluated tasks and should not be taken as evidence of model safety, fairness, or suitability for real-world deployment.

REPRODUCIBILITY STATEMENT

The evaluation uses fixed task sets and a fixed benchmark implementation. Appendix B specifies the evaluation protocol, and Appendix C reports the actual experimental configurations, scoring coverage, and statistical procedures. Run artifacts record configurations, model identifiers, prompts, seeds, outputs, judgments, and action histories to support verification of the reported results. We plan to release the benchmark code and data, together with the exact prompts, schemas, and a sanitized configuration manifest, to facilitate reproducibility.

REFERENCES

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

Black Forest Labs. Flux. 2: Frontier visual intelligence. Official model release, 2025. URL https://bfl.ai/blog/flux-2.

Black Forest Labs. FLUX.2 [klein]: Towards Interactive Visual Intelligence. Official model release, January 2026. URL https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Michael Bolin. Unrolling the Codex agent loop. OpenAI, 2026. URL https://openai.com/index/unrolling-the-codex-agent-loop/.

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*, pp. 3121–3124. IEEE, 2010.

Qi Cai, Jingwen Chen, Chengmin Gao, Zijian Gong, Yehao Li, Yingwei Pan, Yi Peng, Zhaofan Qiu, Kai Yu, Yiheng Zhang, et al. Hidream-o1-image: A natively unified image generative foundation model with pixel-level unified transformer. *arXiv preprint arXiv:2605.11061*, 2026.

Leon Liangyu Chen, Haoyu Ma, Zhipeng Fan, Ziqi Huang, Animesh Sinha, Xiaoliang Dai, Jialiang Wang, Zecheng He, Jianwei Yang, Chunyuan Li, et al. Unit: Unified multimodal chain-of-thought test-time scaling. *arXiv preprint arXiv:2602.12279*, 2026.

Meng Chu, Senqiao Yang, Haoxuan Che, Suiyun Zhang, Xichen Zhang, Shaozuo Yu, Haokun Gui, Zhefan Rao, Dandan Tu, Rui Liu, et al. Visiondirector: Vision-language guided closed-loop refinement for generative image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9203–9212, 2026.

Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Rohun Tripathi, Sangho Lee, Mohammadreza Salehi, Jason Ren, Chris Dongjoo Kim, Yinuo Yang, et al. Molmo2: Open weights and data for vision-language models with video understanding and grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28652–28668, 2026.

Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, et al. Emu3. 5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

DeepSeek-AI. DeepSeek-V4-Pro. Hugging Face model card, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro. Accessed September 26, 2026.

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.

Haiwen Diao, Jiahao Wang, Chenjing Ding, Hanming Deng, Jiangnan Chen, Ruixi Zhang, Ruohui Wang, Wenwen Tong, Xiangyu Fan, Yubo Wang, et al. Sensenova-u1. 5: Towards native unified visual intelligence. *arXiv preprint arXiv:2609.11929*, 2026a.

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, et al. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026b.

Yixuan Ding, Wei Huang, Ruijie Quan, Xiaojuan Qi, and Yi Yang. Is this edit correct? a multi-dimensional benchmark for reasoning-aware image editing. *arXiv preprint arXiv:2606.05172*, 2026. doi: 10.48550/arXiv.2606.05172. URL https://arxiv.org/abs/2606.05172.

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pp. 569–593. Springer, 1992.

Fengyi Fu, Mengqi Huang, Shaojin Wu, Yunsheng Jiang, Yufei Huo, Hao Li, Yinghang Song, Fei Ding, Jianzhu Guo, Qian He, et al. Lance: Unified multimodal modeling by multi-task synergy. *arXiv preprint arXiv:2605.18678*, 2026.

Gemma Team. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Gemma Team. Gemma 4 Technical Report. *arXiv preprint arXiv:2607.02770*, 2026. URL https://arxiv.org/abs/2607.02770.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.

Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models. *Advances in Neural Information Processing Systems*, 38:145576–145615, 2026.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Young-Jun Lee, Seungone Kim, Byung-Kwan Lee, Minkyeong Moon, Yechan Hwang, Jong Myoung Kim, Graham Neubig, Sean Welleck, and Ho-Jin Choi. Refinebench: Evaluating refinement capability of language models via checklists. In *International Conference on Learning Representations*, volume 2026, pp. 81042–81085, 2026.

Pengxiang Li, Zhi Gao, Bofei Zhang, Tao Yuan, Yuwei Wu, Mehrtash Harandi, Yunde Jia, Song-Chun Zhu, and Qing Li. Fire: A dataset for feedback integration and refinement evaluation of multimodal models. *Advances in Neural Information Processing Systems*, 37:101618–101640, 2024.

Yi Li, Haonan Wang, Qixiang Zhang, Boyu Xiao, Chenchang Hu, Hualiang Wang, and Xiaomeng Li. Unieval: Unified holistic evaluation for unified multimodal understanding and generation. *arXiv preprint arXiv:2505.10483*, 2025.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Scaling inference time compute for diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2523–2534. IEEE, 2025.

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

Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, et al. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

Mistral AI. Mistral-Small-3.2-24B-Instruct-2506. Hugging Face model card, 2025. URL https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506. Accessed September 26, 2026.

OpenAI. GPT-5.5 system card. OpenAI, 2026a. URL https://openai.com/index/gpt-5-5-system-card/.

OpenAI. GPT-Image-2. OpenAI model documentation, 2026b. URL https://developers.openai.com/api/docs/models/gpt-image-2. Accessed September 26, 2026.

OpenBMB. MiniCPM-V-4.6. Hugging Face model card, 2026a. URL https://huggingface.co/openbmb/MiniCPM-V-4.6. Accessed September 26, 2026.

OpenBMB. MiniCPM-V-4.6-Thinking. Hugging Face model card, 2026b. URL https://huggingface.co/openbmb/MiniCPM-V-4.6-Thinking. Accessed September 26, 2026.

Gio Paik, Geewook Kim, and Jinbae Im. Mmrefine: Unveiling the obstacles to robust refinement in multimodal large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 26883–26904, 2025.

Zihan Qiu, Zekun Wang, Xiao Li, Yanpeng Li, Yang Xu, Yixuan Wang, Huaqing Zhang, Rui Men, Bochao Mao, Chengruidong Zhang, et al. On the design of qwen3. 8-next architecture: Evaluation, efficiency, and training stability. *arXiv preprint arXiv:2608.30320*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents. Official model release, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.6-35B-A3B: Agentic Coding Power, Now Open to All. Official model release, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Qwen Team. Qwen3.8-27B. Hugging Face model card, 2026c. URL https://huggingface.co/Qwen/Qwen3.8-27B. Accessed September 26, 2026.

SenseNova. SenseNova-U1.5-8B-MoT-Preview. Hugging Face model card, 2026. URL https://huggingface.co/sensenova/SenseNova-U1.5-8B-MoT-Preview. Accessed September 26, 2026.

Guo-Hua Wang, Shanshan Zhao, Xinjie Zhang, Liangfu Cao, Pengxin Zhan, Lunhao Duan, Shiyin Lu, Minghao Fu, Xiaohao Chen, Jianshan Zhao, et al. Ovis-u1 technical report. *arXiv preprint arXiv:2506.23044*, 2025a.

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

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. *Advances in Neural Information Processing Systems*, 37: 128374–128395, 2024.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Towards instruction-aligned multimodal generation. *arXiv preprint arXiv:2506.18871*, 2026.

Xindi Wu, Dingli Yu, Yangsibo Huang, Olga Russakovsky, and Sanjeev Arora. Conceptmix: A compositional image generation benchmark with controllable difficulty. *Advances in Neural Information Processing Systems*, 37:86004–86047, 2024.

Wulin Xie, YiFan Zhang, Chaoyou Fu, Yang Shi, Jianshu Zeng, Bingyan Nie, Hongkai Chen, Zhang Zhang, and Liang Wang. Mme-unify: A comprehensive benchmark for unified multimodal understanding and generation models. In *International Conference on Learning Representations*, volume 2026, pp. 133535–133578, 2026.

Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Keqi Wang, Yibin Wang, et al. Lumina-DiMOO: An omni diffusion large language model for multi-modal generation and understanding. *arXiv preprint arXiv:2510.06308*, 2025.

Zitong Xu, Huiyu Duan, Yifei Nie, Mingda Du, Sijing Wu, Xiongkuo Min, Tianyi Zheng, Jian Zhang, Shusong Xu, Jinwei Chen, et al. Editrefiner: A human-aligned agentic framework for image editing refinement. *arXiv preprint arXiv:2605.07457*, 2026.

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

Yujia Yang, Yuanxiang Wang, Zhenyu Guan, Tiankun Yang, Chenxi Bao, Haopeng Jin, Jinwen Luo, Xinyu Zuo, Lisheng Duan, Haijin Liang, et al. Omni iie bench: Benchmarking the practical capabilities of image editing models. *arXiv preprint arXiv:2603.16944*, 2026b.

Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Idea2img: Iterative self-refinement with gpt-4v for automatic image design and generation. In *European conference on computer vision*, pp. 167–184. Springer, 2024.

Fukun Yin, Shiyu Liu, Yucheng Han, Zhibo Wang, Peng Xing, Rui Wang, Wei Cheng, Yingming Wang, Aojie Li, Zixin Yin, et al. Reasonedit: Towards reasoning-enhanced image editing models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23283–23293, 2026a.

Peiqi Yin, Jiangyun Zhu, Han Gao, Chenguang Zheng, Yongxiang Huang, Taichang Zhou, Ruirui Yang, Weizhi Liu, Weiqing Chen, Canlin Guo, et al. vllm-omni: Fully disaggregated serving for any-to-any multimodal models. *arXiv preprint arXiv:2602.02204*, 2026b.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Ranchi Zhao, et al. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11704–11715, 2026.

Ziyun Zeng, Hang Hua, and Jiebo Luo. Mira: Multimodal iterative reasoning agent for image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9563–9573, 2026.

Xinchen Zhang, Xiaoying Zhang, Youbin Wu, Yanbin Cao, Renrui Zhang, Ruihang Chu, Ling Yang, Yujiu Yang, and Guang Shi. Generative universal verifier as multimodal meta-reasoner. In *International Conference on Learning Representations*, volume 2026, pp. 109211–109243, 2026.

Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, et al. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. *Advances in Neural Information Processing Systems*, 38, 2026.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15329–15339. IEEE, 2025.

A Related work

UMM self-refinement. UMMs place image understanding, generation, and editing in one model (Wu et al., 2026; Deng et al., 2025). This integration allows a model to inspect and revise its own images over multiple rounds. Existing work realizes such loops through draft selection, visual feedback, stopping, rollback, or modular planning and editing (Yang et al., 2024; Zhuo et al., 2025; Wang et al., 2024; Chu et al., 2026; Zeng et al., 2026; Yang et al., 2026a; Xu et al., 2026). Jaiswal et al. compare iterative editing with parallel sampling under matched generator or editor calls (Jaiswal et al., 2026). UniT trains a UMM to reason, generate, and edit across rounds, and compares sequential scaling with image-count-matched Best-of-N (Chen et al., 2026). General UMM benchmarks assess understanding, generation, and their interaction (Xie et al., 2026; Li et al., 2025). VisualRefineBench focuses on autonomous multi-step image self-refinement and whether a UMM delivers the strongest state in its trajectory.

Refinement benchmarks. GenEval, T2I-CompBench, and ConceptMix measure compositional properties of generated images (Ghosh et al., 2023; Huang et al., 2023; Wu et al., 2024). RISEBench, Omni IIE Bench, and RE-Edit evaluate reasoning-informed, multi-turn, or reasoning-aware editing (Zhao et al., 2026; Yang et al., 2026b; Ding et al., 2026). FIRE, MMRefine, and RefineBench evaluate feedback integration, detection and correction, and refinement under varied feedback access, respectively (Li et al., 2024; Paik et al., 2025; Lee et al., 2026). R^3-Bench evaluates verdict and reflection fidelity and gains from model-written rectification instructions (Wang et al., 2026). ViVerBench evaluates visual verification, and OmniVerifier applies it to sequential generation and editing against Best-of-N (Zhang et al., 2026). These studies connect diagnostic measurements or component replacements to downstream image outcomes. VisualRefineBench addresses a specific delivery question by pairing the submitted image with the retrospective trajectory maximum. It pairs this task with controlled tests of three selected sub-abilities: Reflection, Decision, and Editing.

B Data and Evaluation Protocol

B.1 Data Sources and Annotation

Requests and evaluation requirements. Seed material consists of questions and other text collected from the web and organized by domain. DeepSeek-V4-Pro expands this material into generation requests covering scientific charts, textbook and technical diagrams, illustrations, and visual design. GPT-5.5 constructs the corresponding rubrics for human review. The original request defines the model's task; the detailed checks, weights, and reference annotations support external evaluation and are not additional inputs to autonomous refinement.

Reflection and Editing. GPT-Image 2 generates images with localized errors, mostly one or two unmet requirements. GPT-5.5, the Codex harness, and human review verify the errors and produce descriptions. Editing organizes these into trusted repair instructions to correct targets while preserving existing correct content; Reflection withholds them and requires independent diagnosis. The subsets share some data sources, with no one-to-one correspondence between records. A visual error may involve several rubric items.

Decision. Qwen3.5-27B, Qwen-Image, and Qwen-Image-Edit produce multi-round generation and editing trajectories for a given request; GPT-5.5 scores their images. During construction, reviewers inspect the trajectories and assign REVERT when an edit visibly damages the image and returning to an earlier state is appropriate, STOP when the task is substantially complete, and CONTINUE when successive edits make limited progress and further editing remains reasonable. More than one action may be acceptable, including REVERT when an earlier state is also a reasonable choice.

Example sampling. Early tests revealed a strong tendency to choose CONTINUE, motivating a lower proportion of CONTINUE-reference examples in the constructed set. The final distribution is reported in Appendix C.4.2. This curated mix probes stopping and reversion as well as continued editing; its action frequencies are not estimates of those in unrestricted refinement. DecisionScore and class-balanced accuracy are reported separately under their respective aggregation rules.

Dataset organization. Refine-E2E, Reflection, Decision, and Editing each contain 200 records. Decision forms 176 original-request groups; its single-reference-action subset contains 176 exam-

ples in 160 groups. Records separate model-visible inputs from evaluation-only references. All systems in each track use the same fixed tasks, prompts, annotations, and image assets. Requests or images may be reused across subsets, so 800 records do not imply 800 distinct images.

Inspectable records. Task records retain the request, image assets when applicable, rubrics and weights, and the corresponding repair references or acceptable-action sets. Model-visible inputs are kept separate from evaluation references, allowing the information used for each task to be inspected.

B.2 Self-Refinement Protocol

Controller context. Each request supplies the original request, complete ordered image history, current-state marker, remaining image budget, and compact action history. Images remain separate inputs with stable state identifiers; action records retain operations, instructions, and state transitions. The Controller receives no complete rubrics, weights, Judge evidence, or hidden scores. Reversion changes the current state without deleting generated branches.

Table B1: Actions in autonomous refinement. Image operations count successful generation or editing calls.

Action	Required content	State transition	Image operations
REFINE	Nonempty repair instruction	Edit current image; result becomes current	1
REGENERATE	Nonempty redraw plan	Generate from the request and plan; result becomes current	1
REVERT	Different valid historical ID	Target becomes current; history is retained	0
STOP	Action only	Deliver current image and terminate	0

Budget and termination. Initial generation consumes one image operation, and each successful edit or regeneration consumes one more. Inspection, planning, reverting, and stopping consume none; spent budget is never refunded. Budget exhaustion triggers a final non-generating review, including when the budget is one: retain the current image or select a historical image for delivery. A separate control-step cap bounds loops without image generation; reaching it delivers the current image without another Controller request.

Response validation. Actions follow a fixed structure, with required instructions, target identifiers, and currently allowed actions validated. Bounded format repair uses the same images and base context, supplemented by the invalid response and validation error; it cannot change the budget or allowed actions. If no valid response is obtained within this limit, the current image is delivered and the failure reason recorded. Its actual quality still determines the primary score; zero-scored failures may also be reported in a separate sensitivity analysis. Formal Generator and Editor calls allow only one transport attempt; failures and image-operation counts are recorded separately.

Shared start and candidate sets. For each generation configuration, task, and repetition, Direct, the first ORACLE-BON candidate, and the SR initial state share the same saved image and evaluation vector, reusing the saved results. ORACLE-BON selects from the complete pool of initial and additional independent candidates, breaking maximum-score ties by fixed candidate order. SR fixes delivery before external evaluation. Each new image call remains a distinct state even when its pixels repeat; BestScore includes reverted or abandoned branches. Initial-image sharing does not extend across different Generator configurations.

B.3 Diagnostic Task Protocols

B.3.1 Reflection

Three configured generation seeds independently produce three model reports per task. Each distinct valid final report receives one saved Judge score under the same evaluation context. Identical reports reuse the score while retaining the statistical weight of each generation in the task mean. Repetition applies to report generation; Judge scores for identical reports are reused within a run.

The Judge uses the original request, image, hidden rubrics, trusted repair reference, and defect annotations. For repetition r on example i, the report's raw score z_{ir} is 0 when no annotated defect is correctly identified; 1 when at least one is correctly identified, but coverage or precision is insufficient or a relevant false claim remains; and 2 when all annotated defects are sufficiently identified without relevant false claims. Equivalent wording and repeated statements do not change credit; the report's ReflectionScore is $z_{ir}/2$.

Only the report selected by the fixed final-text rule is evaluated, without an additional extraction model or supplementation from separate reasoning. A nonempty final report truncated by the output limit remains evaluated on its delivered content. No-report outcomes, Judge failures, and unknown execution outcomes are handled separately in B.4.

B.3.2 DECISION

Each example receives $R = 3$ independent action outputs. The standard setup allows one initial response and at most one structural repair. Only the action category is returned; no image action is executed, and neither a rationale nor historical image identifier is required. Error clues are checks already judged unsatisfied for the current image and do not automatically apply to historical images. Unlike ordinary E2E reverting, REVERT in Decision delivers a historical image and terminates.

Let $\mathcal{G}$ be the set of 176 request groups. Each group $g \in \mathcal{G}$ contains the indices of its source examples, and $n_g = |g|$ is their number. For example i, $\mathcal{A}_i^\star$ is the set of acceptable actions and a_{ir} is the output from repetition $r = 1, \ldots, R$. The indicator $\mathbf{1}[\cdot]$ equals 1 when its condition holds and 0 otherwise; invalid outputs receive zero. With $|\mathcal{G}|$ denoting the number of groups, DecisionScore averages over repetitions, examples within each group, and then groups:

$$\text{DecisionScore} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \frac{1}{R n_g} \sum_{i \in g} \sum_{r=1}^{R} \mathbf{1}[a_{ir} \in \mathcal{A}_i^\star]. \tag{B.1}$$

Balanced Accuracy uses only examples with one acceptable action. Their indices form the set $\mathcal{I}_{\text{single}}$, which contains 176 examples in 160 request groups. Let $g(i)$ identify example i's request group, n_g^{single} count examples from $\mathcal{I}_{\text{single}}$ in group g, and b_i be example i's sole acceptable action. Each output receives weight $\omega_{ir} = 1/(R n_{g(i)}^{\text{single}})$, so each included group has total weight 1. The action set is $\mathcal{C} = \{\text{STOP}, \text{CONTINUE}, \text{REVERT}\}$; $\perp$ denotes an invalid output. The weighted confusion matrix entry M_{cd} counts outputs with reference action $c \in \mathcal{C}$ and predicted action $d \in \mathcal{C} \cup \{\perp\}$. BA averages the three action recalls:

$$\begin{aligned} M_{cd} &= \sum_{i \in \mathcal{I}_{\text{single}}} \sum_{r=1}^{R} \omega_{ir}\, \mathbf{1}[b_i = c]\, \mathbf{1}[a_{ir} = d], \\ \text{BA} &= \frac{1}{3} \sum_{c \in \mathcal{C}} \frac{M_{cc}}{\sum_{d \in \mathcal{C} \cup \{\perp\}} M_{cd}}. \end{aligned} \tag{B.2}$$

Confirmed terminal delivery failures map to the invalid column and remain in reference-class denominators. Multi-reference examples enter DecisionScore but not this confusion matrix. If any reference class has zero support, BA is unavailable rather than averaged over the remaining classes. Both primary metrics are reported separately, without combining them into a total score.

B.3.3 EDITING

The Editor receives the original request, Before image, reviewed repair instruction, and preservation reminder. The Judge receives the ordered Before/After pair, repair instruction, complete rubrics, and fixed Before labels, and judges only After. Equation (3) defines the initial error weight F_i, repaired weight G_i, and lost passing weight L_i using fixed Before labels and Judge-assessed After labels.

SignedEditScore in Equation (4) clips $(G_i - L_i)/F_i$ to $[-1, 1]$. SignedNetRepair equals $(G_i - L_i)/F_i$ without clipping. RepairRate and DamageRate are G_i/F_i and L_i/F_i, respectively, so DamageRate

may exceed 1. The change in image score, ΔQ_i, divides net repaired weight by the total requirement weight W_i:

$$\Delta Q_i = S_i(I_i^a) - S_i(I_i^b) = \frac{G_i - L_i}{W_i} = \frac{G_i}{W_i} - \frac{L_i}{W_i}. \tag{B.3}$$

The Before score uses fixed labels. The two terms G_i/W_i and L_i/W_i measure repair gain and loss of previously met requirements on the image-score scale.

Every edit independently starts from the same original Before image. SignedEditScore is computed and clipped per trial before within-task and across-task aggregation; averaging repair and loss before clipping is not equivalent. Repetition counts follow the experiment configuration: backends supporting diffusion seeds use the configured independent seeds, whereas backends that do not transmit these seeds may use one unseeded call. After verdicts FAIL, UNCERTAIN, and BLOCKED contribute no passing weight. UNCERTAIN and BLOCKED retain their original statuses to distinguish them from confirmed visual errors.

B.4 Scoring and Aggregation

Evaluation consistency. Each direct comparison fixes the Judge model, prompt, and evaluation settings. Saved scores are reused for the same object only when the full evaluation context also matches: image scoring includes the task and image, Reflection additionally includes the final report, and Editing includes the ordered Before/After pair and repair-evaluation context. An identical After image with a different Before image or repair context thus does not share a score. Scores are reused within each run. Appendix C.1 gives Judge settings; C.6.5 examines cross-run score differences.

Task-level aggregation. For each metric in Refine-E2E, Reflection, or Editing, let $\mathcal{J}$ contain the indices of source tasks with at least one scored result. Let $\mathcal{K}_i$ contain task i's scored repetition indices and s_{ir} be its score for repetition $r \in \mathcal{K}_i$. The counts $|\mathcal{J}|$ and $|\mathcal{K}_i|$ are the numbers of included tasks and repetitions. The mean $\bar{s}$ averages within tasks, then weights tasks equally:

$$\bar{s} = \frac{1}{|\mathcal{J}|} \sum_{i \in \mathcal{J}} \frac{1}{|\mathcal{K}_i|} \sum_{r \in \mathcal{K}_i} s_{ir}. \tag{B.4}$$

Scorable results include protocol-defined zeros; tasks without scored results are not assigned zero. If no task has a scored result, the mean is unavailable. Results record planned executions, executions with known outcomes, and available scores; Appendix C reports coverage. Decision uses the complete planned examples and request-group weights in Equation (B.1).

Confidence intervals bootstrap source tasks, or request groups for Decision. Each BA resample reconstructs the weighted confusion matrix and recomputes the score. Resamples lacking a reference class yield no valid BA; the number of valid resamples is reported. Repeated outputs are resampled together with their source task; C.1.4 gives resample counts and seeds.

Failures and missingness. Confirmed Reflection no-report outcomes score zero, including empty reports, reasoning-only output, unresolved final boundaries, refusal, and recognized model delivery failures. Judge delivery or parsing failures for valid reports remain unscored, never zero. Executed and confirmed terminal invalid Decision outputs score zero and remain in class denominators; unexecuted, missing, or unclassified outcomes remain unknown. Any unknown planned output suppresses both formal Decision primary metrics and intervals. Editing executions with missing After images or incomplete evaluations are excluded from scored means. SR fallback delivery after format failure retains the image's actual score; a separate sensitivity analysis may score failures zero.

Populations for process metrics. Best−Final is defined per trajectory, but trajectories with only one image call offer no candidate-selection opportunity. Gap and BestRetentionRate in the formal result files use only trajectories with multiple image calls; the latter measures the fraction delivering a trajectory-best score. Chapter 4 additionally computes Best−Final over all scored SR trajectories, including initial-only trajectories; the two denominators are identified separately. AvgScore averages every generated state, including subsequently abandoned states. These process measures are reported separately from actual FinalScore.

Raw records and resource accounting. Run artifacts retain task and model identifiers, source and configuration hashes, prompts, schemas, visual inputs, seeds, raw responses, image hashes, state and parent identifiers, actions, delivered-image identifiers, rubric judgments, and failures. Scoring records remain separate from model-visible histories. Image operations define the primary budget; Controller tokens and latency, image-generation cost, and post-hoc Judge resources are distinct quantities. The present results do not establish a matched-total-compute comparison.

C Experimental Settings and Additional Results

Appendix B defines the data and evaluation protocol; this appendix records the experimental settings and additional results. C.1 gives shared settings, C.2 autonomous-refinement results, C.3–C.5 the three diagnostics, and C.6 component comparisons.

C.1 Experimental Setup

C.1.1 Evaluation Scope and Execution Budgets

The experiments include six UMMs, 18 additional VLMs, and 12 dedicated editors, yielding 24 Reflection/Decision entries and 18 Editing entries. Settings below are taken from the saved configurations of the reported experiments; All tracks use fixed task sets, with run seed 42.

Table C1: Task sets and execution settings.

Track	Input size	Execution per source record
Refine-E2E	200 requests	Direct: 1 image operation; Oracle-BoN / SR: at most 4, including the initial image
Reflection	200 source records	3 report generations; seeds 42, 43, 44
Decision	200 examples in 176 prompt groups	3 action generations; seeds 42, 43, 44
Editing	200 source images	3 edits from the same Before image; seeds 42, 43, 44

Refine-E2E allows at most 32 control steps and two responses for action parsing; Decision also allows at most two responses for structured-output parsing. Transport-attempt limits are one for Generator, Editor, and Controller, and three for Judge. Transport retries and structured-output repair are separate limits, and Judge calls do not consume the image-operation budget. Fixed seeds define the experiment settings; identical text or image outputs still depend on the deployment.

C.1.2 Controller and Judge

All Controllers have a 32,768-token output limit, a client-side input guard of 240,000 characters, and `image_detail=auto`. The six UMM Controllers use the `vllm_omni` interface (vLLM-Omni; Yin et al., 2026b), allow up to 16 input images, and set `do_sample=false`. The 18 additional VLMs use `openai_chat` / vLLM (Kwon et al., 2023), allow up to four images, and set `json_mode=true`; their deployment records specify a 32,768-token context and bfloat16 dtype. The character guard is distinct from model context length. Models grouped below use the same decoding settings; "—" indicates an unset configuration field.

Table C2: Controller decoding parameters.

Controller	Temp.	top_p	top_k	Thinking settings	Other sampling settings
U1 / U1.5 / Preview / Ovis-U1 / BAGEL	0	—	—	think=true	—
OmniGen2	0	—	—	think=false	—

Table C2 (continued)

Controller	Temp.	top_p	top_k	Thinking settings	Other sampling settings
qwen3.6-35b-a3b / qwen3.5-9b	1	0.95	20	enable_thinking=true; budget=16384	min_p=0; presence_penalty=1.5; repetition_penalty=1
qwen3.8-27b	1	0.95	20	enable_thinking=true; budget=16384; effort=xhigh	min_p=0; presence_penalty=0; repetition_penalty=1
gemma-3-27b-it / gemma-4-26b-a4b-it / gemma-4-31b-it	1	0.95	64	—	—
internvl3.5-8b / internvl3.5-38b / internvl3.5-14b / internvl3.5-30b-a3b	0.6	0.95	50	budget=16384	—
minicpm-v-4.5	0.6	0.95	20	—	—
minicpm-v-4.6	0.7	1	0	—	repetition_penalty=1
glm-4.6v-flash	0.8	0.6	2	enable_thinking=true; budget=16384	—
qwen3-vl-8b-thinking / qwen3-vl-32b-thinking	1	0.95	20	enable_thinking=true; budget=16384	repetition_penalty=1
minicpm-v-4.6-thinking	0.7	1	0	enable_thinking=true; budget=16384	repetition_penalty=1
mistral-small-3.2-24b-instruct-2506	0.15	—	—	—	—
molmo2-8b	0	—	—	—	—

U1, U1.5, and Preview use the `<image>` placeholder. Qwen3.8-27B uses the same decoding settings in the diagnostic tracks and all Controller-replacement configurations; its template sets both `enable_thinking=true` and `reasoning_effort=xhigh`.

Tracks requiring model-based evaluation share one Judge configuration: `qwen3.8-flash-next`, temperature 0, a 32,768-token output limit, up to 16 input images and 240,000 input characters, `image_detail=auto`, and `json_mode=true`. Template parameters are `enable_thinking=true` and `reasoning_effort=xhigh`; the separate `thinking_token_budget` field is unset. Other sampling parameters are not explicitly set. Rubric evaluation uses a batch size of 8 with a fixed evaluation prompt and response schema. Decision is scored directly against reference actions without Judge calls. Within a run, identical evaluation objects and full contexts reuse scores; C.6.5 examines shared images across runs.

C.1.3 Generator, Editor, and Deployment Records

The tables list image parameters from run settings, using their field names. Every Generator specifies 1024×1024, uses `edit_size=configured`, and validates output dimensions.

Table C3: Generator image parameters.

Generator	Steps	cfg_scale	img_cfg_scale	max_sequence_length
SenseNova-U1-8B-MoT	50	4	1	4096
SenseNova-U1.5-8B-MoT	50	4	1	4096
SenseNova-U1.5-8B-MoT-Preview	50	4	1	4096
Ovis-U1-3B	50	5	0	—
BAGEL-7B-MoT	50	4	1	—
OmniGen2	50	4	2	1024

Editors use the `images_edits` interface and also specify 1024×1024. BAGEL uses `edit_size=configured` with dimension validation; all other editors use `edit_size=auto` without requiring output dimensions to equal the configured size.

Table C4: Editor image parameters.

Editor	Steps	cfg_scale	img_cfg_scale	max_sequence_length
SenseNova-U1-8B-MoT	50	4	1	4096
SenseNova-U1.5-8B-MoT	50	4	1	4096
SenseNova-U1.5-8B-MoT-Preview	50	4	1	4096
Ovis-U1-3B	50	6	1.5	—
BAGEL-7B-MoT	50	4	2	—
OmniGen2	50	5	2	1024
Qwen-Image-Edit-2511	40	1	4	512
FLUX.1-Kontext-dev	28	2.5	1	512
FLUX.2-dev	50	4	1	512
FLUX.2-klein-9B	4	1	1	512
LongCat-Image-Edit	50	4.5	1	512
Lumina-DiMOO	64	2.5	4	512
HiDream-O1-Image	50	5	1	512
Qwen-Image-Edit-2509	40	1	4	512
Step1X-Edit-v1p2	50	6	6	512
FLUX.2-klein-4B	4	1	1	512
Lance_3B	50	4	1	4096
Emu3.5	—	2	2	—

Preview's Generator additionally sets `timestep_shift=3`. BAGEL's Generator sets `timestep_shift=3`, `cfg_text_scale=4`, and `cfg_interval=[0.4,1.0]`; its Editor sets `timestep_shift=3`, `cfg_text_scale=4`, `cfg_img_scale=2`, `cfg_interval=[0,1]`, `cfg_renorm_type=text_channel`, and `cfg_renorm_min=0`. Other listed models have an empty `extra_body`, but may have recorded model-specific adapter defaults. For example, U1/U1.5 deployment records also list `timestep_shift=3` and `cfg_norm=none`.

The configuration manifest preserves model identifiers, declared deployment revisions and visual preprocessing, available template hashes, and each original configuration's SHA-256. Some model versions are identified by model-index or configuration-file hashes; the Judge checkpoint field is operator-declared and its template hash is unavailable. The manifest identifies whether each record denotes a weight revision, model index, or configuration hash.

C.1.4 Comparison Units, Missing Scores, and Intervals

Quality means and differences are computed before rounding and then scaled by 100; a change of 0.01 is one quality point. Method means use their own scored samples, while cross-method and cross-configuration differences use commonly scored task IDs. SR Init, Best, Final, and Final−Init always come from the same trajectory, without substituting a separate Direct record for Init.

Table C5: Sampling units and aggregation of repeats.

Statistic	Aggregation of repeated executions	Independent resampling unit
REFINE-E2E and component replacements	Gains within trajectories; request pairing across conditions	Source request
Reflection	Scored reports averaged within sources, then sources weighted equally	Source record; conditional scores use sources with valid reports with scores

Table C5 (continued)

Statistic	Aggregation of repeated executions	Independent resampling unit
DecisionScore / BA	Equal rolls within examples, examples within prompt groups, and prompt groups overall	Prompt group; weighted confusion matrix recomputed for BA
Editing	Scored edits among three trials averaged within images, then images weighted equally	Source image

Failure and missing-score handling follows Appendix B.4: confirmed missing reports and invalid actions receive protocol-defined zeros, while unknown outcomes and evaluation failures are not imputed as zero. The following sections report the observed score coverage.

Intervals in this chapter use percentile 95% bootstrap estimates from 2,000 resamples of the units above, except for the added no-edit control analysis in C.5.4, which uses 10,000 resamples. Self-refinement gains and primary diagnostic scores retain their officially exported intervals; added paired comparisons, conditional Reflection scores, and component gain decompositions use seed 42. The repeated Preview baseline in C.6 retains the Section 3.2 export, while other component within-trajectory gains are recomputed with seed 42. Paired resampling retains both observations per request, without multiplicity correction. Intervals describe task-sampling uncertainty conditional on recorded executions and scores, not variation from independent reruns or repeated Judge evaluations.

C.1.5 Model Identifiers

The following tables list the source identifiers and references of the evaluated models. References identify the model reports or official releases; the identifiers specify the evaluated checkpoints. Run configurations are reported above, and score coverage appears in the corresponding result sections. UMM Editing entries are counted once.

Table C6: UMMs: source model identifiers.

Model identifier	Reference
`sensenova/SenseNova-U1-8B-MoT`	(Diao et al., 2026b)
`AIDC-AI/Ovis-U1-3B`	(Wang et al., 2025a)
`sensenova/SenseNova-U1.5-8B-MoT`	(Diao et al., 2026a)
`ByteDance-Seed/BAGEL-7B-MoT`	(Deng et al., 2025)
`OmniGen2/OmniGen2`	(Wu et al., 2026)
`sensenova/SenseNova-U1.5-8B-MoT-Preview`	(SenseNova, 2026)

Table C7: Additional VLMs: source model identifiers.

Model identifier	Reference
`Qwen/Qwen3.6-35B-A3B`	(Qwen Team, 2026b)
`Qwen/Qwen3.5-9B`	(Qwen Team, 2026a)
`Qwen/Qwen3.8-27B`	(Qwen Team, 2026c)
`google/gemma-3-27b-it`	(Gemma Team, 2025)
`OpenGVLab/InternVL3_5-8B`	(Wang et al., 2025b)
`OpenGVLab/InternVL3_5-38B`	(Wang et al., 2025b)
`openbmb/MiniCPM-V-4_5`	(Yu et al., 2026)
`openbmb/MiniCPM-V-4.6`	(Yu et al., 2026; OpenBMB, 2026a)
`google/gemma-4-26B-A4B-it`	(Gemma Team, 2026)
`google/gemma-4-31B-it`	(Gemma Team, 2026)
`zai-org/GLM-4.6V-Flash`	(Hong et al., 2025)

Table C7 (continued)

Model identifier	Reference
`Qwen/Qwen3-VL-8B-Thinking`	(Bai et al., 2025)
`Qwen/Qwen3-VL-32B-Thinking`	(Bai et al., 2025)
`openbmb/MiniCPM-V-4.6-Thinking`	(Yu et al., 2026; OpenBMB, 2026b)
`OpenGVLab/InternVL3_5-14B`	(Wang et al., 2025b)
`OpenGVLab/InternVL3_5-30B-A3B`	(Wang et al., 2025b)
`mistralai/Mistral-Small-3.2-24B-Instruct-2506`	(Mistral AI, 2025)
`allenai/Molmo2-8B`	(Clark et al., 2026)

Table C8: Dedicated editors: source model identifiers.

Model identifier	Reference
`Qwen/Qwen-Image-Edit-2511`	(Wu et al., 2025)
`black-forest-labs/FLUX.1-Kontext-dev`	(Black Forest Labs et al., 2025)
`black-forest-labs/FLUX.2-dev`	(Black Forest Labs, 2025)
`black-forest-labs/FLUX.2-klein-9B`	(Black Forest Labs, 2026)
`meituan-longcat/LongCat-Image-Edit`	(Meituan LongCat Team et al., 2025)
`Alpha-VLLM/Lumina-DiMOO`	(Xin et al., 2025)
`HiDream-ai/HiDream-O1-Image`	(Cai et al., 2026)
`Qwen/Qwen-Image-Edit-2509`	(Wu et al., 2025)
`stepfun-ai/Step1X-Edit-v1p2`	(Liu et al., 2025; Yin et al., 2026a)
`black-forest-labs/FLUX.2-klein-4B`	(Black Forest Labs, 2026)
`bytedance-research/Lance`	(Fu et al., 2026)
`BAAI/Emu3.5`	(Cui et al., 2025)

C.2 Additional Autonomous-Refinement Results

C.2.1 Scored Samples and Trajectory Statistics

Each method plans 200 tasks per model; scored coverage is listed below. SR Init, Final, Best, gains, and task proportions use the same scored trajectories, including initial-only trajectories; Oracle-BoN uses its own scored samples. Missing scores and fallback deliveries follow C.1.4.

Table C9: Scored tasks by method.

System	Scored Direct tasks	Scored SR tasks	Scored Oracle-BoN tasks
SenseNova-U1-8B-MoT	198	196	192
SenseNova-U1.5-8B-MoT	200	198	196
SenseNova-U1.5-Preview	199	200	195
Ovis-U1-3B	200	200	199
BAGEL-7B-MoT	200	199	200
OmniGen2	199	197	196

For U1, the counts of improved tasks, regressed tasks, and tasks with Final below Best are 98, 58, and 112, all out of 196. For U1.5, the corresponding counts are 94, 60, and 103 out of 198. Best − Final here uses all scored SR trajectories; the frozen report also exports a separate metric of the same name for multi-image trajectories. SR gain intervals in the main table retain the official exports.

C.2.2 PAIRED SR AND ORACLE-BON COMPARISONS

To check the effect of differing scored-task sets on cross-method comparisons, we compute task-level differences on each model's commonly scored SR and ORACLE-BON tasks and apply the request-paired resampling in C.1.4. ORACLE-BON − SR Final remains positive for every model, with intervals above zero, matching the direction of the differences between each method's own scored-task mean. The two aggregation choices yield gaps less than 0.25 quality points apart.

Table C10: Common-task comparison of ORACLE-BON and SR.

System	Common tasks	Difference of own scored-task means	Paired difference [95% CI]
SenseNova-U1-8B-MoT	189	+6.70	+6.46 [4.52, 8.38]
SenseNova-U1.5-8B-MoT	194	+5.55	+5.75 [3.68, 7.81]
SenseNova-U1.5-Preview	195	+10.30	+10.27 [8.50, 12.05]
Ovis-U1-3B	199	+6.37	+6.35 [5.35, 7.29]
BAGEL-7B-MoT	199	+5.69	+5.65 [4.32, 6.91]
OmniGen2	194	+8.61	+8.46 [7.16, 9.80]

All differences are ORACLE-BON − SR Final in quality points. Paired intervals apply only to the paired estimates, not to differences between each method's own scored-task mean. For U1 and U1.5, the paired SR Best − ORACLE-BON differences are +0.53 points (95% CI: −1.08 to 2.05) and +0.70 points (−0.88 to 2.37), respectively. Both intervals include zero.

C.3 REFLECTION: OVERALL SCORES, VALID-REPORT QUALITY, AND COVERAGE

C.3.1 OVERALL AND CONDITIONAL SCORES

Each model has 200 source records and three report generations per source. The frozen overall score averages available outputs within each source, then sources equally, assigning zero to confirmed outputs without a valid report. The added valid-report score averages valid reports with scores within each source, then equally averages sources with at least one such report. Valid reports receiving a judged zero remain included, as do valid reports truncated at the end. Valid-report rates use executions with known outcomes; all executions in this snapshot are resolved.

Overall intervals retain the official exports. Conditional intervals resample only sources with at least one scored valid report, using the remaining settings in C.1.4. Conditional scores measure report quality when delivered. Models are sorted by unrounded overall score, highest first, then by name.

Table C11: Complete Reflection scores and coverage.

Model	Overall [95% CI]	Valid-report [95% CI]	Valid reports %	Scored valid reports / 600	Eligible sources / 200
Qwen3.8-27B	46.50 [41.50, 51.33]	46.58 [41.50, 51.33]	99.83	599	200
qwen3.6-35b-a3b	32.92 [28.83, 36.92]	33.08 [29.17, 37.17]	99.67	598	200
qwen3.5-9b	27.92 [24.42, 31.50]	28.21 [24.54, 31.96]	99.00	594	200
gemma-4-31b-it	20.42 [16.92, 23.92]	20.81 [17.29, 24.62]	96.67	580	199
gemma-3-27b-it	19.42 [16.58, 22.17]	19.79 [16.92, 22.67]	98.17	589	200
Molmo2-8B	17.92 [14.50, 21.17]	18.67 [15.25, 22.00]	96.83	581	200
internvl3.5-8b	17.50 [15.00, 20.08]	17.67 [15.00, 20.42]	99.33	596	200
gemma-4-26b-a4b-it	17.29 [14.11, 20.52]	17.29 [14.03, 20.60]	99.83	578	199
internvl3.5-38b	17.08 [14.33, 19.75]	17.29 [14.33, 20.29]	98.50	591	200
qwen3-vl-32b-thinking	17.08 [13.67, 20.92]	17.08 [13.58, 20.83]	100.00	600	200
internvl3.5-14b	15.58 [13.17, 18.17]	15.79 [13.12, 18.50]	98.67	592	200

Table C11 (continued)

Model	Overall [95% CI]	Valid-report [95% CI]	Valid reports %	Scored valid reports / 600	Eligible sources / 200
internvl3.5-30b-a3b	15.58 [13.08, 18.33]	16.08 [13.42, 18.71]	97.67	586	200
mistral-small-3.2-24b-instruct-2506	14.83 [12.00, 17.50]	15.00 [12.46, 17.83]	99.00	594	200
minicpm-v-4.5	14.50 [12.00, 16.92]	15.87 [13.17, 18.39]	91.00	546	198
qwen3-vl-8b-thinking	14.33 [11.75, 17.25]	14.33 [11.66, 17.25]	99.33	596	200
SenseNova-U1	13.50 [9.75, 17.50]	25.96 [19.71, 32.21]	52.00	312	104
minicpm-v-4.6	12.25 [10.08, 14.33]	12.25 [10.00, 14.50]	100.00	600	200
Ovis-U1	12.00 [9.00, 14.75]	12.00 [9.25, 15.00]	99.67	598	200
SenseNova-U1.5	11.00 [7.50, 15.00]	25.88 [18.24, 34.12]	42.50	255	85
glm-4.6v-flash	10.25 [7.67, 12.92]	10.75 [8.00, 13.59]	97.17	583	200
OmniGen2	8.67 [6.17, 11.25]	8.75 [6.25, 11.50]	99.00	594	200
minicpm-v-4.6-thinking	6.33 [4.58, 8.42]	6.62 [4.79, 8.71]	98.33	590	200
BAGEL	5.00 [3.00, 7.50]	5.43 [2.99, 8.15]	91.83	551	184
U1.5-Preview	3.75 [2.00, 5.75]	4.44 [2.37, 6.51]	84.50	507	169

C.3.2 Report Coverage and No-Report Reasons

Gemma-4-26B-A4B-IT has 599 valid reports, of which 21 lack a valid Judge score. These remain in the report-validity rate but are excluded from score means rather than assigned zero. A pooled mean over valid reports with scores differs from the conditional score averaged by source by at most 0.31 points across models; the main text uses the latter.

Missing-report reasons for UMMs are listed below. All U1 and U1.5 executions record finish_reason=stop, and neither run marks valid reports as final_truncated. These missing reports include final-text boundary failures and cannot all be attributed to the output limit.

Table C12: Missing-report reason counts for UMMs.

Model	Unclosed thinking	Thinking only	Unresolved final boundary	Recognized generation delivery failure
SenseNova-U1	270	15	3	0
Ovis-U1	0	0	0	2
SenseNova-U1.5	339	6	0	0
OmniGen2	0	0	0	6
BAGEL	0	48	0	1
U1.5-Preview	0	90	3	0

SenseNova implementation checks. Reflection checks with the official implementations and vLLM-Omni both exhibited prolonged reasoning without usable final reports for U1 and U1.5. Earlier official-implementation runs used an 8,192-token output limit; each model delivered 69 valid reports in 200 generations, with the other 131 containing only reasoning. In the reported vLLM-Omni runs with a 32,768-token limit, valid-report counts are 312/600 for U1 and 255/600 for U1.5 (Table C11). Final-report delivery failures therefore persisted across both execution paths. The checks differ in output limits and repetition settings and document this shared behavior; they do not isolate a backend effect or replace the reported benchmark results.

C.4 Decision: Complete Results and Reference Distributions

C.4.1 Scores and Action Recalls

Each model has 600 executions with known outcomes. DecisionScore uses all 200 examples in 176 prompt groups; BA and class recalls use 176 single-reference examples in 160 groups. Rolls are equally weighted within examples, examples within groups, and groups overall. BA intervals retain the official exports, recomputing the weighted confusion matrix at the prompt-group level as specified in C.1.4. Models are ordered by DecisionScore. Invalid-output rates below pool all executions and differ from the single-reference, group-weighted shares in the main figure.

Table C13: Complete Decision scores, recalls, and invalid rates.

Model	Decision Score	BA [95% CI]	Continue recall	Revert recall	Stop recall	All-output invalid %
Molmo2-8B	45.45	37.33 [31.15, 43.07]	63.12	47.11	1.75	0.00
internvl3.5-30b-a3b	39.30	32.88 [28.23, 37.30]	53.55	44.22	0.88	0.00
Qwen3.8-27B	35.51	35.93 [32.31, 39.31]	87.59	17.56	2.63	0.00
minicpm-v-4.5	34.09	33.45 [29.22, 37.90]	79.08	16.89	4.39	0.00
qwen3-vl-8b-thinking	33.52	31.17 [26.98, 35.18]	71.28	22.22	0.00	0.00
qwen3.6-35b-a3b	31.72	34.01 [31.80, 36.07]	92.91	9.11	0.00	0.00
OmniGen2	30.40	33.33 [33.33, 33.33]	0.00	0.00	100.00	0.00
qwen3.5-9b	30.30	33.72 [31.72, 35.68]	93.62	7.56	0.00	0.00
minicpm-v-4.6	29.26	33.33 [33.33, 33.33]	100.00	0.00	0.00	0.00
qwen3-vl-32b-thinking	28.88	33.13 [31.63, 34.38]	97.16	2.22	0.00	0.00
gemma-4-26b-a4b-it	28.69	33.10 [32.53, 33.33]	99.29	0.00	0.00	0.00
internvl3.5-38b	28.50	33.42 [32.09, 35.07]	97.16	1.33	1.75	0.00
mistral-small-3.2-24b-instruct-2506	28.50	33.10 [32.53, 33.33]	99.29	0.00	0.00	0.00
internvl3.5-8b	28.31	33.25 [32.66, 33.75]	99.29	0.44	0.00	0.00
gemma-3-27b-it	28.12	33.33 [33.33, 33.33]	100.00	0.00	0.00	0.00
gemma-4-31b-it	28.12	33.33 [33.33, 33.33]	100.00	0.00	0.00	0.00
glm-4.6v-flash	28.12	33.33 [33.33, 33.33]	100.00	0.00	0.00	0.00
internvl3.5-14b	28.12	33.33 [33.33, 33.33]	100.00	0.00	0.00	0.00
minicpm-v-4.6-thinking	28.03	32.96 [32.14, 33.63]	98.23	0.67	0.00	0.00
SenseNova-U1.5	27.84	32.32 [29.44, 34.76]	93.62	3.33	0.00	1.00
U1.5-Preview	27.56	32.36 [29.91, 34.20]	95.74	1.33	0.00	0.50
SenseNova-U1	26.70	29.53 [25.36, 33.19]	81.91	6.67	0.00	0.00
BAGEL	26.42	31.47 [24.62, 38.42]	40.43	2.67	51.32	1.00
Ovis-U1	1.70	2.13 [0.00, 4.82]	6.38	0.00	0.00	87.50

Always Continue, Revert, and Stop obtain DecisionScores of 28.125, 52.557, and 30.398, respectively, with BA 33.333 for each. No model has a BA 95% interval entirely above 1/3.

C.4.2 Reference Actions and Figure Distributions

The main figure uses only single-reference examples. Raw reference counts are 51 Continue, 81 Revert, and 44 Stop. With the same prompt-group weights, shares are 29.375%, 46.875%, and 23.750%. Model bars show three outputs on this subset, retaining Invalid. The distribution below covers all 200 examples; DecisionScore includes multi-reference examples.

This distribution reflects the construction-stage sampling adjustment described in Appendix B.1. A strong tendency to select CONTINUE remains visible in the evaluated models' action distributions despite its reduced reference share (Figure 4(b)).

Table C14: Example counts by reference-action set.

Accepted-action set	Examples
CONTINUE	51
REVERT	81
STOP	44
CONTINUE / REVERT	2
CONTINUE / STOP	1
REVERT / STOP	21

C.5 Editing: Repair, Preservation Losses, and Repeated Executions

C.5.1 Complete Repair and Preservation Results

All 18 models cover 200 source images with three planned edits each. Scored edits are averaged per image, then images are weighted equally. Models are ordered by SignedEditScore; intervals retain the official exports, with sampling units specified in C.1.4. SignedEditScore follows Equation (4): repaired minus lost weight, divided by initially unmet weight and clipped to $[-1, 1]$.

The image-quality decomposition instead uses total rubric weight W: gain is G/W and loss is L/W. Their difference matches exported RawQualityDelta for every scored edit. SignedEditScore is multiplied by 100 below; quality quantities are in points. Negative-edit rates pool scored executions; within-image variation does not create additional independent sampling units.

Table C15: Complete Editing scores and quality decomposition.

Model	SignedEditScore [95% CI]	Gain G/W	Loss L/W	Net quality change	Negative edits %	Scored edits / 600
SenseNova-U1.5	28.02 [19.27, 36.57]	5.24	2.49	2.75	18.83	600
FLUX.1-Kontext-dev	-2.55 [-9.14, 3.70]	2.27	2.75	-0.48	21.83	600
SenseNova-U1	-7.16 [-14.64, 0.003]	3.04	5.11	-2.07	37.23	599
FLUX.2-klein-9B	-10.26 [-18.89, -2.02]	2.60	5.47	-2.88	35.00	600
U1.5-Preview	-14.13 [-22.43, -5.64]	3.65	6.18	-2.53	46.33	600
Emu3.5	-15.95 [-24.90, -7.39]	3.49	8.79	-5.30	43.24	599
FLUX.2-klein-4B	-16.07 [-24.05, -7.96]	2.16	4.99	-2.82	37.73	599
FLUX.2-dev	-18.89 [-27.56, -10.04]	3.16	10.22	-7.05	43.41	599
Qwen-Image-Edit-2511	-19.89 [-28.26, -11.18]	2.67	8.54	-5.87	42.50	600
Qwen-Image-Edit-2509	-30.70 [-39.00, -22.72]	2.49	10.19	-7.70	50.08	599
BAGEL	-37.43 [-45.14, -28.96]	2.53	12.95	-10.42	58.33	600
LongCat-Image-Edit	-37.91 [-44.66, -31.13]	1.64	8.61	-6.97	54.00	600
HiDream-O1-Image	-42.53 [-49.30, -35.87]	1.16	10.08	-8.92	58.26	599
Step1X-Edit-v1p2	-46.04 [-53.81, -38.46]	2.26	16.50	-14.25	60.83	600
Lumina-DiMOO	-47.43 [-55.26, -39.34]	1.60	17.65	-16.05	59.00	600
Lance_3B	-59.46 [-68.18, -50.53]	2.25	24.84	-22.60	75.00	600
Ovis-U1	-59.59 [-68.24, -50.70]	2.91	26.21	-23.30	75.59	598
OmniGen2	-64.87 [-71.59, -58.27]	1.89	27.72	-25.83	75.17	600

The U1 interval ends at +0.003 points (three decimals) and crosses zero, as does the Kontext interval.

C.5.2 Local Repair Conditions and Repeated Executions

Mean Before quality is 89.47/100; 137 images have one unmet requirement, 46 have two, and 17 have three. Figure C1 groups source images by the number of positive SignedEditScore outcomes across three edits; zero and negative scores both count as non-positive. Only images with all three edits scored enter the 0/1/2/3 categories. Images with fewer scored edits are kept separate, never treating unscored executions as failures. Every bar still covers all 200 source images.

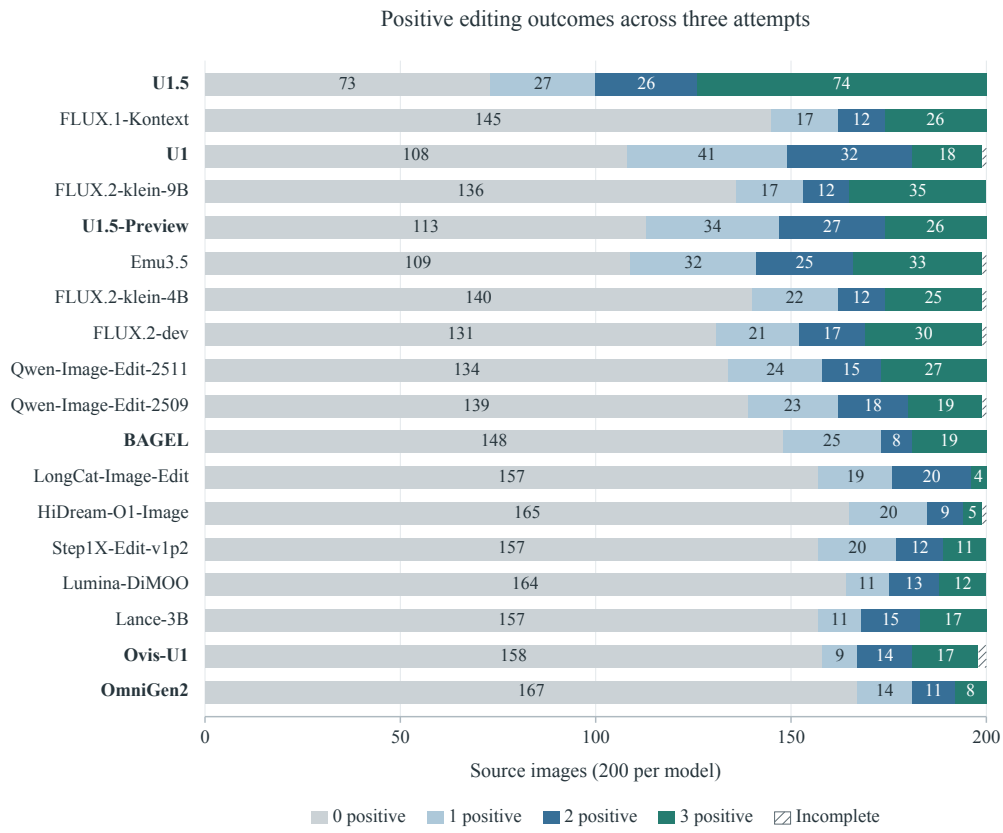


Figure C1: Positive edits across three executions per source. Models are ordered by mean SignedEditScore; bold labels mark UMMs. Numbers count sources; hatching marks incomplete scoring. U1.5 and Preview complete all 200 sources, with 0/1/2/3 positive counts of 73/27/26/74 and 113/34/27/26, respectively; 53 and 61 sources thus mix positive and non-positive outcomes. Variation includes generation and evaluation.

C.5.3 Fixed-Input Comparison of U1.5 and Preview

On the same 200 Before images, each model's three scored edits are averaged within the source, then released U1.5 and Preview are paired across sources. Changes are +1.60 [1.02, 2.17] in repair gain, −3.69 [−4.59, −2.84] in preservation loss, and +5.29 [4.21, 6.28] in net image quality, all in quality points; intervals use the paired source-image procedure in C.1.4. This comparison provides a fixed-input capability reference for the Editor replacement in Section 3.4. The two experiments measure repair under fixed instructions and final quality under autonomous refinement, respectively.

C.5.4 No-Edit Control

We use an Identity editor that returns its input image unchanged as a no-edit control, applying the same Editing evaluation procedure to test whether unchanged images receive repair or damage scores. Of 600 planned executions, 593 succeed, covering all 200 source images; seven editing-interface failures remain unscored. Pixel-level checks confirm that every successful output is identical to its input. Identical image pairs reuse their evaluations across the three executions, so the statistical units are 200 source images, not 593 independent Judge evaluations.

Table C16: No-edit control and two editing models. SignedEditScore is multiplied by 100; net quality change is reported in points. Every row weights the same 200 source images equally.

Model or control	Signed EditScore	Net quality change
Identity (unchanged input)	+2.81	+0.41
U1.5-Preview	−14.13	−2.53
SenseNova-U1.5	+28.02	+2.75

Of the 200 source images, 193 receive zero change scores. Across the remaining seven, eight of the 280 initially unmet requirements are incorrectly judged as repaired; all 2,450 initially satisfied requirements are judged as preserved. The erroneous explanations describe changes absent from the images, such as removing gradients or adding fills, revealing a small positive scoring bias.

Paired by source image, U1.5 exceeds Identity by 25.20 [17.03, 33.71] SignedEditScore points and 2.34 [1.36, 3.36] quality points. Excluding the seven affected source images in a sensitivity analysis, U1.5 still has a mean SignedEditScore of +26.47 and a net quality gain of +2.44 points. These results support net repair gains beyond the spurious gains seen without editing; the control also produces no false damage judgments. Paired intervals use 10,000 source-image bootstrap resamples with seed 42. The main results keep all scored sources without subtracting the control score.

C.6 Component Replacements: Paired Effects and Execution Checks

C.6.1 Configurations and Coverage

B is the U1.5 Preview baseline. C replaces its Controller with Qwen3.8-27B; G and E replace its Generator and Editor, respectively, with released SenseNova-U1.5-8B-MoT. Combined codes indicate the corresponding joint replacements. B reuses the Section 3.2 execution, with identical task-level scores; all seven mixed configurations are terminal. All eight configurations share the same task-data fingerprint and Judge settings. Mixed runs select BoN and SR; protocol-execution and scoring source hashes match the baseline, with code differences confined to configuration and suite files. Inference settings are in C.1, and the fixed-input Editor comparison is in C.5.3.

Table C17: Component configurations, coverage, and within-trajectory gains.

Config.	Scored SR / 200	Scored BoN / 200	Final − Init [95% CI]
B · Baseline	200	195	+0.11 [−1.43, 1.64]
C · Baseline + Controller	198	196	+16.41 [13.81, 19.16]
G · Baseline + Generator	200	198	-2.67 [-4.32, -1.10]
E · Baseline + Editor	200	197	+3.88 [2.40, 5.35]
CG · Baseline + Controller + Generator	197	199	+15.89 [13.36, 18.57]
CE · Baseline + Controller + Editor	197	198	+21.63 [19.17, 24.16]
GE · Baseline + Generator + Editor	200	197	+0.78 [-0.50, 2.13]
CGE · Baseline + Controller + Generator + Editor	199	198	+17.92 [15.32, 20.65]

Within-trajectory gains use each configuration's own scored tasks. B retains the Section 3.2 interval; the other seven configurations use seed-42 resampling as described in C.1.4.

C.6.2 Paired Effects and Gain Decomposition

Contrasts below use common scored tasks. For each row, Δ means the first configuration's value minus the second configuration's value, computed within each paired task before averaging. Each comparison replaces one role: $\Delta\mathrm{Final}$ measures the delivered score change, $\Delta\mathrm{Best}$ the change in

the best image score, and $\Delta(\text{Best} - \text{Final})$ the change in the delivery gap. A negative value in the last column means a narrower gap.

Table C18: All paired effects of single-role replacements.

Contrast	Paired tasks	ΔFinal [95% CI]	ΔBest [95% CI]	Δ(Best − Final) [95% CI]
C − B	198	+17.02 [14.50, 19.80]	+16.61 [14.58, 18.83]	-0.41 [-1.79, 1.02]
CG − G	197	+17.51 [14.97, 20.15]	+14.46 [12.41, 16.86]	-3.05 [-4.49, -1.48]
CE − E	197	+17.07 [14.79, 19.41]	+16.07 [14.00, 18.29]	-1.00 [-2.16, 0.20]
CGE − GE	199	+16.82 [14.44, 19.31]	+16.92 [14.81, 19.19]	+0.09 [-1.23, 1.41]
G − B	200	+3.34 [0.76, 5.88]	+5.11 [2.93, 7.22]	+1.77 [0.29, 3.37]
CG − C	195	+4.03 [2.09, 6.17]	+2.96 [1.27, 4.79]	-1.07 [-2.41, 0.26]
GE − E	200	+2.39 [0.05, 4.72]	+2.25 [0.14, 4.31]	-0.14 [-1.32, 1.04]
CGE − CE	196	+1.94 [-0.17, 4.14]	+2.87 [1.27, 4.62]	+0.93 [-0.44, 2.29]
E − B	200	+4.49 [2.98, 5.91]	+3.32 [2.19, 4.40]	-1.16 [-2.30, 0.05]
CE − C	195	+4.97 [2.79, 7.08]	+3.16 [1.38, 4.83]	-1.81 [-3.04, -0.40]
GE − G	200	+3.54 [1.81, 5.22]	+0.47 [-0.76, 1.73]	-3.07 [-4.37, -1.78]
CGE − CG	196	+2.78 [0.81, 4.77]	+3.09 [1.50, 4.64]	+0.31 [-0.94, 1.47]

Decomposition of delivered gains. Figure 5 computes $\Delta\text{Final} = \Delta\text{Init} + \Delta(\text{Best} - \text{Init}) - \Delta(\text{Best} - \text{Final})$ within each paired task and then averages each term. Here ΔInit is the change in initial image score. The comparisons use 198 common tasks for C−B and 200 each for G−B and E−B. The terms thus add exactly within each figure row. Changes in candidate gain are +15.89 [13.77, 18.34], −1.02 [−2.59, 0.42], and +2.60 [1.21, 3.96] points, respectively; contributions from narrowing the delivery gap are +0.41 [−1.02, 1.79], −1.77 [−3.37, −0.29], and +1.16 [−0.05, 2.30]. Intervals use the paired procedure in C.1.4; intervals for the candidate-gain change with Generator replacement and gap reductions with Controller and Editor replacement all include zero.

C.6.3 Actions and Image Calls

Best−Final uses all scored SR trajectories, including initial-only trajectories. Counts use exact inequalities. Image calls include initial generation; REFINE and REGENERATE are mean executed calls per scored trajectory. Fallback marks at least one invalid fallback. Count categories can overlap.

Table C19: Image improvements, delivery, and call behavior.

Config.	Best > Init	Final < Init	Best > Final	Image calls	REFINE	Regen.	Fallback tasks
B	106/200	74/200	100/200	3.535	2.480	0.055	10
C	167/198	33/198	89/198	3.995	1.677	1.318	0
G	83/200	98/200	121/200	3.420	2.350	0.070	12
E	132/200	60/200	94/200	3.240	2.175	0.065	8
CG	164/197	25/197	88/197	3.919	1.812	1.107	0
CE	178/197	14/197	60/197	3.904	1.701	1.203	0
GE	88/200	59/200	79/200	2.925	1.880	0.045	10
CGE	171/199	26/199	80/199	3.935	2.085	0.849	0

The common subset where C and B both use four image calls without fallback contains 138 requests, with a C−B Final difference of +19.05 [16.09, 22.09] points. This subset is selected after execution to examine the gap at equal call counts; it still includes changes in both feedback and action selection. The full paired C−B change in Best−Final is −0.41 [−1.79, 1.02], whose interval includes zero; a delivery gap remains in the observed results.

C.6.4 COMBINATIONS AND ORACLE-BON COMPARISONS

Each interaction below combines mean Final scores from four configurations on their common scored tasks. For example, CE−C−E+B measures the gain from replacing both Controller and Editor beyond the sum of their separate gains over B. All interaction intervals include zero. CGE−CE is +1.94 [−0.17, 4.14], whereas CGE−CG is +2.78 [0.81, 4.77].

Table C20: Component interactions.

Final interaction	**Common tasks**	**Interaction [95% CI]**
CE − C − E + B	195	+0.42 [-2.04, 2.77]
CG − C − G + B	195	+0.34 [-2.97, 3.80]
GE − G − E + B	200	-0.95 [-3.00, 1.25]

SR and ORACLE-BON are paired within each configuration's own run:

Table C21: SR versus ORACLE-BON within each configuration.

Config.	**Common tasks**	**SR Final − ORACLE-BON [95% CI]**
B	195	-10.27 [-12.05, -8.50]
C	194	+6.72 [4.49, 9.16]
G	198	-12.19 [-14.08, -10.22]
E	197	-6.19 [-7.94, -4.45]
CG	196	+4.85 [2.90, 7.13]
CE	195	+11.66 [9.30, 13.87]
GE	197	-9.72 [-11.52, -7.95]
CGE	197	+7.12 [4.98, 9.37]

C.6.5 SCORE CONSISTENCY FOR SHARED INITIAL IMAGES

Recorded initial-image SHA-256 hashes match for all common tasks within B/C/E/CE and within G/CG/GE/CGE. However, evaluation caches are not unified across these independent runs, so identical images can have different saved scores. The table uses B and G as within-group references; all score differences are quality points.

Table C22: Cross-run score differences for shared initial images.

Contrast	**Same initial / common tasks**	**Different Init scores**	**Mean Init difference**	**Mean absolute Init difference**
C − B	198/198	104	+0.72	4.88
E − B	200/200	108	+0.72	4.52
CE − B	197/197	113	+0.18	4.64
CG − G	197/197	87	-0.77	4.15
GE − G	200/200	100	+0.09	4.24
CGE − G	199/199	98	-0.14	4.44

These Init differences reflect cross-run judgments of identical images; the initial images are unchanged. The analysis keeps each formal judgment under the within-run reuse rule in C.1.2, without unifying cross-run scores after the fact. The intervals remain conditional on these scores.

LIMITATIONS

The results cover four fixed task packages and the models with scored outputs. They do not establish performance on untested domains or error types. Rubric completeness and annotation quality determine what the benchmark can measure. The no-edit control checks erroneous repair and damage

judgments in Editing, while the shared-image analysis documents cross-run score variation; neither replaces representative human validation of the Judge. Intervals describe task-sampling uncertainty conditional on recorded executions and scores. Matched image operations do not imply matched latency or total compute, and ORACLE-BON uses privileged external scores for selection. Controller replacement changes diagnosis, planning, action choice, and generation instructions together, so its aggregate effect does not isolate one internal capability. Planning is evaluated through the autonomous process rather than a separate diagnostic score.

D A WORKED RUBRIC SCORING EXAMPLE

We illustrate image-quality scoring and SignedEditScore with one recorded edit by SenseNova-U1.5-Preview (seed 42) on `preservation_125`. The repair instruction asks the Editor to move the activation peak and vertical dotted line to $\tau_c = 0.65$, keep the peak value at 38, and preserve unaffected content. The rubric contains 16 requirements with total weight $W = 1.00$.

Evaluation inputs. Qwen3.8-Flash-Next receives the ordered Before/After images, repair instruction, full rubric, and fixed Before labels, and judges only After. The Editor receives the request and repair instruction, but no rubric or labels. Table D1 reproduces five requirements and summarizes the remaining eleven.

Table D1: Rubric items and recorded judgments. Weights are unscaled; $\Delta w_j = w_j(y_j^a - y_j^b)$. Only PASS contributes $y_j = 1$. The last row aggregates eleven preserved items and is not a separate requirement.

ID	Original requirement	w_j	Before	After	Δw_j
`sp2a`	Does the 'Binary Fusion' bar show a detection rate of 0.82?	0.08	PASS	PASS	0.00
`sp4c`	Is the y-axis labeled 'Number of Activations'?	0.06	PASS	FAIL	−0.06
`sp5`	Does the true activations curve peak at exactly 38 when τ_c equals 0.65?	0.08	FAIL	UNCERTAIN	0.00
`sp6`	Is there a vertical dotted line at $\tau_c = 0.65$?	0.08	FAIL	PASS	+0.08
`sp10`	Does the chart use a clean grayscale palette with one accent color for the probabilistic bar and true-activation curve?	0.08	PASS	FAIL	−0.08
Other 11	All remaining requirements.	0.62	PASS	PASS	0.00

From labels to repair and loss. Initially unmet items `sp5` and `sp6` define the repairable weight F. Only `sp6` becomes PASS, contributing repaired weight G. The axis label (`sp4c`) and chart style (`sp10`) change from PASS to FAIL, contributing lost weight L. UNCERTAIN contributes no passing weight but remains distinct from a confirmed failure in the record. Thus

$$F = 0.08 + 0.08 = 0.16, \qquad G = 0.08, \qquad L = 0.06 + 0.08 = 0.14. \tag{D.1}$$

Example Judge evidence. For `sp4c`, the saved Judge output records `verdict: fail` and `confidence: 0.9`, with this evidence:

> "Region: right y-axis label; BEFORE: 'Number of Activations'; AFTER: vertical text reads 'Number of Aclvators' (corrupted word); CHANGE: new textual corruption of the axis label; required exact label no longer present."

Confidence is metadata and does not multiply the rubric weight. All judgments here are retained from the recorded evaluation.

Final scores. The fixed Before labels give $S(I^b) = 0.84$. Using Equations (1) and (4),

$$\begin{aligned} S(I^a) &= \frac{0.84 + 0.08 - 0.14}{1.00} = 0.78, \qquad \Delta Q = 0.78 - 0.84 = -0.06, \\ \text{SignedEditScore} &= \text{clip}\left(\frac{0.08 - 0.14}{0.16}, -1, 1\right) = -0.375. \end{aligned} \tag{D.2}$$

On the displayed scales, image quality falls from **84.0 to 78.0** (−6.0 quality points), and SignedEditScore is −37.5. The edit repairs one target, but damage to previously correct content outweighs that repair.