BEYOND PLAUSIBILITY: VERIFIABLE FINE-GRAINED IMAGE EDITING ON STRUCTURED ASSETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-grained image editing requires more than producing a visually plausible result: an editor must execute the requested attribute change precisely while leaving everything else intact. However, existing benchmarks leave a critical gap between realism and verifiability: benchmarks built on realistic images typically rely on human or vision–language model judgments, while deterministic evaluation has largely focused on synthetic shape canvases, with application-oriented extensions primarily limited to charts. This makes it difficult to determine precisely how much of a requested edit was executed, where unintended changes occurred, and whether small differences between models reflect genuine editing capability or evaluator uncertainty. To bridge this gap, we present **VeriEdit-Bench**, a benchmark for fine-grained, instruction-faithful image editing across realistic structured assets with deterministic, four-axis evaluation. Its 1,740 cases are compiled from the source code of 153 Scalable Vector Graphics (SVG) graphics, charts, web interfaces, and presentation slides. Controlled source-code edits preserve the original visual context while yielding exact target images, pixel-level edit masks, and explicit edit specifications, enabling reproducible scoring along four axes: edit fidelity, preservation, localization, and magnitude. Evaluating eleven editors, we find that even the strongest model remains far from full credit; rankings for the same recoloring operation reverse between charts and SVG graphics; and outputs with similar pixel-accuracy profiles can still differ substantially in localization and change magnitude. This decomposition yields graded, verifiable feedback and exposes model-specific capability and failure profiles that holistic scores or evaluator-dependent judgments may obscure.

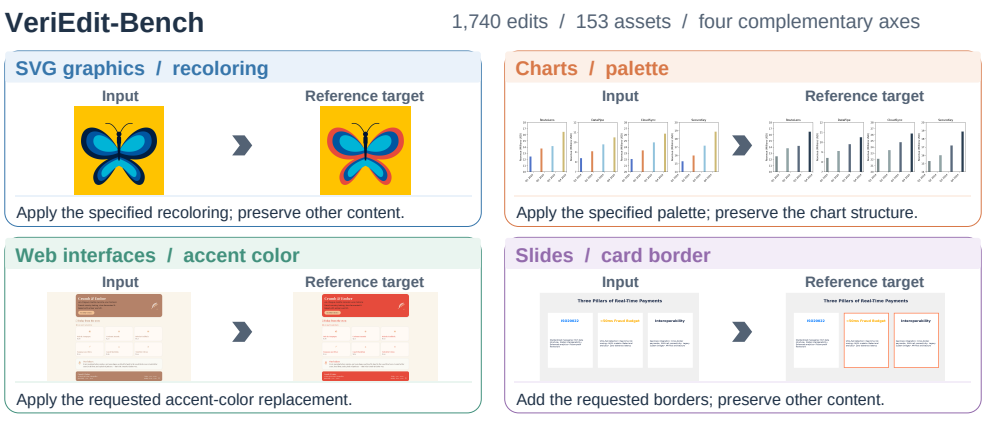


Figure 1: Overview of VeriEdit-Bench. Examples from SVG graphics, charts, web interfaces, and slides; each task specifies a controlled edit that preserves the surrounding context.

1 INTRODUCTION

Image editing has rapidly evolved from coarse semantic manipulation toward instruction-following systems capable of modifying increasingly specific visual attributes while preserving the surround-

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.

ing content. This progress has motivated a parallel line of work on image editing benchmarks. Early and widely used benchmarks evaluate editing models on real images through human judgment, perceptual similarity, or vision–language models (VLMs), covering object addition, removal, replacement, and attribute modification (Zhang et al., 2023; Basu et al., 2023; Ma et al., 2024). More recent benchmarks broaden this scope toward complex, multi-turn, and reasoning-intensive editing (Ye et al., 2025; Yang et al., 2025b; Han et al., 2025). Meanwhile, increasingly sophisticated VLM-based evaluators decompose editing quality into fine-grained factors such as instruction fidelity, preservation, localization, and perceptual quality (Liu et al., 2026). Together, these benchmarks have substantially improved the coverage and interpretability of image editing evaluation.

Despite this progress, a central challenge remains in combining *realism* with *verifiability*. Existing benchmarks already provide reference images, spatial grounding, and fine-grained assessments of edit correctness and content preservation (Zhang et al., 2023; Qian et al., 2025). For open-ended edits of realistic images, however, multiple outputs may satisfy the same instruction, and evaluation commonly relies on human judgments, learned perceptual measures, or VLM judges. Such scores do not necessarily provide direct, deterministic verification against an exact reference, so it remains difficult to determine precisely how much of a requested change was executed, where unintended changes occurred, or whether small differences between models reflect capability or evaluator uncertainty (Kumar et al., 2026).

Deterministic benchmarks offer an important step toward direct verification. PaintBench constructs precisely specified operations and evaluates their outputs against pixel-level ground truth, primarily on synthetic shape canvases, with application-oriented extensions to charts (Xu et al., 2026). StructBench extends evaluation to structured visual generation and editing, assessing factual correctness through VLM-based questions (Zhuo et al., 2025). PaintBench establishes deterministic verification of specified edits, including a chart extension; StructBench uses rendered targets to assess structured facts primarily through VLM questions. What remains underexplored is how target execution and unintended changes vary across realistic structured assets under a common deterministic protocol.

We introduce **VeriEdit-Bench** to bridge this gap. Our key insight is that structured visual assets provide editable source representations that explicitly define their visual attributes—a property typically unavailable in natural-image editing. Rather than synthesizing isolated geometric scenes, we construct 1,740 editing cases by editing 153 constructed source assets: Scalable Vector Graphics (SVG) graphics, charts, web interfaces, and slides. Each edit modifies a specified attribute while preserving the remaining source context. Rendering the original and edited sources yields exact input–target pairs and pixel-level edit masks, accompanied by explicit edit specifications. The source is used only to construct and score tasks; the evaluated editor receives the rendered image and the instruction alone, so the benchmark measures raster editing rather than code manipulation. These annotations enable deterministic evaluation along four complementary axes: *edit fidelity*, whether the requested result is achieved; *preservation*, whether unrelated content remains intact; *localization*, whether modifications are confined to the intended region; and *magnitude*, whether their extent matches the requested change. This formulation turns fine-grained editing into directly verifiable measurements, providing graded feedback even when an edit is only partially successful.

Finally, evaluating eleven contemporary image editors shows that precise editing remains challenging: the best open-weight model reaches a composite of 22 out of 100. The same recoloring operation ranks two models in opposite order on charts and SVG graphics; outputs with nearly identical PaintBench accuracy curves differ sharply in localization and magnitude; and strong target-color attainment coexists with damage to surrounding content. Controlled pixel- and source-level perturbations confirm that each axis responds to the error it is designed to detect. VeriEdit-Bench thus provides a diagnostic account of both the capabilities that current models already possess and the failures that still prevent faithful execution.

Our contributions are summarized as follows:

- We introduce **VeriEdit-Bench**, a benchmark for fine-grained, instruction-faithful image editing containing 1,740 cases constructed from 153 SVG graphics, charts, web interfaces, and presentation slides. By applying controlled edits to source representations, each case provides an exact target image, pixel-level edit mask, and explicit edit specification.
- We propose a deterministic four-axis evaluation framework that decomposes image editing performance into *edit fidelity*, *preservation*, *localization*, and *magnitude*, enabling repro-

ducible and graded measurement of partially correct edits without relying on human or VLM judgments for the core evaluation, and validate its response with controlled pixel-level and source-level perturbations.

- We conduct a systematic evaluation of eleven contemporary image editing models and reveal distinct capability and failure profiles that can be obscured by aggregate or evaluator-dependent scores, providing diagnostic evidence for where current models succeed and where fine-grained editing remains challenging.

2 Related Work

2.1 Benchmarks and Evaluation for Image Editing

Image editing benchmarks have expanded both task coverage and evaluation granularity. MagicBrush (Zhang et al., 2023) provides human-annotated editing triplets, while I^2EBench (Ma et al., 2024) introduces multidimensional evaluation validated against human perception. ImgEdit (Ye et al., 2025) and Complex-Edit (Yang et al., 2025b) extend evaluation to multi-turn editing and compositional instruction complexity, respectively. GIE-Bench (Qian et al., 2025) separates functional correctness from preservation through visual question answering (VQA) and object-aware measurements. LMM4Edit (Xu et al., 2025) and factorized VLM judges (Liu et al., 2026) further provide fine-grained assessment of instruction fidelity, edit quality, and preservation. What these benchmarks do not provide is deterministic verification of what changed, where, and by how much against an exact reference. PaintBench (Xu et al., 2026) offers a complementary approach by evaluating precisely specified operations through deterministic pixel-level comparison, primarily on procedurally generated shape canvases, with extensions to charts. VeriEdit-Bench extends this measurement principle to fine-grained attribute changes embedded in realistic structured assets.

2.2 Generation and Editing of Structured Visuals

Research on structured visuals examines factual correctness, reasoning, and visual fidelity. StructBench (Zhuo et al., 2025) evaluates factual correctness through decomposed VLM questions, while DisciplineGen (Wang et al., 2026b) studies data construction and model training for academic and scientific visuals. PlanViz (Li et al., 2026) targets spatial and procedural reasoning; VDE Bench (Yi et al., 2026) and OCRGenBench (Zhang et al., 2025) emphasize text modification and document or structural fidelity. ChartSync (Yu et al., 2026a) uses programmatically rendered targets to study coupled changes in chart values and geometry, combining objective measurements with VLM-based judgments. FigEdit (Li et al., 2025) shares our specification-to-target rendering foundation, covering single, composite, and interactive chart edits. It combines image similarity, text recognition, and model-based evaluation; our focus is a common deterministic capability profile across asset types. Table 29 summarizes asset coverage, editing operations, and verification in related datasets.

Structured representations also provide an interface for generation and editing. SVGEditBench V2 (Nishina & Matsui, 2025), VectorEdits (Kuchar et al., 2026), and Vector-Bench (Gupta & Hebbar, 2026) evaluate SVG editing; Vector-Bench uses deterministic specifications to assess target attributes and protected structure. DiagramGenBenchmark and DiagramAgent (Wei et al., 2025), alongside VCG-Bench (Su et al., 2026), formulate diagram generation and refinement through code. For presentations, object-model-based editing (Jung et al., 2026) directly modifies structured slide representations, while I2E (Yu et al., 2026b) reconstructs an editable environment from image pixels. These approaches therefore edit through a structured representation, either given or recovered from pixels. VeriEdit-Bench instead evaluates editors that operate directly on the rendered image; the source is used only to construct and score tasks.

3 VeriEdit-Bench

VeriEdit-Bench measures four complementary aspects of precise editing: target appearance, preservation of unrelated content, spatial extent, and visual change magnitude. Each case supplies an input image I, instruction u, rendered target T, allowed edit region E, and source-edit parameters. The editor receives only (I, u) and returns O; the rest serves only to verify the requested transformation.

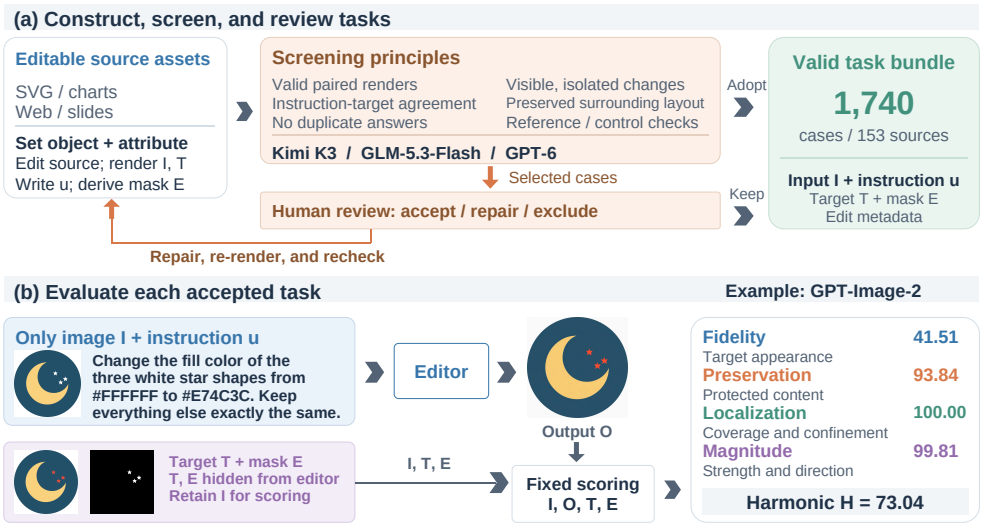


Figure 2: From editable sources to verifiable evaluation. (a) Source edits, large language model (LLM) screening, and author review curate tasks; repairs are re-rendered and rechecked. (b) The editor receives only the input image and instruction. Input, target, edit region, and output yield four scores and their harmonic mean, shown for a GPT-Image-2 output.

3.1 Dataset Construction

From editable sources to verifiable tasks. We construct tasks from SVG graphics, chart plotting programs, and HTML/CSS representations of web interfaces and slides. Adapted generation workflows and custom templates provide these editable assets; their sources and adaptations are documented in Appendix B.1. A controlled source edit fixes the requested attribute values and object scope. Rendering before and after the edit under identical settings gives I and T, while the instruction describes the visible objects, requested changes, and preservation requirements. Requested values include hexadecimal colors, spatial offsets, and multiplicative changes such as $1.5\times$ font size. The controlled source edit and fixed rendering settings jointly determine the visual reference. Edits may affect multiple objects, as in palette replacement. Tasks cover attribute edits of existing visible content, including appearance, visibility, and position; open-ended generation and content insertion are outside the benchmark scope.

Rendered change defines the allowed region. With d_{76} the CIELAB Euclidean distance, we define

$$E(x) = \mathbf{1}[d_{76}(I, T)(x) > 2]\,. \tag{1}$$

The region includes vacated and newly occupied pixels when an object moves; unchanged pixels remain protected. Its small threshold suppresses negligible rendering differences.

Curation for valid measurement. Each candidate must satisfy four checks: matching canvas dimensions, a visible requested change, preserved surrounding layout, and agreement between the instruction and the edited objects and parameters. LLM-assisted screening flags potential violations; all retained tasks are manually reviewed. Identified issues are resolved through source-level checks and verification of the rendered outcomes (Appendix B.1.1).

3.2 Benchmark Composition

The benchmark contains 1,740 cases from 153 sources: 48 SVG graphics, 25 charts, 51 web interfaces, and 29 slides. Each case pairs one source asset with one controlled edit specification, including the operation, affected object scope, and target parameter values; variants of the same operation differ in the requested edit values. Figure 3 shows their distribution across 22 asset–operation combinations. Asset-specific operations cover stroke styles, chart grids, web corner radii, and slide borders; shared operations such as recoloring support comparisons across asset types with the operation held fixed. Cases share source assets, with about eleven cases per source on average; accordingly, Sec. 4 resamples sources rather than individual cases.

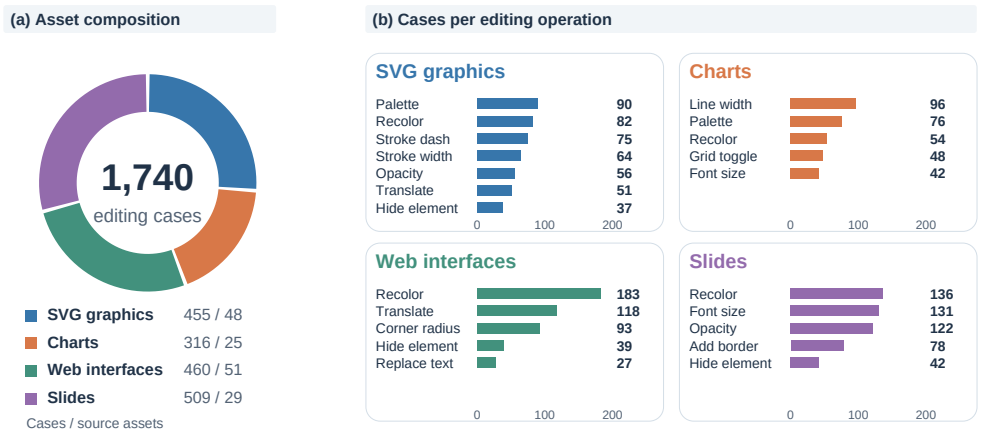


Figure 3: Dataset composition. Left: case shares and case/source counts. Right: operation counts, sorted within each asset type on a shared scale; instruction variants are grouped by operation.

3.3 Evaluation Protocol

Figure 2(b) shows the four-axis evaluation. Images are compared on the target canvas without content-based registration. All scores use fixed pixel-based rules and lie in $[0, 100]$.

For region R and color tolerances $\mathcal{K}$, define

$$Q_{\mathcal{K}}(A, B; R) = \frac{1}{|\mathcal{K}|} \sum_{t \in \mathcal{K}} \frac{1}{|R|} \sum_{x \in R} \mathbf{1}[d_{76}(A, B)(x) \le t]. \tag{2}$$

This averages color-agreement rates to provide graded credit. A shared binary check g screens for apparent edits explained by negligible change, noise, blur, or global color transformations. Failure sets fidelity, localization, and magnitude to zero while leaving preservation independently measured; passing does not establish agreement with the requested target. Preprocessing, checks, and boundary cases are described in Appendix B.2.

Fidelity. Fidelity measures improvement toward the target appearance in the edited region. After mild Gaussian smoothing and one-pixel mask dilation, let $b = Q_{\mathcal{K}_F}(\widetilde{I}, \widetilde{T}; E^+)$ be the unchanged-input baseline, with $\mathcal{K}_F = \{5, 10, 20\}$. We compute

$$F = 100g \ \mathrm{clip}_{[0,1]} \left(\frac{Q_{\mathcal{K}_F}(\widetilde{O}, \widetilde{T}; E^+) - b}{1 - b} \right). \tag{3}$$

Baseline subtraction prevents unchanged pixels from earning edit credit.

Preservation. Preservation measures agreement with the input outside the allowed region: $P = 100 Q_{\{2,5,10,20\}}(O, I; \overline{E})$, using unblurred images and the original mask. Thus, unrelated content can be preserved well even when the requested edit is executed poorly.

Localization. Localization measures coverage of the intended change and confinement to the allowed region. Let G and D be thresholded reference and output change maps, with small background components removed from D. Spatial recall and precision give

$$r_L = \frac{|D \cap G|}{|G|}, \qquad p_L = \frac{|D \cap E|}{|D \cap E| + \min(|D \setminus E|, 4|E|)}, \qquad L = 100g \, \frac{2 r_L p_L}{r_L + p_L}. \tag{4}$$

Capping outside-region area retains sensitivity to target coverage when the intended edit is small; preservation measures agreement over the entire protected region.

Magnitude. Magnitude compares the strength and direction of visual change on the target's effective-change core $C \subseteq E$. Let r_M be the ratio of output to target Lab change amplitudes from I, and c the cosine between their per-pixel Lab change fields. An amplitude agreement $a(r_M)$ gives full credit within 15% of the target and decays linearly outside this band (Appendix B.2):

$$M = 100g \left[\tfrac{1}{2} a(r_M) + \tfrac{1}{2} \frac{c + 1}{2} \right]. \tag{5}$$

Table 1: Performance on 1,740 cases (0–100, higher is better). Models are columns; the four axes are case-weighted means. Cell-mean H averages per-case harmonic composites within each of the 22 asset–operation cells and weights cells equally. CI denotes the 95% bootstrap confidence interval for cell-mean H. **Bold**: best overall; underlined: best open-weight, per metric.

Metric	GPT-Image-2	FireRed-Edit	Qwen-Image-Edit-2511	Qwen-Image-Edit	FLUX.2-dev	BAGEL	Step1X-Edit	FLUX.1-Kontext	FLUX.2-klein	OmniGen2	Instruct Pix2Pix
Fidelity F	**37.10**	13.42	12.22	9.38	8.16	6.66	5.24	3.78	4.85	5.16	2.36
Preservation P	84.52	85.09	87.16	82.06	82.61	68.75	82.78	**91.64**	78.81	60.52	39.19
Localization L	**56.92**	44.29	46.64	44.00	43.30	42.38	40.09	28.16	38.81	35.52	42.30
Magnitude M	**84.09**	68.76	66.92	60.53	61.65	61.38	59.47	46.69	60.34	56.80	64.16
Cell-mean H	**47.13**	21.85	20.47	16.26	13.11	10.81	9.78	8.78	8.42	7.32	3.83
CI lower	44.95	20.28	18.97	14.85	11.68	9.49	8.44	7.63	7.21	6.11	3.09
CI upper	49.05	23.29	21.85	17.64	14.44	12.28	11.20	10.00	9.69	8.56	4.69

It measures visual change relative to the reference rather than a source parameter such as font size.

Capability profiles and composite score. The capability profile (F, P, L, M) records performance on each dimension. Their harmonic mean $H = 4/(F^{-1} + P^{-1} + L^{-1} + M^{-1})$ summarizes joint performance, with $H = 0$ if any axis is zero. H is the default summary of joint performance, while the individual axes retain partial credit and diagnose specific aspects of the output. For example, an unchanged input scores $(0, 100, 0, 0)$.

4 Experiments

4.1 Experimental Setup

We evaluate eleven editors. Ten open-weight models: FireRed-Edit (Qiao et al., 2026), FLUX.2-dev and FLUX.2-klein (Black Forest Labs, 2025), FLUX.1-Kontext (Black Forest Labs et al., 2025), Qwen-Image-Edit and Qwen-Image-Edit-2511 (Wu et al., 2025a)[1], BAGEL (Deng et al., 2025), Step1X-Edit (Liu et al., 2025), OmniGen2 (Wu et al., 2025b), and InstructPix2Pix (Brooks et al., 2023); GPT-Image-2 (OpenAI, 2026a) provides a closed-source comparison.

We evaluate one output per model and case using model-specific inference settings. Images are centered on a white square canvas with their aspect ratio preserved; outputs are mapped back to the reference canvas for scoring. Appendix B.4 specifies the checkpoints, processing resolutions, and coordinate mapping. All analyses use the same model outputs. The primary score, cell-mean H, averages per-case composites within each asset–operation cell and weights the 22 cells equally; individual-axis entries in Table 1 average cases equally. Confidence intervals for H come from 10,000 source-level bootstrap resamples within each asset type, keeping all operations of a source together and sharing draws across models, reported as 95% bias-corrected and accelerated (BCa) intervals (Appendix A.13). Seed and instruction-wording controls are reported in Appendix A.8.

Scorer validation. We check that each axis responds to the error it is designed to detect. On all 1,740 cases, the rendered target scores $(100, 100, 100, 100)$ and the unchanged input $(0, 100, 0, 0)$. Controlled perturbation results appear in Appendix A.6. Constructed source-level editing errors give complementary mean responses: omitting one color group lowers F to 49 and L to 70 while preservation stays at 100; editing an extra group keeps $F = 96$ but lowers P to 72; and swapping target colors keeps $L \approx 100$ but drops F to 3. Each failure mode is thus visible on the axis that measures it.

4.2 Main Results

Precise editing remains challenging. Table 1 ranks GPT-Image-2 first with a cell-mean H of 47.1, more than twice the best open-weight model, FireRed-Edit (21.9), followed by Qwen-Image-Edit-2511 and Qwen-Image-Edit. Paired bootstrap intervals separate every adjacent pair among the six leading models and OmniGen2 from InstructPix2Pix, whereas the four adjacent comparisons between BAGEL and OmniGen2 remain unresolved (Appendix Table 27); a case-weighted compar-

[1]Qwen-Image-Edit-2511 model and configuration: `https://huggingface.co/Qwen/Qwen-Image-Edit-2511`.

ison appears in Appendix A.1. Composites are limited mainly by fidelity: open-weight F ranges from 2.4 to 13.4 while P exceeds 78 for all but three models, and between 53% and 85% of open-weight outputs receive $H = 0$. Across all models, the zero-composite breakdown shows that 87% arise from outputs that pass the eligibility check and change the region measurably ($L, M > 0$) yet do not raise agreement with the target above the unchanged-input baseline; the check itself accounts for 11% (Appendix Table 12). The benchmark thus separates substantial gaps from uncertain adjacent rankings, while even the strongest editor scores below half of the attainable composite.

Capability depends on the requested operation. The operation breakdown (Appendix Figure 8) shows that FireRed-Edit edits web visibility well but struggles with corner radii; GPT-Image-2 likewise combines strong SVG recoloring with weak web corner-radius control. Both perform less well on chart font-size edits than on their strongest operations. Appendix A.2 gives the corresponding four-axis profiles. Comparing the two Qwen-Image-Edit checkpoints likewise reveals larger gains for chart line width and web visibility than for slide typography or web corner radii (Appendix A.12). Overall scores alone obscure operation-specific strengths and gains across model versions.

4.3 Same Operation, Different Assets

Setup. To test whether performance on the same editing operation generalizes across asset types, we evaluate recoloring on 455 cases spanning SVG graphics, charts, web interfaces, and slides. We report case-mean H for each asset type with 95% confidence intervals obtained by resampling source assets (Appendix A.9).

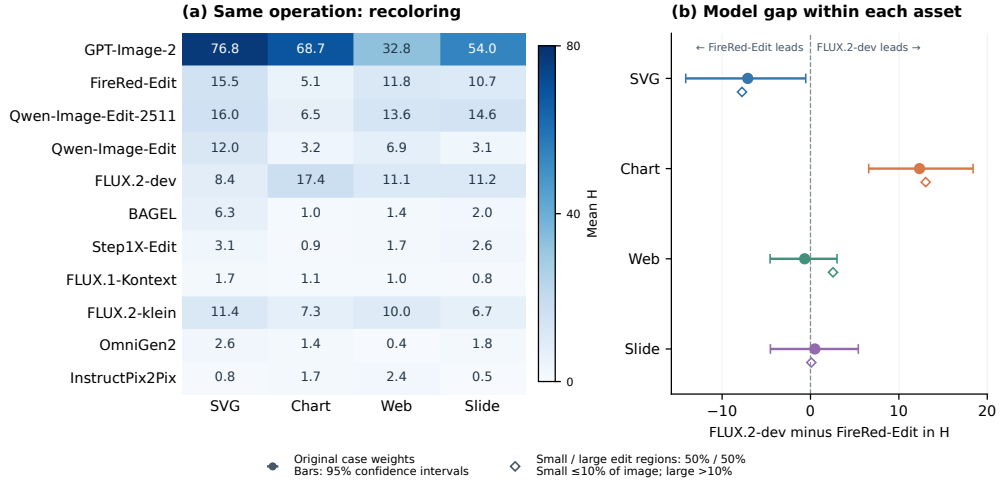
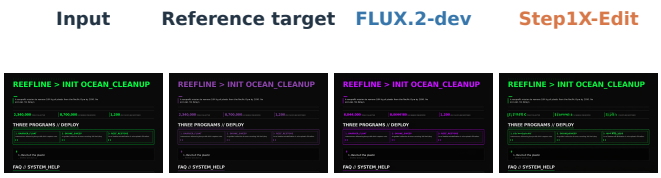


Figure 4: Fixed-operation comparison on 455 recoloring cases and 5,005 outputs from Table 1. Left: all eleven models' mean H by asset type. The color bar shows the shared score scale. Right: the FLUX.2-dev–FireRed-Edit gap with source-group bootstrap intervals; hollow diamonds give the gap after equally weighting edits covering at most 10% and more than 10% of the image.

Results and analysis. Figure 4 shows a ranking reversal: FLUX.2-dev outperforms FireRed-Edit on charts but trails it on SVG. Both gap intervals exclude zero, and equal weighting of small and large edit regions preserves the reversal. GPT-Image-2 also performs better on charts than on web interfaces. Although chart rankings broadly correlate with other asset rankings, they miss these pairwise and absolute differences. Appendix A.9 gives further cross-asset comparisons; Appendix A.4 examines spatial precision across edit-region sizes. A chart result cannot substitute for measuring the same attribute in other structured assets. These differences characterize the observed collections; they do not isolate layout complexity as a cause.

4.4 Distinct Failures behind Matched Pixel-Accuracy Profiles

Setup. PaintBench deterministically measures target-pixel correctness inside the edit region and preservation outside it against a known reference, using mean intersection over union (mIoU) and accuracy curves across color tolerances. Our profile additionally quantifies where and how much the output changes relative to the input, separately from target appearance. To test whether these measurements expose failures that pixel accuracy alone does not, we score the same outputs of all



Scores (0–100)	FLUX.2-dev	Step1X-Edit
PaintBench		
mIoU	0.04	0.03
Edit accuracy	0.09	0.06
Preservation accuracy	84.47	84.06
VeriEdit-Bench		
Fidelity	0.00	0.00
Preservation	93.69	93.29
Localization	**82.56**	**60.94**
Magnitude	**97.38**	**59.02**

PaintBench accuracy: FLUX.2-dev–Step1X-Edit gaps ≤2 points at every tolerance.

Figure 5: Complete images and per-output scores. PaintBench rows average CIE76 tolerances 0–10; shading highlights localization and magnitude differences. Larger views and full accuracy curves appear in Appendix Figures 19 and 20.

eleven models with both scorers on a 607-case panel (one case per source–operation group, all 153 sources) and select same-case model pairs whose PaintBench mIoU differs by at most one point and whose two accuracy curves differ by at most two points at every tolerance (Appendix A.11).

Results and analysis. Of 33,385 same-case pairs, 416 satisfy this matching rule, and 223 of them differ by at least 20 points in L or M, spanning 68 cases and 56 sources (Appendix Table 24). Figure 5 illustrates the distinction with a matched pair from the 101-case diagnostic panel (Appendix A.11): FLUX.2-dev recolors the requested page elements to a magenta-shifted purple rather than the requested #8e44ad, while Step1X-Edit leaves the target pixels green and repaints unrelated page regions. Both receive zero fidelity. FLUX.2-dev better matches the reference change magnitude and localizes its changes more precisely. The axis profiles distinguish these errors behind similar target-pixel accuracy. Further examples across asset types appear in Appendix Figures 10–13. Separately measuring target agreement and edit execution reveals which capability needs improvement, even when pixel-accuracy profiles nearly coincide.

4.5 Independent Analysis of Color Attainment

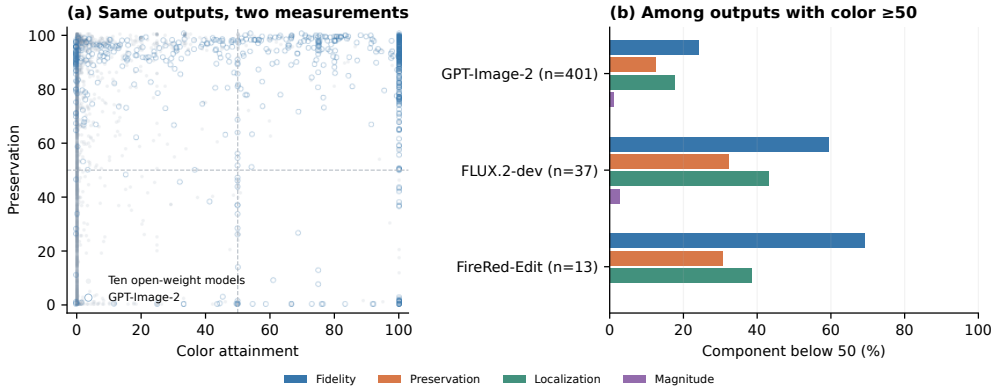


Figure 6: Attribute execution and precise editing on identical outputs. Left: all 6,831 eligible outputs, showing color attainment against preservation. Transparency reduces overlap; hollow points mark GPT-Image-2. Right: component scores below 50 among the color-score $\geq$ 50 outputs of GPT-Image-2, FLUX.2-dev, and FireRed-Edit; the denominators are printed beside their names. Components can overlap. Appendix A.10 reports every model and alternative color cutoffs.

Setup. The preceding analysis separates different failures behind similar pixel-accuracy profiles. We now ask whether strong attainment of a requested attribute can still coexist with other editing failures. We use color because color-editing tasks span all four asset types and target-color attainment is directly measurable. We check whether the same outputs also preserve unrelated content and confine changes to the intended region. For 621 eligible cases per model, CIEDE2000 measures target-color attainment within stable object regions (Appendix B.3). This diagnostic is separate from the four axes and does not enter H. We examine axis scores below 50 when color attainment is at least 50. These thresholds define descriptive slices of continuous scores, not a new success criterion.

Results and analysis. GPT-Image-2 leads color attainment (Table 2), whereas every open-weight model scores below 7. The criterion is strict: a core pixel counts only within a CIEDE2000 distance

Table 2: Independent color-attainment experiment on 621 color-editing cases. Values average the covered asset–operation cells (0–100). Color attainment is not a component of the four-axis composite. Model order matches Table 1; **bold**: best model; underlined: best open-weight model.

Metric	GPT-Image-2	FireRed-Edit	Qwen-Image-Edit-2511	Qwen-Image-Edit	FLUX.2-dev	BAGEL	Step1X-Edit	FLUX.1-Kontext	FLUX.2-klein	OmniGen2	Instruct Pix2Pix
Color attainment	**64.92**	2.98	4.65	2.25	6.41	1.00	0.85	0.26	3.40	0.60	0.45

of at most 5 of the requested color (Appendix B.3). Open-weight editors do alter the target regions, with nonzero localization and magnitude on 73–98% of these outputs, yet 63–85% of them receive zero color attainment (Appendix Table 22): the requested color is rarely reached. Among GPT-Image-2's outputs meeting the color threshold, 12.5% have low preservation and 17.7% have low localization (Figure 6). The pattern persists at a color cutoff of 80 and also occurs for FLUX.2-dev; complete counts and uncertainty estimates appear in Appendix A.10. Strong color attainment therefore does not by itself establish control over the full edit. Measuring it alongside preservation and localization reveals whether progress toward the requested attribute damages the existing design.

4.6 Aggregation and Scoring Sensitivity



Figure 7: Fixed-output ablations. Left: changing the aggregation of identical axis scores changes the reward for unbalanced edits. Right: rank correlation across all eleven models after changing the scoring configuration. Complete model scores and single-parameter sensitivity are in Appendix A.5.

Aggregation changes what the summary rewards. We compare harmonic, geometric, and arithmetic means of each output's four axes, followed by the same cell weighting (Figure 7). Arithmetic aggregation substantially raises scores for unbalanced edits, particularly for FLUX.1-Kontext, because strong components offset weak ones. The harmonic composite emphasizes joint performance, while the axis profile retains partial strengths.

The ranking is robust to scoring choices. Across the 44 single-parameter settings, the full eleven-model ranking remains unchanged in 40 settings. The remaining 4 settings change only lower-ranked positions; the leading five models never move. These variations concern close default scores whose adjacent comparisons are statistically unresolved (Appendix Table 27). Appendix A.5 reports all settings and model scores.

5 Conclusion

We introduced **VeriEdit-Bench** to bridge the gap between realism and verifiability in fine-grained image editing. Controlled source-code edits across SVG graphics, charts, web interfaces, and slides yield exact targets, edit masks, and specifications for deterministic evaluation of fidelity, preservation, localization, and visual change magnitude. Our evaluation of eleven editors shows that precise editing remains challenging. Similar pixel-accuracy profiles can conceal distinct failures, strong target-color attainment can coexist with poor preservation or localization, and model advantages can reverse across asset types for the same recoloring operation. VeriEdit-Bench supports reproducible diagnosis of capabilities and the contexts in which gains hold, beyond aggregate scores.

AI USE STATEMENT

The authors led method implementation and experimental design, with AI providing suggestions for their consideration. We used generative AI to assist with asset generation, data cleaning and screening, experimental analysis, and manuscript polishing. For dataset curation, Kimi K3, GLM-5.3-Flash, and GPT-6 assisted instruction and image checks. All retained tasks were manually reviewed, and reported issues were adjudicated (Appendix B.1.1).

The authors reviewed the AI-assisted code, experiments, analyses, and text, made the final scientific decisions, and take responsibility for all claims and artifacts in this work.

ETHICS STATEMENT

VeriEdit-Bench evaluates image editors on programmatically rendered structured assets. The authors perform task inspection and refinement for dataset quality control, while benchmark scores are computed automatically using fixed evaluation rules.

REPRODUCIBILITY STATEMENT

Appendix B documents construction, scoring, and inference settings; Appendix A provides supporting experiments. The supplementary material includes the scorer code computing the four axes and H, usage instructions, and an anonymized subset of cases; the complete benchmark and per-case scores will be released publicly upon acceptance.

REFERENCES

Samyadeep Basu, Mehrdad Saberi, Shweta Bhardwaj, Atoosa Malemir Chegini, Daniela Massiceti, Maziar Sanjabi, Shell Xu Hu, and Soheil Feizi. Editval: Benchmarking diffusion based text-guided image editing methods. *CoRR*, abs/2310.02426, 2023. doi: 10.48550/ARXIV.2310.02426. URL https://doi.org/10.48550/arXiv.2310.02426.

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

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 kontext: Flow matching for in-context image generation and editing in latent space. *CoRR*, abs/2506.15742, 2025. doi: 10.48550/ARXIV.2506.15742. URL https://doi.org/10.48550/arXiv.2506.15742.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 18392–18402. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01764. URL https://doi.org/10.1109/CVPR52729.2023.01764.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Shi Guang, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *CoRR*, abs/2505.14683, 2025. doi: 10.48550/ARXIV.2505.14683. URL https://doi.org/10.48550/arXiv.2505.14683.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and Trevor Darrell. Autopresent: Designing structured visuals from scratch. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 2902–2911. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00276. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ge_AutoPresent_Designing_Structured_Visuals_from_Scratch_CVPR_2025_paper.html.

Yug Aditi Gupta and Prannay Hebbar. Vector-bench: Can models surgically edit SVG code? *CoRR*, abs/2607.19056, 2026. doi: 10.48550/ARXIV.2607.19056. URL https://doi.org/10.48550/arXiv.2607.19056.

Feng Han, Yibin Wang, Chenglin Li, Zheming Liang, Dianyi Wang, Yang Jiao, Zhipeng Wei, Chao Gong, Cheng Jin, and Jiaqi Wang. Unireditbench: A unified reasoning-based image editing benchmark. *arXiv preprint arXiv:2511.01295*, 2025.

Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. Chartllama: A multimodal LLM for chart understanding and generation. *CoRR*, abs/2311.16483, 2023. doi: 10.48550/ARXIV.2311.16483. URL https://doi.org/10.48550/arXiv.2311.16483.

Kyudan Jung, Hojun Cho, Jooyeol Yun, Soyoung Yang, Jaehyeok Jang, and Jaegul Choo. Talk to your slides: High-efficiency slide editing via language-driven structured data manipulation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 3370–3399. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-ACL.166. URL https://doi.org/10.18653/v1/2026.findings-acl.166.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1mz0OyDz.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Josef Kuchar, Marek Kadlcík, Michal Spiegel, and Michal Stefánik. Vectoredits: A dataset and benchmark for instruction-based editing of vector graphics. In Stelios Piperidis, Núria Bel, Henk van den Heuvel, Nancy Ide, Simon Krek, and Antonio Toral (eds.), *Proceedings of the Fifteenth Language Resources and Evaluation Conference, LREC 2026, Palma de Mallorca, Spain, May 13-15, 2026*, pp. 11119–11124. ELRA Language Resource Association / ACL, 2026. doi: 10.63317/5GC5IBTB5K8I. URL https://doi.org/10.63317/5gc5ibtb5k8i.

Divake Kumar, Sina Tayebati, Devashri Naik, Ranganath Krishnan, and Amit Ranjan Trivedi. Vlm judges can rank but cannot score: Task-dependent uncertainty in multimodal evaluation. *arXiv preprint arXiv:2604.25235*, 2026.

Junxian Li, Kai Liu, Leyang Chen, Weida Wang, Zhixin Wang, Jiaqi Xu, Fan Li, Renjing Pei, Linghe Kong, and Yulun Zhang. Planviz: Evaluating planning-oriented image generation and editing for computer-use tasks. *CoRR*, abs/2602.06663, 2026. doi: 10.48550/ARXIV.2602.06663. URL https://doi.org/10.48550/arXiv.2602.06663.

Shawn Li, Ryan A. Rossi, Sungchul Kim, Sunav Choudhary, Franck Dernoncourt, Puneet Mathur, Zhengzhong Tu, and Yue Zhao. Charts are not images: On the challenges of scientific chart editing. *CoRR*, abs/2512.00752, 2025. doi: 10.48550/ARXIV.2512.00752. URL https://doi.org/10.48550/arXiv.2512.00752.

Runzhou Liu, Hailey Weingord, Sejal Mittal, Prakhar Dungarwal, Anusha Nandula, Bo Ni, Samyadeep Basu, Hongjie Chen, Nesreen K. Ahmed, Li Li, Jiayi Zhang, Koustava Goswami, Subhojyoti Mukherjee, Branislav Kveton, Puneet Mathur, Franck Dernoncourt, Yue Zhao, Yu Wang, Ryan A. Rossi, Zhengzhong Tu, and Hongru Du. Human-aligned MLLM judges for fine-grained image editing evaluation: A benchmark, framework, and analysis. *CoRR*, abs/2602.13028, 2026. doi: 10.48550/ARXIV.2602.13028. URL https://doi.org/10.48550/arXiv.2602.13028.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai,

Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1x-edit: A practical framework for general image editing. *CoRR*, abs/2504.17761, 2025. doi: 10.48550/ARXIV.2504.17761. URL https://doi.org/10.48550/arXiv.2504.17761.

Zimu Lu, Yunqiao Yang, Houxing Ren, Haotian Hou, Han Xiao, Ke Wang, Weikang Shi, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Webgen-bench: Evaluating llms on generating interactive and functional websites from scratch. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/6841eed8bb6a2ec49e49235c8115efee-Abstract-Datasets_and_Benchmarks_Track.html.

Yiwei Ma, Jiayi Ji, Ke Ye, Weihuang Lin, Zhibin Wang, Yonghan Zheng, Qiang Zhou, Xiaoshuai Sun, and Rongrong Ji. I2ebench: A comprehensive benchmark for instruction-based image editing. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/48fecef47b19fe501d27d338b6d52582-Abstract-Conference.html.

Kunato Nishina and Yusuke Matsui. Svgeditbench V2: A benchmark for instruction-based SVG editing. *CoRR*, abs/2502.19453, 2025. doi: 10.48550/ARXIV.2502.19453. URL https://doi.org/10.48550/arXiv.2502.19453.

OpenAI. Chatgpt images 2.0 system card. https://deploymentsafety.openai.com/chatgpt-images-2-0, 2026a.

OpenAI. GPT-6 Astra system card, September 2026b. URL https://deploymentsafety.openai.com/gpt-6-astra.

Yusu Qian, Jiasen Lu, Tsu-Jui Fu, Xinze Wang, Chen Chen, Yinfei Yang, Wenze Hu, and Zhe Gan. Gie-bench: Towards grounded evaluation for text-guided image editing. *CoRR*, abs/2505.11493, 2025. doi: 10.48550/ARXIV.2505.11493. URL https://doi.org/10.48550/arXiv.2505.11493.

Changhao Qiao, Chao Hui, Chen Li, Cunzheng Wang, Dejia Song, Jiale Zhang, Jing Li, Qiang Xiang, Runqi Wang, Shuang Sun, Wei Zhu, Xu Tang, Yao Hu, Yibo Chen, Yuhao Huang, Yuxuan Duan, Zhiyi Chen, and Ziyuan Guo. Firered-image-edit-1.0 technical report. *CoRR*, abs/2602.13344, 2026. doi: 10.48550/ARXIV.2602.13344. URL https://doi.org/10.48550/arXiv.2602.13344.

Xiaoyan Su, Peijie Dong, Zhenheng Tang, Song Tang, Yuyao Zhai, Kaitao Lin, Liang Chen, Yuhang Gai, Yuyu Luo, Qian Wang, and Xiaowen Chu. Vcg-bench: Towards A unified visual-centric benchmark for structured generation and editing. *CoRR*, abs/2605.15677, 2026. doi: 10.48550/ARXIV.2605.15677. URL https://doi.org/10.48550/arXiv.2605.15677.

Feiyu Wang, Jiayuan Yang, Zhiyuan Zhao, Da Zhang, Bingyu Li, Peng Liu, and Junyu Gao. Introsvg: Learning from rendering feedback for text-to-svg generation via an introspective generator-critic framework. *CoRR*, abs/2603.09312, 2026a. doi: 10.48550/ARXIV.2603.09312. URL https://doi.org/10.48550/arXiv.2603.09312.

Zhaokai Wang, Mingxin Liu, Zirun Zhu, Ziqian Fan, Yiguo He, Mohan Zhang, Leyao Gu, Xiangyu Zhao, Ning Liao, Shaofeng Zhang, Xuanhe Zhou, Zhihang Zhong, Junchi Yan, and Xue Yang. Disciplinegen-1m: A large-scale dataset for multidisciplinary visual generation and editing. *CoRR*, abs/2607.02290, 2026b. doi: 10.48550/ARXIV.2607.02290. URL https://doi.org/10.48550/arXiv.2607.02290.

Jingxuan Wei, Cheng Tan, Qi Chen, Gaowei Wu, Siyuan Li, Zhangyang Gao, Linzhuang Sun, Bihui Yu, and Ruifeng Guo. From words to structured visuals: A benchmark and framework for text-to-diagram generation and editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 13315–13325. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01243. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wei_From_Words_to_Structured_Visuals_A_Benchmark_and_Framework_for_CVPR_2025_paper.html.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming 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. *CoRR*, abs/2508.02324, 2025a. doi: 10.48550/ARXIV.2508.02324. URL https://doi.org/10.48550/arXiv.2508.02324.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Towards instruction-aligned multimodal generation. *CoRR*, abs/2506.18871, 2025b. doi: 10.48550/ARXIV.2506.18871. URL https://doi.org/10.48550/arXiv.2506.18871.

Ronghuan Wu, Wanchao Su, and Jing Liao. Chat2svg: Vector graphics generation with large language models and image diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 23690–23700. Computer Vision Foundation / IEEE, 2025c. doi: 10.1109/CVPR52734.2025.02206. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wu_Chat2SVG_Vector_Graphics_Generation_with_Large_Language_Models_and_Image_CVPR_2025_paper.html.

Kai Xu, Ellis Brown, Shrikar Madhu, Rob Fergus, He He, and Saining Xie. Paintbench: Deterministic evaluation of precise visual editing. *CoRR*, abs/2606.00188, 2026. doi: 10.48550/ARXIV.2606.00188. URL https://doi.org/10.48550/arXiv.2606.00188.

Zitong Xu, Huiyu Duan, Bingnan Liu, Guangji Ma, Jiarui Wang, Liu Yang, Shiqi Gao, Xiaoyu Wang, Jia Wang, Xiongkuo Min, Guangtao Zhai, and Weisi Lin. Lmm4edit: Benchmarking and evaluating multimodal image editing with lmms. In Cathal Gurrin, Klaus Schoeffmann, Min Zhang, Luca Rossetto, Stevan Rudinac, Duc-Tien Dang-Nguyen, Wen-Huang Cheng, Phoebe Chen, and Jenny Benois-Pineau (eds.), *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pp. 6908–6917. ACM, 2025. doi: 10.1145/3746027.3755294. URL https://doi.org/10.1145/3746027.3755294.

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, Gongye Liu, Xiaomei Nie, Deng Cai, and Yujiu Yang. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=sGpCzsfd1K.

Siwei Yang, Mude Hui, Bingchen Zhao, Yuyin Zhou, Nataniel Ruiz, and Cihang Xie. Complex-edit: Cot-like instruction generation for complexity-controllable image editing benchmark. *CoRR*, abs/2504.13143, 2025b. doi: 10.48550/ARXIV.2504.13143. URL https://doi.org/10.48550/arXiv.2504.13143.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025,*

NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/bec783f119b969772c35243b1c801090-Abstract-Datasets_and_Benchmarks_Track.html.

Hongzhu Yi, Yujia Yang, Yuanxiang Wang, Zhenyu Guan, Jiahuan Chen, Chenxi Bao, Tiankun Yang, Yixuan Yuan, Tianyu Zong, Xinming Wang, Tao Yu, Ruiwen Tao, Haijin Liang, Jin Ma, Jinwen Luo, Xinyu Zuo, and Jungang Xu. VDE bench: Evaluating the capability of image editing models to modify visual documents. *CoRR*, abs/2602.00122, 2026. doi: 10.48550/ARXIV.2602.00122. URL https://doi.org/10.48550/arXiv.2602.00122.

Jiakang Yu, Yixuan Chai, Tianci Wang, Rihui Jin, Guangkai Xu, Hongtao Deng, Xun Zhu, Wang Gao, Xinrun Guo, and Haipang Wu. Chartsync: A benchmark for visuo-logical cascading chart editing. *CoRR*, abs/2607.10301, 2026a. doi: 10.48550/ARXIV.2607.10301. URL https://doi.org/10.48550/arXiv.2607.10301.

Jinghan Yu, Junhao Xiao, Chenyu Zhu, Jiaming Li, Jia Li, HanMing Deng, Xirui Wang, Guoli Jia, Jianjun Li, Xiang Bai, Bowen Zhou, and Zhiyuan Ma. I2E: from image pixels to actionable interactive environments for text-guided image editing. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 44827–44850. Association for Computational Linguistics, 2026b. doi: 10.18653/V1/2026.ACL-LONG.2076. URL https://doi.org/10.18653/v1/2026.acl-long.2076.

Z.ai. GLM-5.3-Flash: Frontier intelligence, flash cost, August 2026. URL https://z.ai/blog/glm-5.3-flash.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/64008fa30cba9b4d1ab1bd3bd3d57d61-Abstract-Datasets_and_Benchmarks.html.

Peirong Zhang, Haowei Xu, Jiaxin Zhang, Xuhan Zheng, Guitao Xu, Yuyi Zhang, Junle Liu, Zhenhua Yang, Wei Zhou, and Lianwen Jin. Ocrgenbench: A comprehensive benchmark for evaluating ocr generative capabilities. *arXiv preprint arXiv:2507.15085*, 2025.

Le Zhuo, Songhao Han, Yuandong Pu, Boxiang Qiu, Sayak Paul, Yue Liao, Yihao Liu, Jie Shao, Xi Chen, Si Liu, and Hongsheng Li. Factuality matters: When image generation and editing meet structured visuals. *CoRR*, abs/2510.05091, 2025. doi: 10.48550/ARXIV.2510.05091. URL https://doi.org/10.48550/arXiv.2510.05091.

A Additional Experiments

This appendix provides detailed operation profiles, visual evidence, and supplementary comparisons for Section 4.

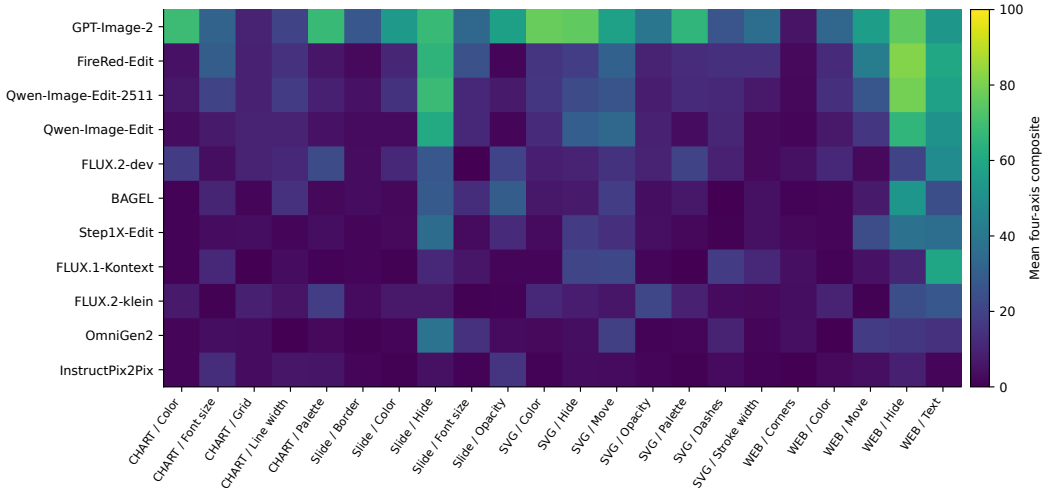


Figure 8: Four-axis composite scores for all eleven models across the 22 asset–operation combinations. Each cell averages the corresponding cases.

A.1 Effect of Case Weighting

Setup. We change only the weights assigned to per-case H. With n_c cases and mean $\overline{H}_c$ in cell c, cell weighting gives $\frac{1}{22}\sum_c \overline{H}_c$, whereas case weighting gives $\frac{1}{1740}\sum_c n_c \overline{H}_c$.

Table 3: Effect of case weighting on the same 1,740 cases. Both rows average per-case harmonic composites. Cell-mean H weights the 22 asset–operation cells equally; case-mean H weights individual cases equally. Model order follows Table 1; **bold** marks the best model and underlining the best open-weight model.

Metric	GPT-Image-2	FireRed-Edit	Qwen-Image-Edit-2511	Qwen-Image-Edit	FLUX.2-dev	BAGEL	Step1X-Edit	FLUX.1-Kontext	FLUX.2-klein	OmniGen2	Instruct Pix2Pix
Cell-mean H	**47.13**	21.85	20.47	16.26	13.11	10.81	9.78	8.78	8.42	7.32	3.83
Case-mean H	**44.90**	17.76	16.29	11.49	11.34	9.09	7.31	5.73	7.02	6.12	3.66

Results and analysis. Table 3 retains GPT-Image-2, FireRed-Edit, and Qwen-Image-Edit-2511 as the top three. Case weighting places FLUX.2-klein and OmniGen2 above FLUX.1-Kontext; the leading seven models retain their order.

The changes involve lower-ranked models whose adjacent default gaps are statistically unresolved (Table 27). Cell weighting prevents combinations with more variants from dominating the comparison, whereas case weighting reflects the benchmark's actual case distribution.

A.2 Per-operation Axis Profiles

Setup. We average each axis within every asset–operation cell for all eleven models (Figure 9).

Results and analysis. Separating the four axes identifies whether a low composite reflects weak target execution, preservation, localization, or change magnitude.

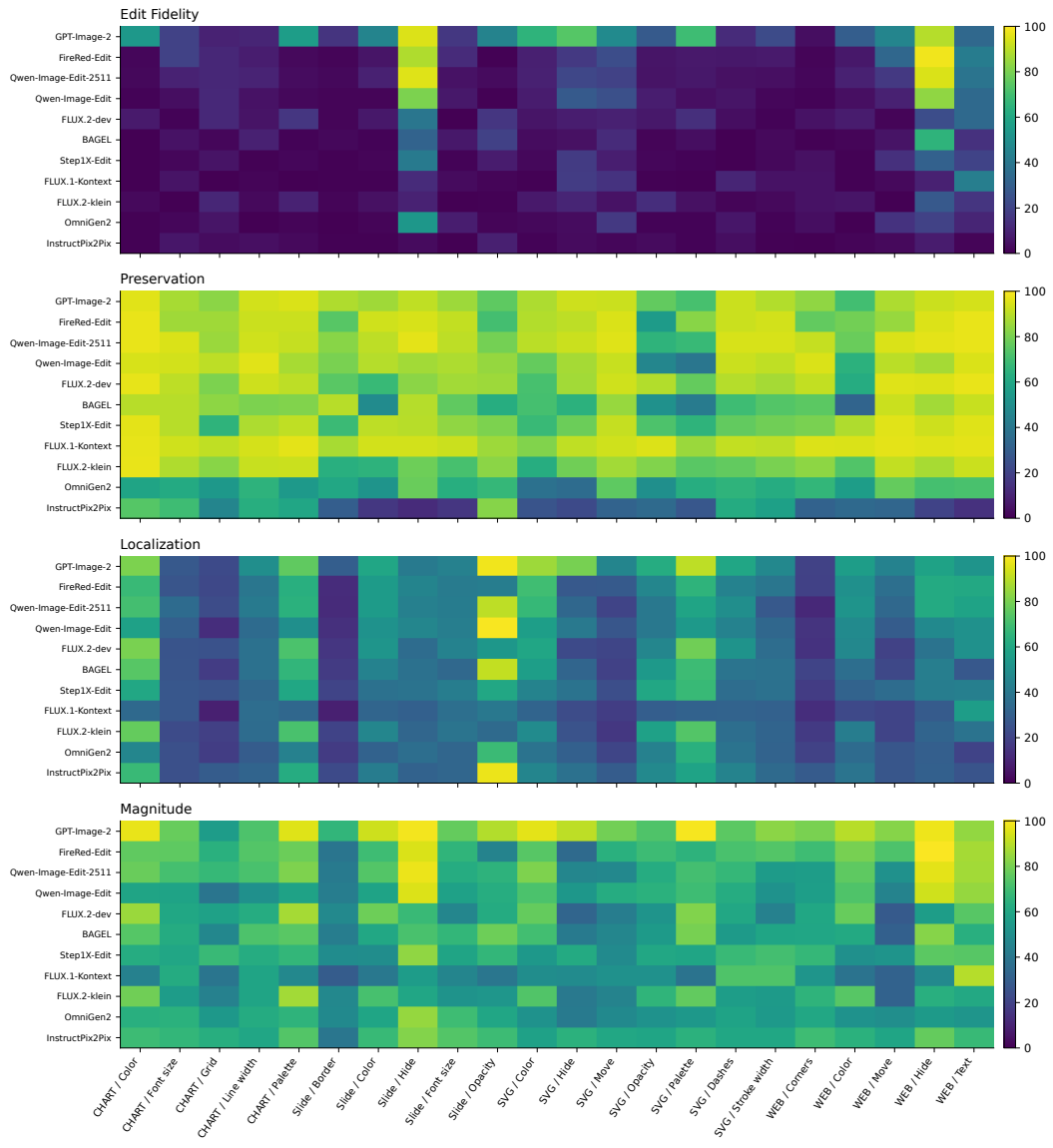


Figure 9: Full eleven-model axis profiles across all 22 asset–operation cells. Each panel averages one axis within each cell, with a shared 0–100 scale.

A.3 Complete Images for the Qualitative Analysis

Setup. Figures 10–13 share a common layout: input, target, and edit mask above three model outputs. Orange shows the evaluated mask without dilation. All scores use the complete images.

Results and analysis. The complete views place target execution and changes to protected content in the same visual context. For example, the web case distinguishes removal of the requested heading from retaining it (Figure 12).



Figure 10: SVG recoloring. Only the chimney and door should change color; the roof and surrounding content remain protected.

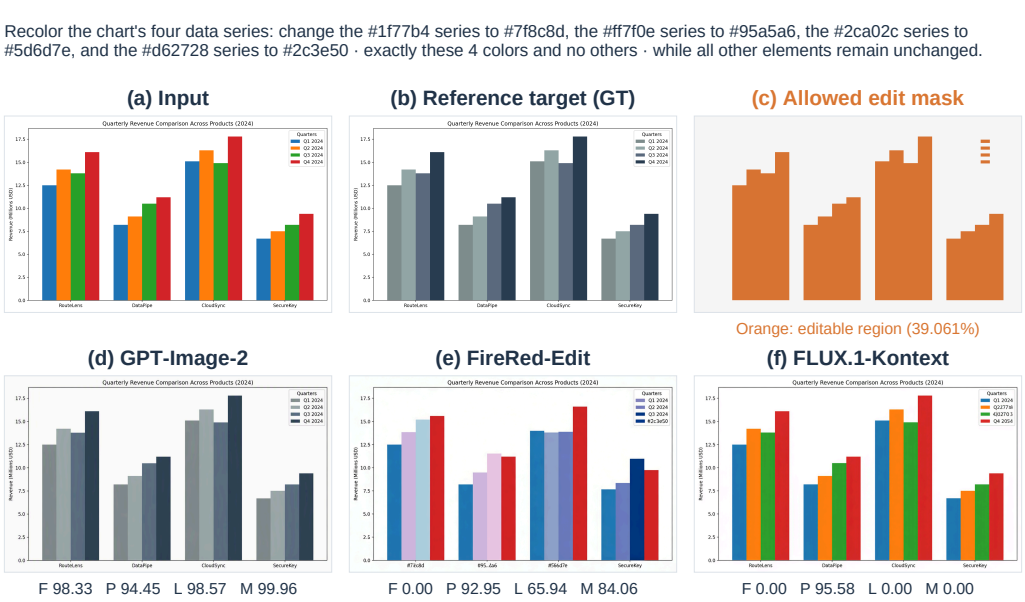


Figure 11: Chart palette change. The four bar series should take the new muted palette while axes, labels, and surrounding content remain protected. GPT-Image-2 applies the palette; FireRed-Edit produces a partial recolor with hex-code text artifacts; FLUX.1-Kontext leaves the chart unchanged.

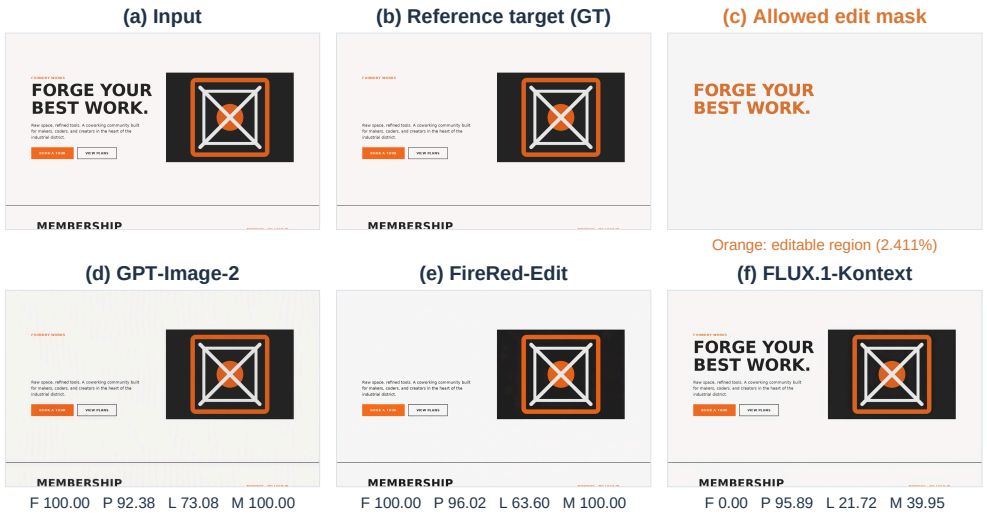


Figure 12: Web heading hiding. The specified heading should disappear while its space and all surrounding content remain unchanged. GPT-Image-2 and FireRed-Edit remove the heading; FLUX.1-Kontext retains it.



Figure 13: SVG element hiding. The large teal body should disappear while every other element remains unchanged. GPT-Image-2 removes it; FireRed-Edit instead recolors it; FLUX.1-Kontext leaves it in place.

A.4 Spatial Precision by Edit-region Size

Setup. We divide detected changed-pixel area by allowed edit-mask area, retaining zero-score outputs. Ratios above one imply changes outside the allowed region.

Results and analysis. The pooled medians are 84.28 for masks below 0.1%, 10.57 for 0.1–1%, and 1.00 for masks covering at least 25% of the image. Figure 14 breaks down the distributions and region-size effects by model, showing how much the pooled pattern varies across editors. Operation and content also vary with region size, so this comparison describes an association rather than an isolated effect of edit size.

A.5 Complete Aggregation and Scoring Ablations

Setup. We compare three within-case means with cell weighting fixed (Table 4) and vary color distance, scoring resolution, and tolerances on identical outputs (Table 5). CIEDE2000 replaces the shared CIE76 distance function without retuning thresholds; Lab-vector fits and magnitude calculations stay fixed. Scorer-resolution controls downsample the already generated predictions with Lanczos, then restore them with the indicated interpolation; model inference settings remain fixed. Tighter and looser settings scale F/P tolerance sets, the effective-change threshold (6), and the global-fit reference-separation cap (8) by $2/3$ and 1.5; magnitude's relative-error band becomes 0.10 and 0.20 (default 0.15).

The parameter sweep scales each of 11 parameters by 0.5, 0.75, 1.5, or 2 while fixing all others. Table 6 identifies the defaults and settings. For localization's background cap, these multipliers correspond to two, three, six, and eight times the allowed area, around the default four.

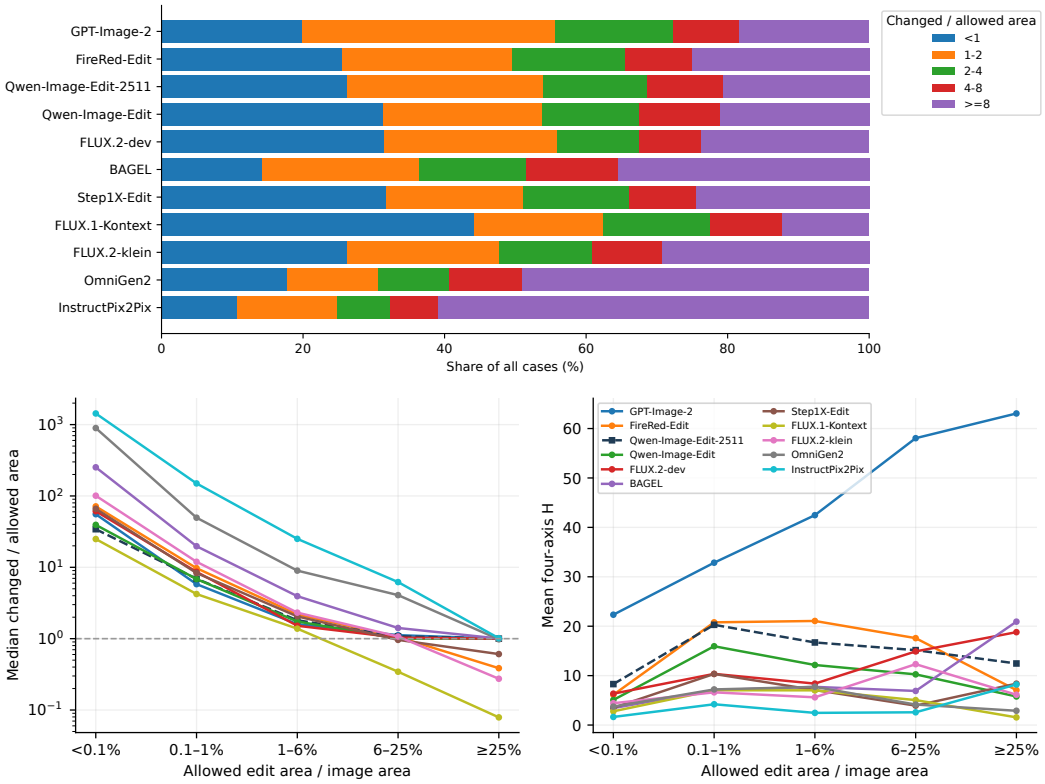


Figure 14: Detected-change area relative to the allowed edit region. Top: each model's distribution across ratio bins. Bottom: median area ratio (left) and mean composite H (right) for each model, grouped by allowed-region size. All 19,140 outputs enter the analysis, including zero-score outputs.

Results and analysis. Table 4 quantifies how much each model benefits when strong axes can compensate for weak ones. Similar rankings can accompany different absolute scores (Table 5), so scores from different configurations are not interchangeable. Figure 15 shows the effects of fidelity tolerances on the five leading models and of all parameters on the eleven-model ranking. The background-cap settings preserve the full model ranking. The cap limits how strongly widespread redraw dominates localization, while preservation evaluates the entire protected region; the comparison supports ranking stability over the tested range.

Table 4: Aggregation of identical four-axis measurements on all 19,140 outputs. Each column applies the same 22-cell macro average; bold marks the best model within that column. The columns represent different scoring choices, not model improvements.

Model	Harmonic	Geometric	Arithmetic
GPT-Image-2	**47.13**	**52.26**	**66.55**
FireRed-Edit	21.85	26.65	54.60
Qwen-Image-Edit-2511	20.47	25.85	54.71
Qwen-Image-Edit	16.26	20.11	50.53
FLUX.2-dev	13.11	17.11	49.47
BAGEL	10.81	13.73	46.08
Step1X-Edit	9.78	13.10	48.01
FLUX.2-klein	8.42	11.60	46.07
FLUX.1-Kontext	8.78	11.52	44.07
OmniGen2	7.32	9.37	39.57
InstructPix2Pix	3.83	5.57	36.62

Table 5: Macro composites under six scoring configurations on identical outputs. Bold marks the best model in each column. Color distance, prediction resolution, and tolerance choices can change absolute scores and close model orderings.

Model	Default	CIEDE2000	Half: bicubic	Half: nearest	Tighter	Looser
GPT-Image-2	**47.13**	**50.92**	**45.03**	**45.64**	**40.81**	**51.31**
FireRed-Edit	21.85	26.15	20.52	20.82	19.45	25.20
Qwen-Image-Edit-2511	20.47	25.68	18.97	19.44	17.56	24.92
Qwen-Image-Edit	16.26	19.72	15.14	15.43	14.34	18.63
FLUX.2-dev	13.11	18.36	12.19	12.47	10.22	16.44
BAGEL	10.81	13.79	10.29	10.40	9.07	13.26
Step1X-Edit	9.78	11.04	9.15	9.34	8.77	11.39
FLUX.2-klein	8.42	12.60	7.98	8.15	6.75	11.59
FLUX.1-Kontext	8.78	9.49	8.13	8.32	8.28	9.35
OmniGen2	7.32	8.58	7.13	7.18	6.59	8.19
InstructPix2Pix	3.83	4.60	3.29	3.41	2.84	5.37

Table 6: Single-parameter sweep. Each default is multiplied by 0.5, 0.75, 1.5, and 2, in that order. Setting numbers identify the points in Figure 15. All other constants remain fixed.

Parameter	Default	Settings
Fidelity tolerances	$\{5, 10, 20\}$	1–4
Preservation tolerances	$\{2, 5, 10, 20\}$	5–8
Effective-change threshold	6	9–12
Global-fit reference-separation cap	8	13–16
Minimum relative change / fit residual	0.05	17–20
Minimum local coherence	0.25	21–24
Amplitude range for coherence check	0.9	25–28
Background-component area fraction	0.0005	29–32
Localization background cap	4	33–36
Blur-match residual cap	2	37–40
Magnitude amplitude agreement band	0.15	41–44

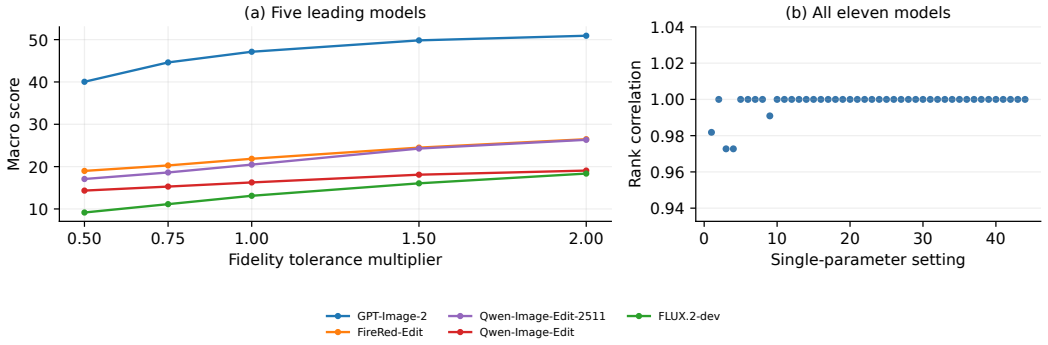


Figure 15: Scoring sensitivity on identical outputs. Left: composite scores of the five models leading under the default configuration as fidelity tolerances are scaled; a multiplier of one is the default. Right: eleven-model rank correlation for each of the 44 single-parameter settings. All other parameters and predictions stay fixed.

A.6 Controlled Change Amplitudes

Setup. We vary pixel-change amplitude on the fixed 101-case panel described in Appendix A.11. Each case contributes six constructed variants and a reference output, with color measured on the same 50 eligible cases throughout. The multiplier changes pixel amplitude, not semantic task completion.

Results and analysis. Figure 16 and Table 7 show that fidelity and magnitude rise as the output approaches the reference; overshooting lowers them while localization remains perfect. Preservation

remains nearly perfect because changes outside the allowed region are negligible. The axes retain credit for correct spatial support while exposing appearance and amplitude errors.

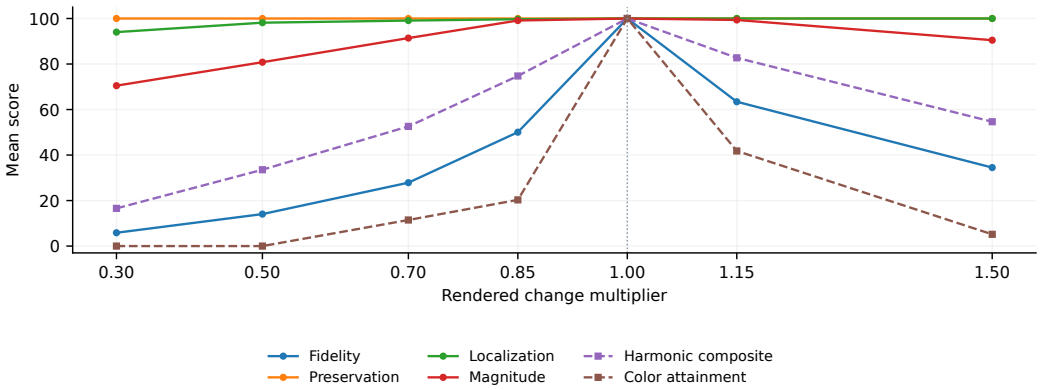


Figure 16: Response to controlled rendered changes on 101 fixed cases. Outputs are $\mathrm{clip}(I + \alpha(T - I), 0, 255)$, using six change multipliers and the reference target at $\alpha = 1$. Curves average cases equally; color attainment covers the same 50 eligible cases at every setting. α controls pixel-change amplitude, not the fraction of a semantic task completed.

Table 7: Complete means for controlled change amplitudes. F, P, L, M and H each cover 101 cases; color attainment C covers the same 50 cases for every multiplier. No best-model marking applies to constructed inputs.

α	F	P	L	M	H	C
0.3	5.85	100.00	94.02	70.49	16.57	0.00
0.5	14.05	100.00	98.16	80.80	33.55	0.00
0.7	27.86	100.00	99.09	91.40	52.59	11.46
0.85	50.05	100.00	99.69	99.11	74.74	20.30
1	100.00	100.00	100.00	100.00	100.00	100.00
1.15	63.42	100.00	100.00	99.36	82.73	41.80
1.5	34.50	100.00	100.00	90.47	54.65	5.17

Local errors and small perturbations.

Setup. A fixed panel takes five cases per asset–operation cell, prioritizing source diversity. Each condition in Table 8 uses all 110 cases. We vary the fraction of target-region pixels edited, the pixel-change amplitude, and background color, and apply blur, JPEG compression, shifts, 1% centered rescaling, and deterministic reconstruction through the Stable Diffusion v1.4 variational autoencoder (VAE). Shifting the entire target moves protected content; shifting only the target-minus-input difference isolates displacement of the requested change. Rescaling uses bilinear interpolation and center cropping or edge padding; the VAE uses the latent mode in float32.

Results and analysis. Replacing half the allowed pixels with target values preserves the background and earns partial fidelity and localization. A background color offset lowers preservation and localization even when the target region is correct; fidelity can also fall near the boundary because its comparison is smoothed. Editing half the region and applying half the change amplitude produce distinct profiles, separating spatial coverage from change strength.

Mild blur leaves the scores near the reference, whereas compression, displacement, and rescaling lower appearance and spatial agreement (Table 8). VAE reconstruction preserves magnitude more closely than target appearance: fidelity has a median of 91.57, with 8 of 110 cases below 40. These responses distinguish agreement with the reference appearance from the strength of the visual change, and quantify their sensitivity to image transformations. The scoring stage itself applies no resampling: editor outputs are restored to the reference coordinates by the shared wrapper before scoring (Appendix B.4), and the constructed variants inherit the canvas dimensions. These rows therefore quantify transformations that arise editor-side in deployed editors, not artifacts of the evaluation pipeline.

Table 8: Controlled changes on 110 fixed development cases (five per asset–operation cell). Means retain all 110 cases per row. Shifted differences modify the source by a translated target-minus-source field; whole-image shifts also move protected content.

Constructed output	F	P	L	M
Half of target-region pixels	50.17	100.00	66.65	75.03
Correct target; background RGB +20	93.51	75.89	54.86	100.00
Half change amplitude	19.89	100.00	93.72	82.24
1.5 times change amplitude	49.07	99.84	100.00	93.51
Target, JPEG quality 95	90.10	98.89	87.21	99.73
Target, Gaussian $\sigma = 0.4$	99.99	99.61	98.89	99.99
Whole target shifted 0.25 px	94.98	99.38	96.99	99.92
Edit difference shifted 0.25 px	92.25	99.92	99.64	99.91
Whole target shifted 0.5 px	88.44	99.12	95.65	99.68
Edit difference shifted 0.5 px	83.61	99.90	99.06	99.54
Whole target shifted 1 px	80.54	98.91	91.18	98.75
Edit difference shifted 1 px	73.23	99.87	94.72	97.70
Target, JPEG quality 85	87.93	98.12	84.45	99.64
Whole target shifted 2 px	73.42	98.31	84.52	97.28
Edit difference shifted 2 px	63.12	99.74	89.27	92.97
Target rescaled by $1.01\times$	63.06	94.91	64.10	92.92
Target rescaled by $0.99\times$	62.45	96.78	64.70	92.91
Target, SD v1.4 VAE encode–decode	82.71	93.28	81.12	99.51

Known source-level errors.

Setup. To test explicit editing mistakes beyond pixel perturbations, we render seven constructions on eight fixed source assets (two per type): the exact target, unchanged input, an omitted color group, a different group, an extra group, swapped target colors, and background damage. Assets are fixed by source ID before scoring; candidate color declarations must change visible pixels. Each target is rendered twice to verify repeatability.

Results and analysis. Table 9 shows complementary responses: omission reduces target agreement and coverage, whereas extra edits damage preservation. Swapping colors largely preserves localization but reduces fidelity, so magnitude alone does not establish correct assignment.

Table 9: Source-rendered controls on eight fixed assets (two per type). Values average each per-asset score equally. Groups are defined by source color declarations; they need not be semantic objects.

Construction	F	P	L	M	H
Exact target	100.00	100.00	100.00	100.00	100.00
Unchanged input	0.00	100.00	0.00	0.00	0.00
Omit one group	49.48	100.00	69.79	75.97	62.36
Edit another group	0.00	71.66	0.00	0.00	0.00
Edit an extra group	96.40	71.67	73.45	99.92	71.44
Swap target colors	2.89	100.00	99.90	68.80	6.78
Damage background	62.54	70.44	74.43	98.79	70.05

A.7 Effects of Shared Checks and the Composite

Setup. We examine all 19,140 outputs and remove the shared rejection check and the localization background cap separately, keeping all other rules fixed. Table 10 separates the first triggered condition, which determines the reported score, from independent triggers, which may overlap. Tables 13 and 14 report all axes and the primary score. The model–operation breakdown in Figure 17 shows where rejection occurs.

Table 10: Shared-check triggers on all 19,140 outputs. First-trigger counts follow the execution order and are mutually exclusive. Independent counts evaluate each condition separately and can overlap. Passing is not proof of correct target appearance.

Condition	First trigger	Independent
Empty region	0	0
Negligible change	1122	1122
Blur match	95	997
Local cancellation	114	117
Global color transformation	61	384
Passes all checks	17748	—

Results and analysis. Table 11 reports each model's rejection rate, zero-composite rate, and the change in the primary score when the check is removed. The change never exceeds 0.05 points, so the check does not drive the ranking; rejection rates stay below 10% for all models except FLUX.1-Kontext (21%).

Table 11: Effect of the shared check. Rejected and $H = 0$ are percentages of 1,740 outputs per model; ΔH is the change in the 22-cell mean when the check is removed.

Model	Rejected	H=0	ΔH
GPT-Image-2	1.3	21.0	0.00
FireRed-Edit	5.5	52.5	0.00
Qwen-Image-Edit-2511	6.7	54.7	0.02
Qwen-Image-Edit	8.0	65.9	0.01
FLUX.2-dev	7.1	66.3	0.01
BAGEL	6.4	73.6	0.00
Step1X-Edit	7.6	74.6	0.01
FLUX.2-klein	6.9	75.7	0.05
FLUX.1-Kontext	21.0	77.5	0.01
OmniGen2	6.3	78.1	0.01
InstructPix2Pix	3.2	84.9	0.02

Sources of zero composites. Table 12 decomposes the $H = 0$ outputs by the axis that vanishes. Outputs rejected by the shared check ($F = L = M = 0$) number 1,392, matching the first-trigger rejections in Table 10. Of the 12,611 zero composites, 10,998 (87%) pass the check and produce a measurable change ($L, M > 0$) yet do not raise agreement with the target above the unchanged-input baseline; zero preservation accounts for 190 and the remaining combinations for 31. Zero composites therefore arise predominantly from fidelity, not from the check discarding the output.

Table 12: Composition of zero composites among each model's 1,740 outputs. Categories are mutually exclusive and assigned in order: rejected by the shared check ($F = L = M = 0$); zero preservation ($P = 0$); zero fidelity only ($F = 0$ with $L, M > 0$), meaning agreement with the target does not exceed the unchanged-input baseline; and other combinations. The rejected total equals the first-trigger rejections in Table 10.

Model	$H = 0$	Check rejected	Zero P	Zero F only	Other
GPT-Image-2	365	23	20	322	0
FireRed-Edit	914	96	13	800	5
Qwen-Image-Edit-2511	951	116	15	816	4
Qwen-Image-Edit	1146	139	21	982	4
FLUX.2-dev	1154	124	9	1016	5
BAGEL	1281	111	29	1141	0
Step1X-Edit	1298	132	4	1156	6
FLUX.2-klein	1317	120	7	1189	1
FLUX.1-Kontext	1348	366	0	977	5
OmniGen2	1359	109	8	1241	1
InstructPix2Pix	1478	56	64	1358	0
All models	12,611	1,392	190	10,998	31

Removing the cap lowers scores while preserving the full model ranking. Paired source-group intervals in Table 15 separate uncertainty in model gaps from changes induced by this scoring choice.

Table 13: Without the shared check. Identical 19,140 outputs; axes are case means and H is the 22-cell macro mean. ΔH is relative to the fixed default, with all other rules unchanged.

Model	F	P	L	M	H	ΔH
GPT-Image-2	37.10	84.52	56.97	84.60	47.14	+0.002
FireRed-Edit	13.42	85.09	44.57	70.78	21.86	+0.005
Qwen-Image-Edit-2511	12.23	87.16	47.21	69.29	20.49	+0.015
Qwen-Image-Edit	9.39	82.06	44.78	63.23	16.27	+0.006
FLUX.2-dev	8.17	82.61	44.26	64.36	13.12	+0.012
BAGEL	6.66	68.75	43.21	63.84	10.81	+0.003
Step1X-Edit	5.24	82.78	41.07	62.61	9.79	+0.007
FLUX.2-klein	4.86	78.81	39.76	63.03	8.47	+0.046
FLUX.1-Kontext	3.78	91.64	31.66	55.23	8.79	+0.013
OmniGen2	5.17	60.52	36.07	59.18	7.33	+0.013
InstructPix2Pix	2.37	39.19	42.74	65.50	3.84	+0.017

Table 14: Without the localization background cap. Identical 19,140 outputs; axes are case means and H is the 22-cell macro mean. ΔH is relative to the fixed default, with all other rules unchanged.

Model	F	P	L	M	H	ΔH
GPT-Image-2	37.10	84.52	53.17	84.09	44.75	-2.389
FireRed-Edit	13.42	85.09	39.80	68.76	20.03	-1.824
Qwen-Image-Edit-2511	12.22	87.16	43.23	66.92	19.04	-1.434
Qwen-Image-Edit	9.38	82.06	39.58	60.53	14.40	-1.864
FLUX.2-dev	8.16	82.61	38.38	61.65	12.08	-1.029
BAGEL	6.66	68.75	34.34	61.38	9.48	-1.327
Step1X-Edit	5.24	82.78	35.07	59.47	8.95	-0.825
FLUX.2-klein	4.85	78.81	32.91	60.34	7.60	-0.821
FLUX.1-Kontext	3.78	91.64	26.17	46.69	8.11	-0.670
OmniGen2	5.16	60.52	24.99	56.80	4.95	-2.363
InstructPix2Pix	2.36	39.19	26.77	64.16	2.93	-0.900

Table 15: Pairwise gaps without the localization cap. Pairs follow the default ordering; intervals use 10,000 paired source-group resamples. The final column measures the change in each gap relative to the default.

Model pair	ΔH	95% interval	Gap change interval
GPT-Image-2 / FireRed-Edit	+24.72	[22.65, 26.66]	[-0.91, -0.21]
FireRed-Edit / Qwen-Image-Edit-2511	+0.99	[-0.21, 2.20]	[-0.69, -0.06]
Qwen-Image-Edit-2511 / Qwen-Image-Edit	+4.64	[3.41, 5.98]	[-0.03, 0.92]
Qwen-Image-Edit / FLUX.2-dev	+2.32	[0.85, 3.83]	[-1.32, -0.32]
FLUX.2-dev / BAGEL	+2.60	[1.03, 4.23]	[-0.26, 0.81]
BAGEL / Step1X-Edit	+0.53	[-0.97, 1.86]	[-1.03, 0.02]
Step1X-Edit / FLUX.1-Kontext	+0.85	[-0.35, 2.14]	[-0.52, 0.23]
FLUX.1-Kontext / FLUX.2-klein	+0.50	[-0.88, 1.84]	[-0.23, 0.55]
FLUX.2-klein / OmniGen2	+2.65	[1.15, 4.21]	[0.95, 2.17]
OmniGen2 / InstructPix2Pix	+2.03	[0.90, 3.20]	[-2.21, -0.79]

A.8 Generation and Instruction Controls

Setup. We evaluate FireRed-Edit and FLUX.2-dev on the same 94 cases with four seeds, holding checkpoints and sampling settings fixed. For instruction wording, we append displacement as a percentage of canvas size on 169 translation cases for FLUX.2-klein and FLUX.1-Kontext.

Results and analysis. The seed score ranges remain separated (Table 16), showing that the observed lead exceeds seed variation on this panel. Adding the percentage description changes case-mean H by less than half a point for both editors (Table 17).

Table 16: All four seed runs on the same 94 cases. Scores average the covered operation cells equally; bold marks the larger score within each seed.

Model	Seed 0	Seed 1	Seed 2	Seed 3
FireRed-Edit	**23.60**	**24.12**	**22.18**	**23.85**
FLUX.2-dev	17.39	19.79	17.11	16.50

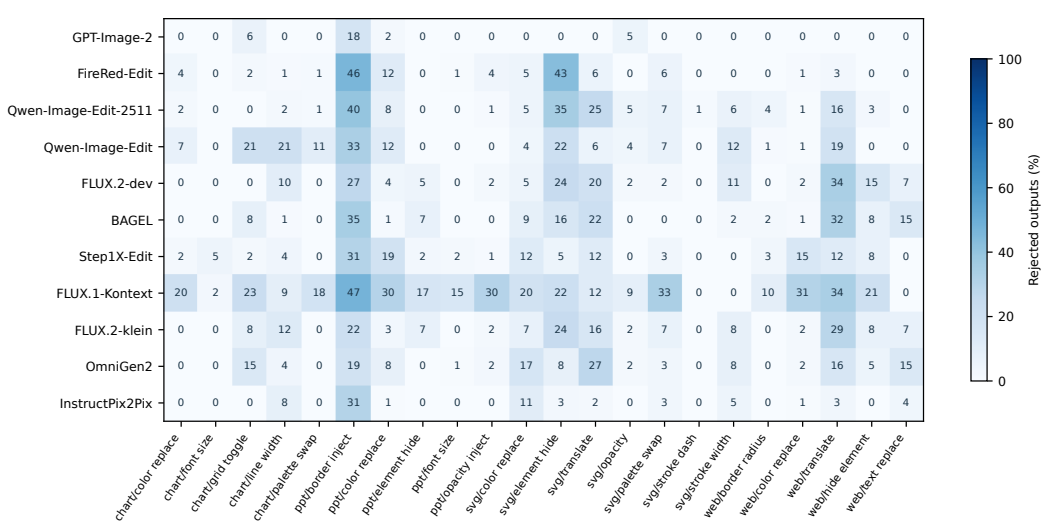


Figure 17: Shared-check rejection rates for every model–operation combination. Each cell uses all outputs in that combination as its denominator. Values are percentages, rounded for display. Rejection leaves preservation independently measured.

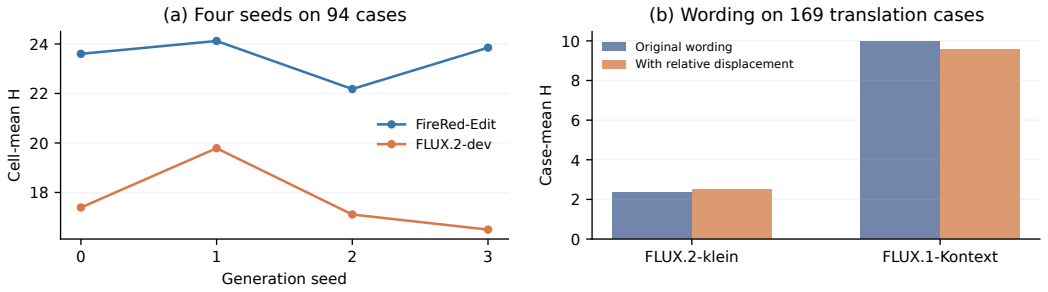


Figure 18: Generation controls on their actual subsets: four seeds on 94 cases for FireRed-Edit and FLUX.2-dev, and two instruction forms on 169 translation cases for FLUX.2-klein and FLUX.1-Kontext. Seed scores use cell-macro aggregation; wording scores use case-weighted means.

Table 17: Matched wording control on 169 cases per model. The same source image and requested edit are retained; the alternative adds displacement as a percentage of the canvas size. Scores are case-mean H; bold marks the larger model score for each instruction form.

Model	Pixel instruction	With percentage	Change
FLUX.2-klein	2.37	2.48	+0.11
FLUX.1-Kontext	**9.97**	**9.55**	-0.42

A.9 Same-operation Comparison Across Assets

Setup. The SVG/chart/web/slide recoloring subsets contain 82/54/183/136 cases from 35/14/40/29 sources. Their median allowed areas are 3.92/7.74/3.00/1.78% of the image.

Within each asset type, we draw 10,000 bootstrap resamples of source assets, keeping cases from the same source together and sharing the draws across models. We report pointwise 95% percentile intervals for case-mean H and paired model differences. The coarse area strata receive equal weight; resamples missing a required area stratum are excluded from the area-standardized intervals.

Table 19 describes the four fixed finer bins ($\leq 1\%$, 1–5%, 5–10%, and $> 25\%$), retaining 402 cases. Their sparse-bin means are descriptive and receive no confidence intervals.

Results and analysis. Table 18 supplies the four-axis breakdown. The FLUX.2-dev–FireRed-Edit ranking reversal between charts and SVG persists after area standardization (Table 20). Area weighting adjusts region-size mixture without equating source content, colors, or layout.

Table 18: Recoloring on the same operation across asset types. Entries give mean H [95% source-cluster bootstrap interval] and the four mean axes. All 455 cases and eleven models are retained. Bold marks the highest mean per asset and measurement.

Asset	Model	H [95% CI]	F	P	L	M
SVG (82)	GPT-Image-2	**76.78** [70.18, 82.43]	**64.99**	87.73	**85.47**	**95.78**
	FireRed-Edit	15.53 [10.55, 20.93]	8.78	88.67	69.30	74.07
	Qwen-Image-Edit-2511	15.98 [10.21, 21.82]	9.23	**89.81**	66.88	80.64
	FLUX.2-dev	8.43 [4.53, 12.61]	5.48	70.88	59.28	76.37
	Qwen-Image-Edit	11.99 [7.27, 17.27]	7.74	76.24	55.87	71.59
	BAGEL	6.35 [3.06, 9.84]	3.39	70.65	56.15	69.89
	FLUX.2-klein	11.44 [7.13, 16.25]	6.81	62.27	48.42	72.96
	Step1X-Edit	3.09 [0.82, 5.81]	2.05	67.89	44.58	53.21
	FLUX.1-Kontext	1.67 [0.26, 3.76]	0.74	81.16	32.22	49.09
	OmniGen2	2.63 [0.15, 5.48]	2.15	37.30	38.16	51.09
	InstructPix2Pix	0.82 [0.05, 2.34]	0.84	25.86	45.40	57.39
Chart (54)	GPT-Image-2	**68.68** [53.97, 81.52]	**53.29**	95.68	**80.40**	**96.69**
	FireRed-Edit	5.05 [2.16, 8.42]	1.69	**96.75**	67.74	75.72
	Qwen-Image-Edit-2511	6.48 [3.17, 9.86]	2.29	96.74	70.10	77.06
	FLUX.2-dev	17.39 [11.68, 23.41]	7.05	96.47	79.71	84.98
	Qwen-Image-Edit	3.22 [0.97, 5.98]	1.16	93.75	57.04	58.30
	BAGEL	0.99 [0.10, 2.30]	0.31	89.38	74.13	73.63
	FLUX.2-klein	7.27 [3.63, 11.31]	2.49	96.60	76.25	78.40
	Step1X-Edit	0.87 [0.00, 2.34]	0.31	95.81	60.15	62.04
	FLUX.1-Kontext	1.09 [0.00, 2.77]	0.34	96.10	34.36	43.77
	OmniGen2	1.39 [0.45, 2.45]	0.44	58.50	45.96	63.18
	InstructPix2Pix	1.69 [0.27, 3.44]	0.48	73.47	67.89	68.46
Web (183)	GPT-Image-2	**32.83** [25.13, 41.16]	**30.03**	69.62	**55.47**	**89.08**
	FireRed-Edit	11.80 [8.14, 15.72]	6.56	78.81	50.12	79.46
	Qwen-Image-Edit-2511	13.58 [8.54, 19.38]	9.28	76.98	51.80	75.06
	FLUX.2-dev	11.14 [7.78, 14.79]	7.69	61.97	46.87	76.57
	Qwen-Image-Edit	6.90 [4.43, 9.41]	3.98	64.32	47.88	72.59
	BAGEL	1.39 [0.36, 2.73]	1.19	32.33	36.13	60.58
	FLUX.2-klein	9.98 [6.49, 13.96]	5.58	72.67	42.23	74.10
	Step1X-Edit	1.68 [0.47, 3.22]	0.77	87.94	31.70	50.37
	FLUX.1-Kontext	0.96 [0.41, 1.59]	0.57	**94.15**	21.92	38.00
	OmniGen2	0.36 [0.09, 0.69]	0.17	55.57	34.86	50.91
	InstructPix2Pix	2.40 [0.63, 5.14]	1.31	34.23	38.44	63.93
Slide (136)	GPT-Image-2	**54.03** [45.43, 62.30]	**45.22**	85.91	**59.28**	**92.20**
	FireRed-Edit	10.68 [7.16, 14.39]	5.72	92.59	54.98	69.13
	Qwen-Image-Edit-2511	14.60 [9.20, 20.50]	9.19	90.04	54.75	73.17
	FLUX.2-dev	11.17 [8.08, 14.44]	6.43	67.38	52.27	78.08
	Qwen-Image-Edit	3.09 [1.45, 4.96]	1.53	88.97	50.93	57.94
	BAGEL	1.99 [0.30, 4.15]	1.08	48.22	44.20	60.09
	FLUX.2-klein	6.69 [3.74, 10.00]	4.67	64.75	45.97	70.76
	Step1X-Edit	2.58 [1.02, 4.41]	1.41	90.22	37.30	49.18
	FLUX.1-Kontext	0.77 [0.12, 1.72]	0.54	**93.06**	32.62	40.03
	OmniGen2	1.80 [0.88, 2.86]	1.21	51.58	31.38	58.82
	InstructPix2Pix	0.48 [0.01, 1.24]	0.54	16.39	41.92	67.27

Table 19: Cases / source assets in the recoloring area strata. Coarse strata retain all 455 cases; the four shared fine strata retain 402. An asset can contribute cases to more than one stratum.

	Coarse area strata		Fine shared area strata			
Asset	$\leq 10\%$	$> 10\%$	$\leq 1\%$	1–5%	5–10%	$> 25\%$
SVG	51 / 28	31 / 19	19 / 14	23 / 16	9 / 8	21 / 14
Chart	30 / 8	24 / 6	8 / 2	14 / 4	8 / 2	8 / 2
Web	151 / 36	32 / 6	57 / 17	60 / 14	34 / 7	18 / 3
Slide	105 / 29	31 / 13	53 / 15	27 / 10	25 / 7	18 / 9

Table 20: Mean H after equal weighting of the area strata defined in Table 19. Best model per asset and weighting is bold. These comparisons describe area-standardized performance, not a causal effect of asset type.

	Two coarse strata (all 455 cases)				Four fine strata (402 cases)			
Model	SVG	Chart	Web	Slide	SVG	Chart	Web	Slide
GPT-Image-2	**78.19**	**70.56**	**30.48**	**62.33**	**75.29**	**64.74**	**28.13**	**56.35**
FireRed-Edit	16.63	5.12	10.30	11.99	14.97	4.52	9.07	10.69
Qwen-Image-Edit-2511	17.78	6.81	10.59	15.44	14.59	5.28	11.78	17.35
FLUX.2-dev	8.88	18.15	12.85	12.08	7.11	16.89	9.61	12.49
Qwen-Image-Edit	12.67	3.29	8.47	3.12	10.60	2.86	6.29	3.86
BAGEL	8.02	1.07	2.74	4.05	6.15	0.69	1.34	2.59
FLUX.2-klein	11.41	7.69	12.46	8.11	10.79	5.52	9.03	7.62
Step1X-Edit	3.52	0.97	2.27	3.33	2.24	1.47	2.23	3.03
FLUX.1-Kontext	1.92	0.98	0.92	0.58	0.52	1.32	1.19	0.65
OmniGen2	3.03	1.50	0.36	1.51	3.67	1.56	0.30	1.33
InstructPix2Pix	0.66	1.54	1.60	0.81	0.82	2.12	2.00	0.79

A.10 Color Attainment and Component Diagnosis on the Same Outputs

Setup. We extend the conditional analysis to all models and three color cutoffs on the same 621 cases (Table 21). Counts for different axes overlap; rates are undefined when no output meets a cutoff. We resample numerators and denominators together over 10,000 source-group draws from the 127 covered assets.

Results and analysis. For GPT-Image-2 at the main color cutoff, the 95% percentage intervals are [7.2, 18.4] for low preservation and [11.7, 24.6] for low localization.

Table 21: Conditional diagnosis on the same 621 color cases per model. For each color-attainment cutoff c, n is the number of outputs with color score $\geq c$; subsequent columns count $F/P/L/M < 50$. These are descriptive slices, not new pass criteria.

	Color ≥ 32					Color ≥ 50					Color ≥ 80				
Model	n	F	P	L	M	n	F	P	L	M	n	F	P	L	M
GPT-Image-2	449	134	60	95	5	401	97	50	71	5	270	43	27	44	0
FireRed-Edit	20	14	5	9	0	13	9	4	5	0	7	4	1	2	0
Qwen-Image-Edit-2511	38	27	7	15	2	30	21	6	11	2	18	12	3	5	0
FLUX.2-dev	52	32	14	21	2	37	22	12	16	1	15	8	2	6	0
Qwen-Image-Edit	16	14	4	6	0	8	7	2	3	0	4	4	0	2	0
BAGEL	5	4	1	1	0	4	3	1	1	0	3	3	1	1	0
FLUX.2-klein	21	11	2	7	0	17	8	2	7	0	9	5	2	5	0
Step1X-Edit	4	2	1	1	1	4	2	1	1	1	4	2	1	1	1
FLUX.1-Kontext	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OmniGen2	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
InstructPix2Pix	4	4	3	4	1	1	1	1	1	0	0	0	0	0	0

Table 22 shows that open-weight models alter the requested regions in most outputs but rarely reach the requested color.

Table 22: Color attainment and edit execution on the same 621 color-editing cases. Columns give the case-mean color attainment (Table 2 instead averages asset–operation cells), the percentage of outputs with zero attainment and with attainment of at least 50, the case-mean localization and magnitude, and the percentage of outputs with both $L > 0$ and $M > 0$.

Model	Attainment	Zero (%)	≥ 50 (%)	L	M	$L, M > 0$ (%)
GPT-Image-2	61.47	8.2	64.6	69.94	93.46	99.5
FireRed-Edit	3.04	74.7	2.1	58.96	73.81	95.2
Qwen-Image-Edit-2511	5.48	68.8	4.8	58.50	75.55	96.1
Qwen-Image-Edit	2.32	74.7	1.3	51.54	65.67	93.9
FLUX.2-dev	6.44	63.4	6.0	60.13	79.47	97.7
BAGEL	0.93	82.8	0.6	52.19	67.30	98.2
Step1X-Edit	0.91	84.1	0.6	45.68	54.63	88.4
FLUX.2-klein	3.31	71.3	2.7	54.88	75.29	96.8
FLUX.1-Kontext	0.22	84.5	0.0	29.48	41.48	72.6
OmniGen2	0.50	82.6	0.2	40.74	55.80	94.8
InstructPix2Pix	0.60	81.0	0.2	48.41	65.40	97.7

A.11 Official PaintBench Measurements on Identical Outputs

Setup. The fixed diagnostic panel contains 12 SVG, 12 chart, 41 web, and 36 slide cases from 40 sources. The panel was assembled to cover known scoring boundaries, namely edits that resemble global image perturbations and edits near the change-detection threshold, together with 24 stratified random controls. It is not a random sample of the benchmark, so pair frequencies do not estimate prevalence.

We score its 1,111 model outputs using PaintBench's `compute_problem_stats` at commit `a648b9f`. For CIE76 tolerance $t \in \{0, \ldots, 10\}$, correct edit pixels CE_t and incorrect preservation pixels IP_t give

$$\mathrm{IoU}_t = \frac{CE_t}{|E| + IP_t}, \qquad E = \{x : I(x) \neq T(x)\}, \qquad \mathrm{mIoU} = \frac{1}{11}\sum_{t=0}^{10} \mathrm{IoU}_t\,.$$

We retain PaintBench's own edit region, both accuracy curves, and changed-pixel counts. Its preservation accuracy on this panel is a separate measure from our P, which uses the protected region and tolerance set defined in Section 3.3. Both scorers use our image alignment, without PaintBench's separate resize/center-crop preprocessing. Reference targets attain mIoU 100; unchanged inputs have zero edit accuracy at tolerance zero and preservation accuracy 100.

We compare 27 matching windows based on mIoU and mean edit/preservation accuracy (Table 25). For the 101-case panel's 5,555 same-case model pairs, the strict comparison requires an mIoU difference of at most one point and edit- and preservation-accuracy differences of at most two points at every tolerance, with all scores on a 0–100 scale.

Results and analysis. Table 23 gives per-model means. The main figure illustrates a pair satisfying this strict rule. The differing matched pairs span 13 cases. In the illustrated webpage, FLUX.2-dev and Step1X-Edit both miss the requested target appearance. Their spatial coverage and change magnitude nevertheless differ (Table 23 gives panel-wide means).

Figure 20 retains both complete accuracy curves for the pair shown in Figure 5.

Table 23: Same aligned outputs, official PaintBench measurements and our four-axis measurements. Means over 101 fixed diagnostic cases per model (0–100), covering 11 models. These are measurements of this panel, not scores on PaintBench's original benchmark. Best values are bold within each column.

Model	mIoU	Edit acc.	Pres. acc.	F	P	L	M
GPT-Image-2	**8.20**	**26.67**	77.96	**36.89**	90.36	**50.99**	**84.84**
FireRed-Edit	2.80	16.11	82.19	15.93	88.66	50.97	72.32
Qwen-Image-Edit-2511	2.48	17.11	81.83	15.36	89.01	47.74	70.61
Qwen-Image-Edit	2.18	15.38	78.57	13.88	87.18	45.39	64.98
FLUX.2-dev	1.36	9.11	76.45	9.57	86.10	45.30	69.70
FLUX.2-klein	1.18	7.83	70.68	5.43	81.77	40.76	67.89
BAGEL	1.39	10.26	62.33	7.93	72.01	36.24	58.88
Step1X-Edit	1.19	10.26	78.03	6.90	86.87	37.79	63.58
OmniGen2	0.40	8.61	46.65	6.85	54.83	30.44	62.04
FLUX.1-Kontext	0.78	7.06	**85.84**	2.57	**93.22**	22.69	39.94
InstructPix2Pix	0.19	3.41	19.71	2.02	27.74	35.11	64.13

Comparison across all source assets.

Setup. To examine these distinctions across sources, we use a stratified random panel: one case per source–operation group (seed 20260917), giving 607 cases from all 153 sources and 6,677 existing outputs. Both panels use the same scoring rules and similarity windows.

Results and analysis. Among 33,385 same-case model pairs, 416 satisfy the strict matching rule; 223 of these differ by at least 20 points in L or M, spanning 68 cases and 56 sources. Table 24 reports the complete difference distributions, including unmatched pairs. This comparison shows that the diagnostic distinction extends beyond the boundary-focused panel. Frequencies describe this source–operation sample, not the prevalence among all 1,740 cases; models sharing a source are not independent observations.

Table 24: Complete difference distributions on the fixed 607-case panel: 33,385 same-case pairs, including 416 matched pairs. Maximum curve gap considers both accuracy curves at all eleven tolerances. All differences are on a 0–100 scale.

Pairs	Difference	Min.	Q1	Median	Q3	Max.
All pairs	$\lvert\Delta\mathrm{mIoU}\rvert$	0.00	0.08	1.00	6.30	96.00
All pairs	Maximum curve gap	0.00	38.70	69.29	92.27	100.00
All pairs	$\max(\lvert\Delta L\rvert, \lvert\Delta M\rvert)$	0.00	12.25	27.49	47.32	100.00
Matched pairs	$\lvert\Delta\mathrm{mIoU}\rvert$	0.00	0.00	0.00	0.02	0.85
Matched pairs	Maximum curve gap	0.00	0.02	0.40	1.10	1.99
Matched pairs	$\max(\lvert\Delta L\rvert, \lvert\Delta M\rvert)$	0.00	7.06	23.79	44.91	100.00

Table 25: Sensitivity to the numerical similarity window. d_I bounds the mIoU gap; d_A bounds each mean edit/preservation-accuracy gap; d_{LM} is the minimum localization or magnitude gap. Counts concern 5,555 same-case model pairs on the 101-case panel. The stricter rule used in the main text bounds each accuracy gap at every tolerance by two points, with $d_I = 1$: 78 pairs qualify, and 28 differ by at least 20 points in localization or magnitude, spanning 13 cases.

d_I	d_A	d_{LM}	Similar pairs	Differing pairs	Cases
0.5	1	10	213	123	48
0.5	1	20	213	92	38
0.5	1	30	213	71	31
0.5	2	10	379	219	66
0.5	2	20	379	164	56
0.5	2	30	379	127	46
0.5	5	10	874	516	78
0.5	5	20	874	395	71
0.5	5	30	874	293	57
1	1	10	221	124	49
1	1	20	221	93	39
1	1	30	221	72	32
1	2	10	402	230	66
1	2	20	402	170	57
1	2	30	402	132	47
1	5	10	945	556	79
1	5	20	945	419	73
1	5	30	945	310	60
2	1	10	222	124	49
2	1	20	222	93	39
2	1	30	222	72	32
2	2	10	408	232	66
2	2	20	408	172	57
2	2	30	408	134	47
2	5	10	990	583	82
2	5	20	990	440	75
2	5	30	990	325	62

A.12 Comparing Qwen-Image-Edit Versions

Setup. We compare Qwen-Image-Edit and Qwen-Image-Edit-2511 on identical source images and instructions for all 1,740 cases. Both use bfloat16 (BF16), true classifier-free guidance (CFG) scale 4.0, and seed 42, with 50 steps for Qwen-Image-Edit and 40 for Qwen-Image-Edit-2511 (Appendix B.4). Table 26 reports per-operation means and paired intervals using the source-group procedure in Section 4.1.

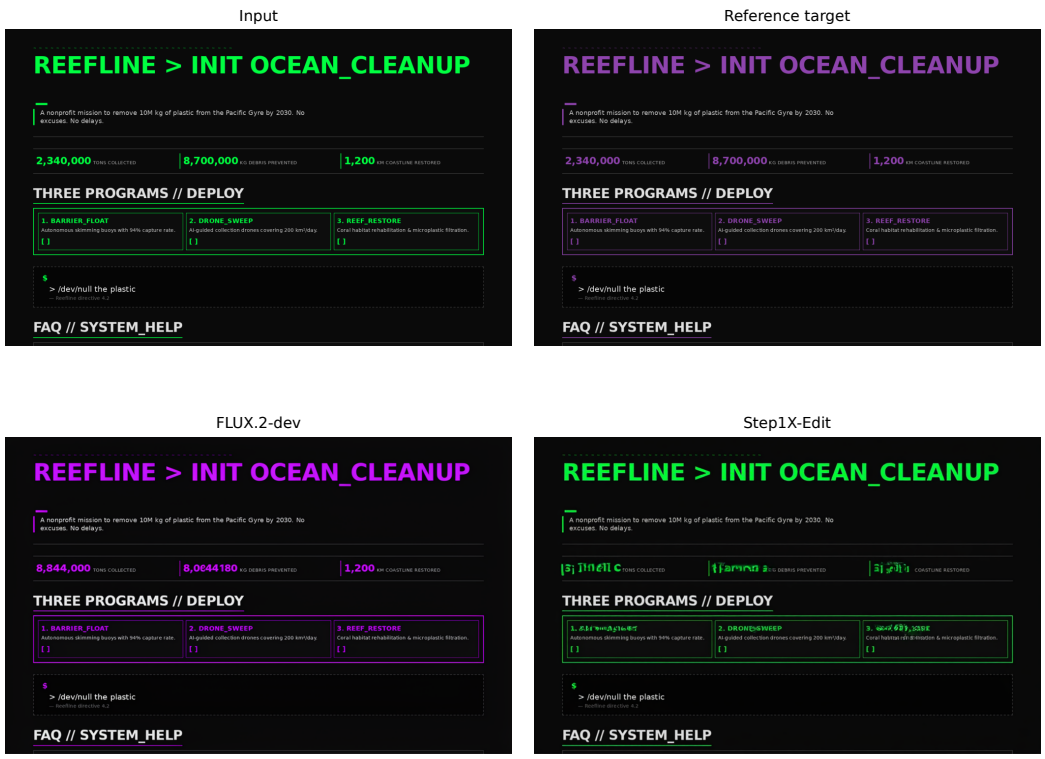


Figure 19: Complete task and instruction, with the input, reference target, and FLUX.2-dev/Step1X-Edit outputs used in the comparison.

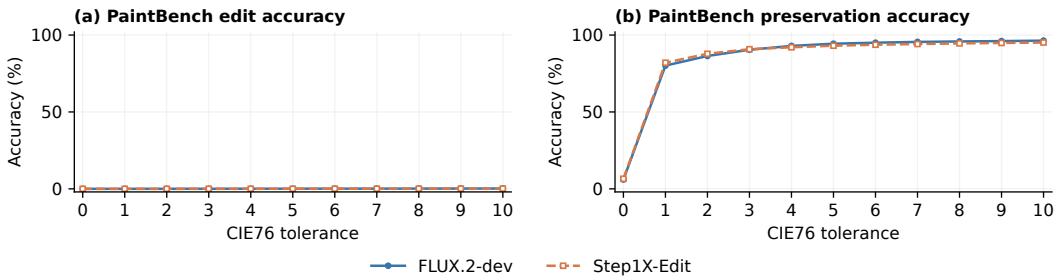


Figure 20: Complete PaintBench accuracy curves for the FLUX.2-dev/Step1X-Edit pair in Figure 5. Each curve retains all eleven CIE76 tolerances. The maximum between-model accuracy gaps are 0.07 points for editing and 1.87 points for preservation, both within the two-point matching bound at every tolerance.

Table 26: Qwen-Image-Edit version comparison by operation. The two H columns use all cases in each cell. All 1,740 cases use identical actual source images and instructions; Δ is Qwen-Image-Edit-2511 minus Qwen-Image-Edit, with pointwise 95% source-group BCa intervals. Bold marks the higher full-set mean, not a significance test. The final row weights cells equally.

Asset	Operation	Original H	2511 H	n	Δ [95% CI]
Chart	Color	3.22	**6.48**	54	+3.26 [-0.34, +7.60]
	Font size	7.11	**20.32**	42	+13.21 [+2.18, +21.57]
	Grid	**9.75**	9.27	48	-0.49 [-1.44, -0.13]
	Line width	9.68	**17.50**	96	+7.81 [+3.85, +13.98]
	Palette	4.95	**8.62**	76	+3.66 [-2.47, +8.56]
Slide	Border	2.94	**4.82**	78	+1.87 [+0.08, +5.15]
	Color	3.09	**14.60**	136	+11.51 [+6.64, +17.39]
	Hide	61.26	**68.35**	42	+7.10 [+1.37, +14.96]
	Font size	**11.66**	11.50	131	-0.16 [-3.60, +3.78]
	Opacity	1.48	**7.08**	122	+5.59 [+2.76, +10.11]
SVG	Color	11.99	**15.98**	82	+3.99 [-1.19, +9.16]
	Hide	**29.73**	23.00	37	-6.72 [-18.33, +4.77]
	Move	**33.35**	25.78	51	-7.57 [-22.00, +5.71]
	Opacity	**9.32**	7.98	56	-1.35 [-8.29, +5.24]
	Palette	3.42	**12.11**	90	+8.68 [+4.02, +15.36]
	Dashes	10.75	**11.37**	75	+0.62 [-8.32, +6.50]
	Stroke width	2.52	**6.76**	64	+4.24 [+0.29, +9.26]
Web	Corners	1.54	**2.29**	93	+0.74 [-0.79, +4.01]
	Color	6.90	**13.58**	183	+6.68 [+1.57, +13.32]
	Move	15.90	**26.57**	118	+10.66 [+4.19, +18.18]
	Hide	65.72	**79.17**	39	+13.45 [+8.94, +19.26]
	Text	51.47	**57.32**	27	+5.85 [+1.83, +13.40]
Cell-mean H		16.26	**20.47**	1740	+4.21 [+2.95, +5.46]

Results and analysis. Qwen-Image-Edit-2511 improves aggregate performance, with the largest gains in web visibility, chart font size, slide recoloring, web translation, SVG palette replacement, and chart line width. Slide typography shows no clear improvement; the gain in web corner-radius control is small, and both versions still score poorly. Negative point estimates for SVG visibility, translation, and opacity have intervals containing zero. Chart grid editing shows a small decline with an interval below zero. The comparison reflects the configured editors and their step counts. *A version-level improvement is informative when accompanied by an account of which capabilities improve.*

A.13 Uncertainty with Shared Source Assets

Setup. For each of 10,000 bootstrap draws, we sample sources with replacement within each asset type and include all operations belonging to each sampled source. Models share the same draw. We then recompute the 22 cell means and weight cells equally, as in the primary score. No cell was empty in these draws. Bias-corrected and accelerated intervals use leave-one-source-out estimates with the same grouping. This preserves dependencies between operations that reuse a source, while conditioning on the tested outputs and inference settings. The same source grouping is used for the paired Qwen-Image-Edit comparison in Table 26. Same-operation comparisons retain the asset-specific source resampling described in Appendix A.9.

Results and analysis. Table 1 reports the marginal intervals and Table 27 the paired gaps between adjacent models. The paired intervals separate the six leading models from one another and OmniGen2 from InstructPix2Pix, whereas the four adjacent pairs between BAGEL and OmniGen2 remain statistically unresolved; close point estimates should not be read as a reliable ordering.

Table 27: Paired gaps in cell-mean H between adjacent models of Table 1 under the default scoring configuration. ΔH is the first model minus the second; intervals are 95% BCa intervals from 10,000 source-group bootstrap draws shared across models. An interval excluding zero separates the pair.

Model pair	ΔH	95% interval
GPT-Image-2 / FireRed-Edit	$+25.28$	$[23.29, 27.38]$
FireRed-Edit / Qwen-Image-Edit-2511	$+1.38$	$[0.02, 2.61]$
Qwen-Image-Edit-2511 / Qwen-Image-Edit	$+4.21$	$[2.95, 5.46]$
Qwen-Image-Edit / FLUX.2-dev	$+3.15$	$[1.72, 4.65]$
FLUX.2-dev / BAGEL	$+2.30$	$[0.53, 3.91]$
BAGEL / Step1X-Edit	$+1.03$	$[-0.58, 2.70]$
Step1X-Edit / FLUX.1-Kontext	$+1.00$	$[-0.40, 2.36]$
FLUX.1-Kontext / FLUX.2-klein	$+0.35$	$[-1.02, 1.88]$
FLUX.2-klein / OmniGen2	$+1.11$	$[-0.53, 2.71]$
OmniGen2 / InstructPix2Pix	$+3.49$	$[2.00, 4.93]$

B Dataset Construction and Scoring Details

This appendix provides the asset sources and implementation choices supporting the measurements in Section 3.

B.1 Source Asset Construction

For SVG assets, we adapt the template generation stage of Chat2SVG (Wu et al., 2025c), and generation components of IntroSVG (Wang et al., 2026a), with simplified variants and custom outline templates. Chart generation uses reference code from ChartMimic (Yang et al., 2025a), alongside LLM-based and programmatic synthesis inspired by ChartLlama (Han et al., 2023) and FigureQA (Kahou et al., 2018), respectively. For web assets, we adapt OpenUI[2] system prompts, draw on WebGen-Bench (Lu et al., 2025) for instruction design, and use a custom two-stage generation workflow with layout categories informed by Open Design[3]. For slides, we adapt AutoPresent (Ge et al., 2025) code generation prompts and convert the resulting slides to HTML. The source counts refer to constructed assets, not deployed websites or collected presentation files. The SVG pool has 12 sources from each of four construction variants (Chat2SVG, a simplified variant, IntroSVG, and custom outlines). Charts comprise 10 ChartMimic-inspired, 10 ChartLlama-inspired, and five FigureQA-inspired sources. Web assets comprise 25 custom two-stage, 13 OpenUI-prompt, and 13 WebGen-inspired sources.

Task verification.

We require paired renders to have matching canvas dimensions, a visible requested change, preserved surrounding layout, and agreement between the instruction and the edited objects and parameters. Repeated rendered answers for the same source and operation are deduplicated. AI-assisted screening uses Kimi K3 (Kimi Team et al., 2026), GLM-5.3-Flash (Z.ai, 2026), and GPT-6 (OpenAI, 2026b). All retained cases have an AI inspection record.An author with a computer vision research background reviews all retained tasks and resolves reported ambiguities against the editable sources. Object references are stated in visible terms, and rendered targets preserve background cutouts and other protected content.

Inputs, targets, masks, and edit metadata are retained together.

Source-level variation. We describe the 153 sources before counting their edit variants (Figure 21). For these descriptors only, source images are resized to at most 512 pixels on the longest side; color entropy uses eight bins per RGB channel, and edge density uses fixed Canny thresholds of 80 and 160. These pixel descriptors capture variation in palette and visual structure; sources differ within each asset type.

Model-specific image preparation is specified in Appendix B.4.

[2] https://github.com/wandb/openui

[3] https://github.com/nexu-io/open-design

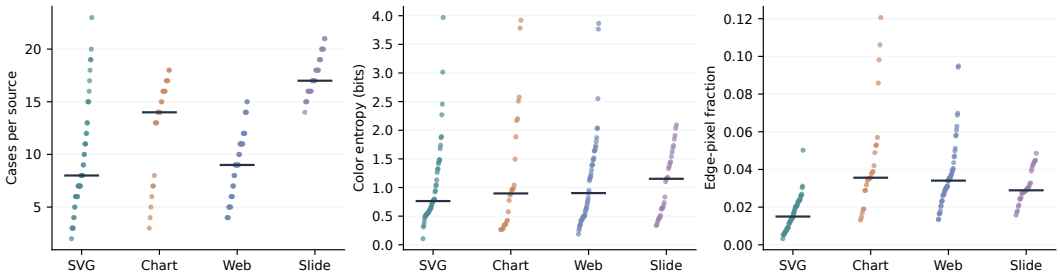


Figure 21: Source-level descriptors. Each point is one source; horizontal bars mark medians. Multiple edits of a source contribute only one point. Color entropy describes palette dispersion, and edge density describes visible detail under fixed preprocessing.

B.1.1 Human Verification of Task Validity

Setup. One author with a computer vision research background inspected the input image, instruction, rendered target, and edit mask for all 1,740 tasks retained through screening with Kimi K3, GLM-5.3-Flash, and GPT-6, covering 153 source assets and all 22 asset–operation combinations. The criteria were identifiable target objects, agreement between the requested change and target, and preservation of unrelated content. The human confirmation rate of the screening outcome was measured as the fraction of retained tasks accepted without concerns during human review.

Results and analysis. During curation, human review accepted 1,721 of 1,740 tasks without concerns (98.91% were accepted without concerns on initial human inspection) and raised questions about 19 tasks. Source and render inspection identified 11 cases requiring clearer object references and two requiring target correction; the other six were consistent with their instructions. The clarifications and corrections were checked against the rendered outcomes. These results show how human review resolves residual object-reference and rendering issues that survive multi-model screening. All confirmed issues were resolved, and the corrected tasks passed rechecking. All benchmark scores reported in this paper use the final, corrected evaluation references.

B.2 Image Preparation and Scoring Details

Image preparation. After the inference wrapper restores the reference canvas (Appendix B.4), the scorer compares RGB images at the target dimensions. Its fallback for mismatched dimensions uses bicubic interpolation; when the absolute difference in aspect ratio exceeds 0.02, centered black padding precedes resizing. No content-based registration is applied. The allowed region in Equation 1 follows rendered change, including vacated and newly occupied pixels, while unchanged pixels inside an object's bounding box remain protected.

Fidelity and preservation. For fidelity, each image is smoothed by a 3×3 Gaussian filter with $\sigma = 0.6$, and the mask is dilated by a 3×3 rectangular kernel, giving $\widetilde{I}, \widetilde{T}, \widetilde{O}$ and E^{+}. Both the prediction–target agreement and unchanged-input baseline use the same channel and tolerances $\{5, 10, 20\}$. A baseline $b > 0.995$ is flagged as uninformative and receives zero fidelity; an empty mask also receives zero fidelity. Preservation instead uses unblurred images, the original mask, and tolerances $\{2, 5, 10, 20\}$. When no pixels lie outside E, preservation is 100 because there is no protected region to evaluate.

Spatial change maps. We use a localization threshold of 6 if at least $\max(16, 0.05|E|)$ input–target pixels exceed that distance, and 2 otherwise. Thresholding input–target and input–output differences gives G and the candidate output change map; an empty G falls back to E. All candidate pixels inside E are retained. Outside E, only eight-connected components with at least $\max(16, \lfloor 0.0005N \rfloor)$ pixels survive, where N is the canvas area. This produces D without erasing coherent thin bands. Empty detected changes or zero recall/precision give $L = 0$. The precision denominator caps background change at $4|E|$ to keep widespread redraw from overwhelming spatial diagnosis; preservation still measures agreement across the full protected region. All four axes are pixel-area weighted, so an object contributes in proportion to its rendered area; background components below the threshold above are treated as rendering noise by L.

Effective-change core and amplitude. The core C first retains pixels in E whose input–target distance exceeds 6. If fewer than $\max(20, 0.2|E|)$ pixels remain, the threshold falls to 2; if the core is still too small, it becomes E. Using Lab pixel vectors, the amplitude ratio and agreement are

$$r_M = \frac{\sum_{x \in C} \|O_x - I_x\|_1}{\sum_{x \in C} \|T_x - I_x\|_1}, \qquad a(r_M) = \max\{0, 1 - \max(0, |r_M - 1| - 0.15)\}. \tag{6}$$

If the target amplitude is below 10^{-6}, the ratio is 1 when the output amplitude is also below that threshold, and 0 otherwise. The direction term compares the complete Lab change fields on C:

$$c = \frac{\sum_{x \in C} (O_x - I_x)^\top (T_x - I_x)}{\sqrt{\sum_{x \in C} \|O_x - I_x\|_2^2} \sqrt{\sum_{x \in C} \|T_x - I_x\|_2^2}}. \tag{7}$$

Keeping pixel locations avoids cancellation between opposing changes, for example the vacated and newly occupied regions of a translated object. A change field with norm below 10^{-6} is treated as zero: two zero fields receive $c = 1$, while only one zero field gives $c = 0$. The amplitude term gives full credit within 15% of the requested visual change and decreases linearly beyond that band.

Eligibility for attributable editing. Fidelity, localization, and magnitude share the same deterministic check g. It examines change strength relative to the target, cancellation of changes within local blocks, similarity to a blurred input, and whether a background-fitted global color transform also explains the apparent edit. Blur and global-transform checks require both agreement with the fitted transformation and residual change below 5% of the target amplitude. Target comparisons protect requested transform-like edits; a fitted transformation alone cannot erase a measurable local edit. A failed check sets these three axes to zero; preservation is measured separately. Passing does not establish that the requested appearance is correct, which remains the role of fidelity. The checks and thresholds are fixed across models; their exact implementation and configuration are included in the supplementary code.

Exact rejection rules. Let m_T be the median input–target CIE76 distance on C, and define the residual ratio for an explanatory image X as $\rho_X = \sum_{x \in C} \|O_x - X_x\|_1 / \max(\sum_{x \in C} \|T_x - I_x\|_1, 10^{-6})$, using Lab vectors. The first satisfied rule sets $g = 0$; otherwise $g = 1$.

1. *Negligible change:* E or C is empty, or $r_M < 0.05$.
2. *Blur match:* for a Gaussian-blurred input B_σ, $\sigma \in \{1, 2, 3\}$, require $\rho_{B_\sigma} < 0.05$ and a median B_σ–O distance on C below both $\min(2, 0.8m_T)$ and the median O–T distance. The median B_σ–T distance must be at least $\min(2, 0.8m_T)$.
3. *Local cancellation:* when $r_M < 0.9$, dilate C by eight pixels and retain Lab changes there. Within 16×16 blocks, sum the signed changes in each channel. Divide the sum of absolute block sums by the sum of absolute pixel changes, assigning zero when that denominator is at most 10^{-9}; reject if the maximum ratio across channels is below 0.25.
4. *Global color transformation:* fit an affine Lab map $AI + t$ outside E, using at most 65,536 pixels selected by deterministic stride. Require at least 4,096 background pixels, $|E| \leq 0.8N$, and a smallest singular value of centered background Lab samples, normalized by $\sqrt{n-1}$, of at least 0.3. An eligible fit has $\max_{ij} |(A - \mathrm{Id})_{ij}| \leq 1$, $\|t\|_\infty \leq 40$, and median background residual below 6. Reject only if $\rho_{AI+t} < 0.05$, its median core residual to O is below $\min(2, 0.75m_T)$, and that to T is at least $\min(8, 0.75m_T)$. If the singular-value condition fails, fit only luminance when its background variance is at least 1: require $0.8 \leq a \leq 1.25$, $|b| \leq 40$, and the same median residual thresholds using luminance differences. For this fallback, ρ still uses all three Lab channels, with the input's chromatic channels unchanged. Insufficient data or an ineligible fit leaves this rule inactive.

The affine fit minimizes $\sum_{x \in S} \|AI_x + t - O_x\|_2^2$ over sampled background pixels S by ordinary least squares, without robust weighting; the luminance fallback minimizes its one-dimensional counterpart. These are per-output explanatory fits. Thresholds are fixed operational choices developed using reference checks and constructed counterexamples. Sensitivity and removal tests quantify the consequences of these choices (Appendices A.5 and A.7).

These rules detect specified incidental changes, rather than certifying instruction correctness. On all 1,740 cases, the rendered target receives $(100, 100, 100, 100)$ and the unchanged input $(0, 100, 0, 0)$ at the reported precision. The largest fidelity baseline is 0.9709, below the 0.995 cutoff.

B.3 Separate Color-attainment Measurement

CIEDE2000 accounts for perceptual nonuniformity in Lab color space; the color-agreement tests and change-region thresholds instead use Euclidean Lab distance (CIE76). For the 621 covered color-editing cases, we compare predicted colors with target references inside stable object cores. These cores follow the stored object metadata: two erosions with a 3×3 kernel, or distance-transform ridges for thin objects. A pixel matches its nearest target color within CIEDE2000 tolerance $\min(5, 0.4 d_{\mathrm{src,tgt}})$, where the source–target separation is recorded during construction; the tolerance is 5 when that separation is unavailable. Target references use these same cores. We retain a single-color reference if it covers the core sufficiently for full credit. Otherwise, and for multicolor references, we augment the stored colors with the most frequent uncovered target colors until every core pixel is within tolerance. Target shares use the same nearest-color assignment as prediction shares; unused color centers are removed. Reference construction never uses model outputs.

For a single-color object, matching fraction q gives $100 \,\mathrm{clip}_{[0,1]}(q/0.55)$. For a multicolor object, the score is $100 \sum_k \min(\widehat{s}_k, s_k)$, where s_k is the target share and $\widehat{s}_k$ the matched prediction share assigned to target color k. An object instead receives zero when at least 90% of its core matches the background and at most 10% of its full region differs from the background, using CIEDE2000 tolerance 5; this rule is disabled when the target itself resembles the background. Case scores average object scores. This diagnostic supports Section 4.5 and does not enter (F, P, L, M) or H.

B.4 Inference Configurations

Each model–case pair contributes one fixed output, with the exact input image, instruction, and inference settings retained in the run records.

Table 28 records the inference settings. Open-weight runs use BF16, except InstructPix2Pix (half-precision floating point, FP16), and seed 42, except FLUX.2-dev, which uses $(42 + \mathrm{CRC32}(\text{case ID})) \bmod 2^{31}$. For a reference canvas of size $W \times H$, the shared input wrapper centers the image on a white square of side $S = \max(W, H, 1024)$; for the released assets, this adds padding without scaling the image content. The open-weight editors use a 1024×1024 image for the generative branch, except InstructPix2Pix, which retains the incoming square size. Step1X-Edit v1.0 uses `size_level=1024`. GPT-Image-2 requests a 1024×1024 output through its API.

For every editor, the saved content box is scaled to the returned image's width and height, cropped to remove padding, and resized to the reference dimensions with Lanczos interpolation when needed. This coordinate mapping uses the recorded padding and canvas dimensions to return each output to the reference coordinate system.

Table 28: Inference settings for the evaluated editors. Checkpoint gives the exact public release evaluated; for open-weight models, Revision denotes the public repository commit. Guidance values follow each model's own parameterization; T/I denotes text/image guidance.

Model	Checkpoint	Revision	Steps	Guidance
GPT-Image-2	—	API	—	—
FireRed-Edit	FireRed-Edit-1.1	`3bc3f2a1`	40	True CFG 4.0
Qwen-Image-Edit-2511	Qwen-Image-Edit-2511	`6f3ccc0b`	40	True CFG 4.0
FLUX.2-dev	FLUX.2-dev	`26afe3a7`	50	4.0
Qwen-Image-Edit	Qwen-Image-Edit	`ac7f9318`	50	True CFG 4.0
BAGEL	BAGEL-7B-MoT	`5019f57d`	50	T/I: 4.0/1.5
Step1X-Edit	Step1X-Edit v1.0	`752c0bc2`	50	4.0
FLUX.2-klein	FLUX.2-klein-base-4B	`a3b4f484`	50	4.0
FLUX.1-Kontext	FLUX.1-Kontext-dev	`24e9dedc`	50	4.0
OmniGen2	OmniGen2	`df5dca8a`	50	T/I: 5.0/2.0
InstructPix2Pix	InstructPix2Pix	`31519b5c`	30	T/I: 7.5/1.5

FireRed-Edit and both Qwen-Image-Edit checkpoints use `true_cfg_scale=4.0` with a single-space negative prompt. BAGEL applies guidance over $[0.4, 1.0]$ with timestep shift 3.0. GPT-Image-2 uses the Azure image-editing API (version `2025-04-01-preview`), medium quality, and one 1024×1024 output per request; no seed, step count, or guidance scale is sent to the service.

C Coverage of Related Datasets

Table 29 compares representative general image-editing benchmarks with closely related chart and SVG editing benchmarks. The latter include both image-output and SVG-code editing interfaces. VeriEdit-Bench combines source-rendered references and separate pixel-based edit/preservation measurements across four design asset types.

Table 29: Asset coverage, editing operations, and verification in related datasets. ✓ indicates coverage documented in the paper's task taxonomy or examples; — means not explicitly reported there. ✓† denotes PaintBench's TinyGrafixBench extension. Presence does not imply equal task counts or coverage of every domain–operation combination.

(a) Asset coverage and verification

Dataset	Other visual domains	Design assets				Verification	
		SVG artwork	Charts	Web	Slides	Source target	Regional pixels
MagicBrush (Zhang et al., 2023)	Natural images	—	—	—	—	—	—
GIE-Bench (Qian et al., 2025)	Photographs; illustrations	—	—	—	—	—	—
PaintBench (Xu et al., 2026)	Shape canvases; network graphs†	—	✓†	—	—	✓	✓
StructBench (Zhuo et al., 2025)	Math, science, puzzles, graphs, tables	—	✓	—	—	✓	—
FigEdit (Li et al., 2025)	—	—	✓	—	—	✓	—
ChartSync (Yu et al., 2026a)	—	—	✓	—	—	✓	—
SVGEditBench V2 (Nishina & Matsui, 2025)	—	✓	—	—	—	✓	—
VectorEdits (Kuchar et al., 2026)	—	✓	—	—	—	✓	—
Vector-Bench (Gupta & Hebbar, 2026)	—	✓	—	—	—	✓	—
VeriEdit-Bench	—	✓	✓	✓	✓	✓	✓

(b) Editing operations

Dataset	Color	Geometry	Text / labels	Font size	Lines / borders	Opacity / blending	Hide / remove
MagicBrush	✓	—	✓	—	—	—	✓
GIE-Bench	✓	✓	✓	—	—	—	✓
PaintBench	✓	✓	—	—	✓	✓	✓
StructBench	✓	✓	✓	—	✓	—	✓
FigEdit	✓	✓	✓	✓	—	—	✓
ChartSync	—	✓	✓	—	—	—	—
SVGEditBench V2	✓	✓	—	—	✓	—	✓
VectorEdits	—	✓	—	—	✓	—	—
Vector-Bench	✓	✓	—	—	✓	✓	✓
VeriEdit-Bench	✓	✓	✓	✓	✓	✓	✓

Source target: a renderable reference program, including paired target SVGs. Regional pixels: separate target-agreement and preservation checks using fixed pixel rules. Vector-Bench verifies requested attributes and protected structure directly in SVG code; SVGEditBench V2 and VectorEdits combine rendered-image similarity with geometric or validity measures. All three evaluate SVG-code inputs and outputs. ChartSync combines textual and image measurements with VLM judgments of text–geometry consistency. VeriEdit-Bench evaluates image-and-instruction inputs and image outputs, with source programs hidden from editors. Geometry includes position, size, layout, or shape; lines/borders includes outline style and border appearance; opacity/blending includes alpha blending.