# RUBRICEDIT-BENCH: BENCHMARKING COMPLEX INFOGRAPHIC EDITING WITH FINE-GRAINED DIAGNOSTIC EVALUATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Instruction-guided image editing models have advanced rapidly, yet their capabilities on complex infographic editing remain insufficiently evaluated. Infographics tightly couple text, data, layout, and structural relations, while existing benchmarks provide limited coverage of such structured edits and holistic evaluation protocols offer little diagnostic information about specific failures. We introduce RubricEdit-Bench, a benchmark comprising 1,483 human-reviewed cases across four categories and 15 fine-grained tasks, with attributes characterizing target multiplicity, reasoning demand, and source-image complexity. RubricEdit-Bench further adopts a diagnostic evaluation protocol that decomposes each edit into typed atomic criteria and performs source-grounded, localized verification, enabling failures to be identified at the criterion level. We evaluate 15 frontier open- and closed-source editing models, with Overall scores ranging from 1.88 to 4.89 on a five-point scale. The best-performing model achieves exact success on only 85.2% of cases, indicating that complete task fulfillment remains challenging even for frontier models. Edits requiring complex reasoning show a 0.48-point performance drop compared with edits requiring little or no reasoning. Coordinated layout, data, and structural edits also remain particularly challenging. RubricEdit-Bench provides a systematic and diagnostic testbed for identifying these remaining capability gaps and guiding progress toward reliable, structure-aware image editing.

## 1 INTRODUCTION

Recent instruction-guided image editing models have made substantial progress in following increasingly complex instructions, from general-purpose editing to unified editing systems and practical open implementations (Brooks et al., 2023; Zhang et al., 2023; Sheynin et al., 2024; Yu et al., 2025; Liu et al., 2025). Infographic editing presents a particularly demanding setting, as infographics integrate text, graphics, numerical information, and structural relations within tightly coupled visual structures (Peng et al., 2025; Tang et al., 2026). A successful edit often requires precise localization of the target, preservation of semantic consistency across heterogeneous content, and coordinated changes among interdependent elements. Infographics therefore provide a valuable setting for examining whether image editing models can perform precise, semantically consistent, and structure-aware modifications.

Existing image editing benchmarks, however, only partially capture these demands. General-purpose image editing benchmarks cover a broad range of common operations, such as object manipulation, attribute modification and multi-turn editing, providing a useful overview of general editing capability (Zhang et al., 2023; Ma et al., 2024; Ye et al., 2025; Wu et al., 2025). More specialized benchmarks extend evaluation to particular content types, including text-centric images, visual documents, and charts (Tuo et al., 2024; Zhang et al., 2026; Yi et al., 2026; Yang et al., 2025; Chen et al., 2025). However, the coordinated layout, data, and structural editing capabilities required by infographics remain insufficiently represented. In addition, existing benchmarks are typically organized around task categories, while cases within the same category can differ substantially in the number of affected targets, the reasoning required to infer dependent changes, and the complexity of



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.

the source image. This limits their ability to isolate the factors that drive model failures in complex infographic editing.

Moreover, reliable evaluation of infographic editing remains challenging. A common paradigm uses VLM-based evaluators to assign either a holistic score or scores along a small set of evaluation aspects. These evaluators can also provide reasons for their judgments, but both the scoring process and the accompanying explanations often remain coarse-grained, making it difficult to identify localized semantic or consistency errors. Recent studies explore finer-grained evaluation through question-based verification, factorized judgments, verifier-style decomposition, and adaptive rubrics (Hu et al., 2023; Tang et al., 2026; Liu et al., 2026; Guo et al., 2026; Kan et al., 2026). While these approaches provide finer-grained judgments, the visual evidence supporting each judgment is often not explicitly grounded. Without a clear link between an evaluation result and the relevant image regions or visual relationships, evaluators may overlook localized inconsistencies or produce judgments that are not fully supported by the visual content, reducing scoring reliability. This issue is particularly important for complex infographic edits, where correctness often depends on small local details and consistency among multiple related elements.

To address these limitations, we introduce **RubricEdit-Bench**, an all-in-one benchmark for complex infographic editing. It comprises 1,483 human-reviewed cases organized into four high-level categories and 15 fine-grained tasks, together with difficulty attributes that characterize multi-target editing, reasoning demand, and source-image complexity. RubricEdit-Bench further incorporates a fine-grained evaluation methodology that decomposes editing instructions into atomic criteria and performs grounded, localized verification. This design enables both systematic model comparison and diagnostic analysis of individual editing failures.

Using RubricEdit-Bench, we systematically evaluate 15 open- and closed-source image editing models. Our results reveal substantial differences across models and editing capabilities, and show that strong visual quality does not necessarily imply faithful execution of editing instructions. Current models show persistent weaknesses in complex data and structural manipulations. Further analyses by task and difficulty attribute reveal systematic variation in model performance. These findings unveil the capabilities and remaining limitations of current image editing models and can also inform future research.

Our main contributions are summarized as follows:

**i) An All-in-One Benchmark for Complex Infographic Editing.** We introduce **RubricEdit-Bench**, which integrates human-reviewed benchmark data, an infographic-specific task taxonomy, difficulty attributes, and a fine-grained evaluation methodology based on atomic criterion decomposition, grounding, and localized verification.

**ii) A Systematic Analysis of Image Editing Models.** We benchmark 15 open- and closed-source image editing models and systematically analyze their capabilities and limitations across tasks and difficulty factors, revealing persistent weaknesses in faithful, data-aware, and structure-aware editing and providing insights for future image editing research.

## 2 Related Work

**Image Editing Benchmarks.** Instruction-guided image editing has progressed from task-specific manipulation toward general-purpose systems capable of handling increasingly diverse and complex instructions (Brooks et al., 2023; Zhang et al., 2023; Liu et al., 2025). Accordingly, recent benchmarks have expanded beyond basic object manipulation to cover broader editing operations, complex instructions, reasoning-intensive tasks, and practical scenarios (Ye et al., 2025; Zhou et al., 2026; Bai et al., 2026). While these benchmarks substantially broaden the scope of general image editing evaluation, their task spaces are largely organized around editing operations or model capabilities, and editing difficulty is often characterized through task categories or dedicated benchmark subsets rather than factorized instance-level attributes. Meanwhile, several benchmarks target structured visual content. WeEdit studies multilingual text-centric editing (Zhang et al., 2026), VDE Bench focuses on dense visual-document editing (Yi et al., 2026), and chart-oriented benchmarks evaluate structured chart modifications (Yang et al., 2025; Chen et al., 2025). In the closely related infographic domain, IGenBench evaluates text-to-infographic generation through atomic questions covering textual content, data, and visual encodings (Tang et al., 2026). However, infographic *edit-*

*ing* introduces an additional source-conditioned requirement: models must modify requested content while maintaining consistency among text, data encodings, layout, and structural relations, as well as preserving unrelated source content. RubricEdit-Bench is designed for this setting, with a unified task taxonomy spanning element, layout, data, and structure editing, together with factorized attributes for analyzing what makes an editing instance difficult.

**Image Editing Evaluation.** Image editing evaluation has gradually moved beyond perceptual similarity and global image–text alignment toward VLM-based assessment and editing-specific reward models (Ku et al., 2023; Wu et al., 2026; Luo et al., 2026). A central limitation of holistic scoring is that a single score may obscure which requirement of a complex instruction has failed, especially when an edit involves multiple dependent visual constraints. Recent work therefore explores more structured evaluation paradigms. Edit-R1 decomposes editing instructions into verifiable principles and evaluates them individually (Guo et al., 2026), while RubricRM constructs instruction-specific evaluation dimensions and scoring criteria (Kan et al., 2026). Question-based evaluation offers a related approach: TIFA evaluates generated images using prompt-derived questions (Hu et al., 2023), and IGenBench applies atomic yes/no verification to infographic generation (Tang et al., 2026). These approaches demonstrate the value of decomposing complex requirements, but infographic editing further requires reasoning over both the source image and the edited output. RubricEdit-Bench therefore adopts instance-specific atomic criteria together with source-image grounding and localized source–output verification, enabling the evaluation to separately examine requested modifications, consistency among dependent elements, and preservation of non-target content.

# 3 RUBRICEDIT-BENCH

RubricEdit-Bench provides a systematic testbed for complex infographic editing, covering localized element manipulation as well as edits involving layout, data, and structural relations. It is designed to characterize editing capabilities, case difficulty, and editing correctness at a fine-grained level.

## 3.1 BENCHMARK DATA

RubricEdit-Bench contains **1,483 source–instruction pairs**. Each case consists of a unique source infographic and a single-turn instruction specifying a local edit to be performed on the source image.

We construct the benchmark through a VLM-assisted, human-verified curation pipeline. Candidate source images are sampled from several large-scale infographic and image-editing corpora. To improve coverage of structured editing scenarios that are scarce in the data, we additionally construct task-specific source infographics with controlled elements and relations, such as chart encodings, information modules, graph connections, and hierarchical or process structures. GPT-5.5 is used to generate image-grounded editing instructions, audit task assignments, and filter candidate sources and source–instruction pairs. Sources with visible artifacts, poor legibility, or structural defects, as well as cases with ambiguous targets, invalid instructions, or infeasible edits, are removed. We further conduct a strict human review of 2,280 source images with six reviewers working on disjoint subsets, retaining 1,915 images. After task-aware balancing and final quality control, 1,483 cases are retained for the final benchmark.

The final set provides near-balanced coverage of the fine-grained tasks defined in Section 3.2, with 95–100 cases per task. It also spans the editing conditions introduced in Section 3.3, including 736/747 single-/multi-target cases, 1,082/147/254 R0/R1/R2 cases, and 1,021/462 simple/complex-source cases. Source images contain 46.4% English, 49.0% Chinese, and 4.7% mixed Chinese–English content, and cover diverse resolutions and aspect ratios. Detailed construction procedures and statistics are provided in Appendix A.

## 3.2 TASK TAXONOMY

Each case in RubricEdit-Bench consists of one source infographic and one instruction specifying a localized edit. We define **15 fine-grained tasks** to cover the major editing operations considered in the benchmark, as shown in Figure 1. These tasks can be further grouped into four high-level categories according to the type of content or relation being edited: **Element**, **Layout**, **Data**, and **Structure**. Detailed task definitions and category-wise statistics are provided in Appendix B.

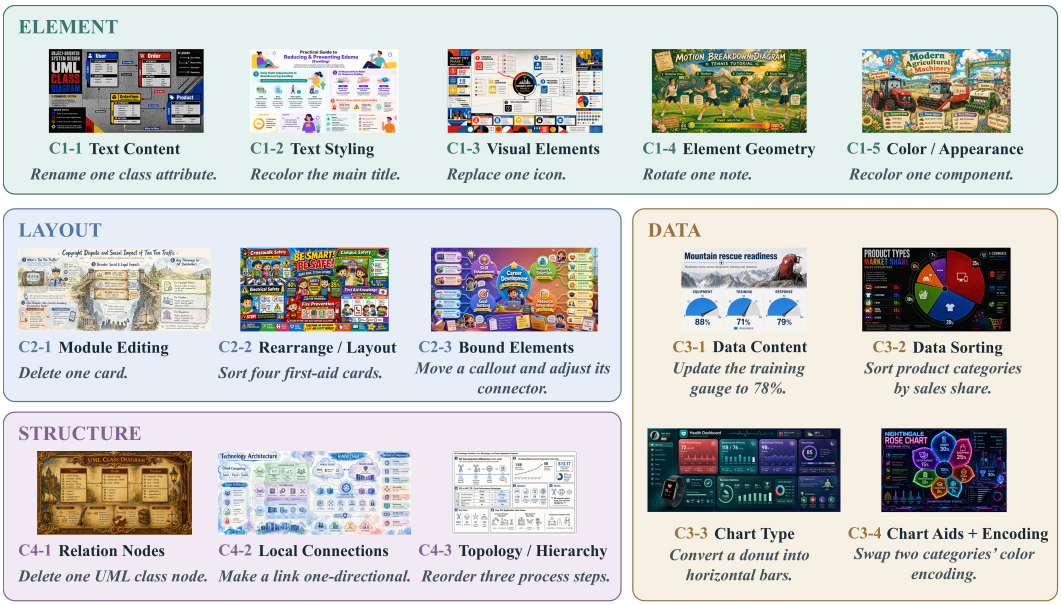



Figure 1: **Task taxonomy of RubricEdit-Bench.** 15 fine-grained editing tasks are grouped into four high-level categories: Element, Layout, Data, and Structure.

**Element** includes relatively fundamental and commonly studied operations on text and visual elements, such as content modification, style adjustment, geometric transformation, and appearance editing. Such operations are also widely considered in general image-editing benchmarks. **Layout**, **Data**, and **Structure** focus on more structured infographic editing scenarios. Layout tasks involve information modules, spatial organization, and bound elements; Data tasks involve numerical content, ordering, chart representation, and visual encoding; Structure tasks involve nodes, connections, hierarchies, and process topology. These categories place greater emphasis on maintaining consistency among semantically related visual elements, which is a central challenge in complex infographic editing.

### 3.3 Difficulty Attributes

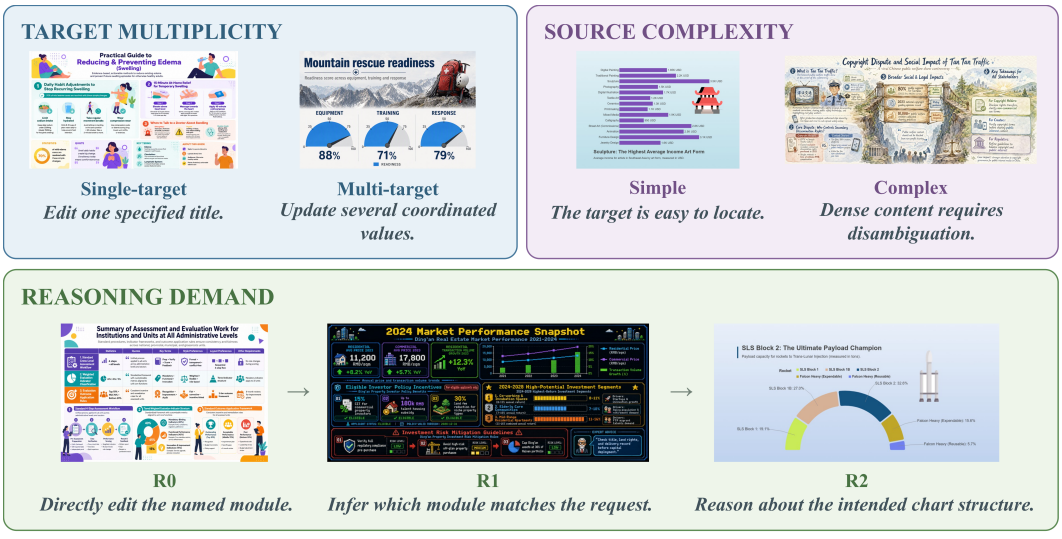



Figure 2: **Difficulty attributes in RubricEdit-Bench.** Representative cases illustrate target multiplicity, reasoning demand, and source difficulty.

Cases within the same task can differ substantially in editing difficulty. We therefore annotate each case with three complementary diagnostic attributes: **target multiplicity**, **reasoning demand**, and **source difficulty**, as illustrated in Figure 2.

**Target multiplicity** distinguishes edits involving a single target from those requiring changes to multiple distinguishable targets. **Reasoning demand** is divided into three levels: R0 directly specifies the target and desired result; R1 requires lightweight inference or target selection from the source image; and R2 requires combining multiple elements to derive an ordering, mapping, relation, or structure. **Source difficulty** characterizes how difficult the target is to locate or disambiguate in the source image, with *simple* and *complex* levels.

These attributes are treated as complementary diagnostics rather than independent difficulty factors. Together, they allow subsequent analysis to separate different conditions associated with model performance degradation.

### 3.4 Evaluation Methodology

Infographic edits combine textual, visual, spatial, and relational requirements, and the same instruction can therefore lead to very different failure modes. We introduce an evaluator that turns each edit into a set of locally inspectable decisions instead of relying on a single holistic score. The complete procedure is summarized in Figure 3.

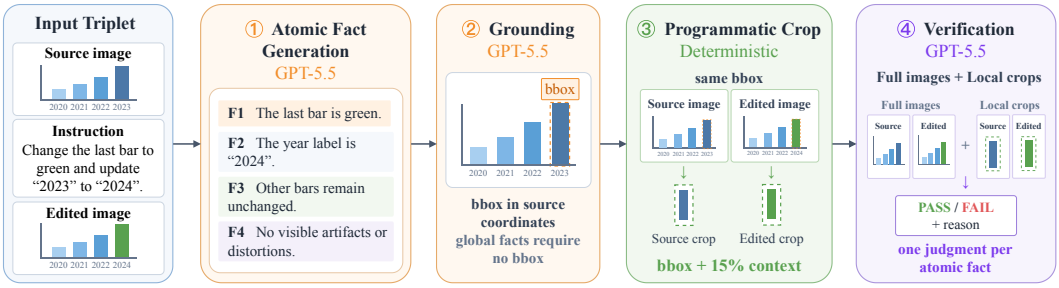



Figure 3: The evaluation procedure decomposes an edit into verifiable criteria, grounds local evidence, and produces criterion-level judgments from matched source–output regions.

#### 3.4.1 Limitations of Holistic Evaluation

Taking one case in our benchmark (000730) as an example, the previous GPT-5.5 judge assigned the SenseNova U1 (see more details in Appendix J) result an overall score of 2.33/5, but this scalar fail to identify which requirements caused the deductions or where the corresponding strengths and defects occur. A free-form holistic rationale can likewise be difficult to reconcile with observable image evidence. Figure 4 illustrates the same case and the evidence used by our evaluator.

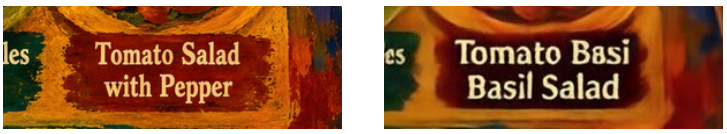
#### 3.4.2 Evaluation Procedure

**Atomic criterion decomposition.** For a case $x = (I_s, q, I_e)$, we represent its requirements as $\mathcal{C}(x) = \{c_i\}_{i=1}^{K}$, where each $c_i = (r_i, t_i, b_i, y_i, e_i)$ contains one falsifiable requirement, its type, a source-coordinate evidence box or `null`, a binary verdict, and observable evidence. The intended final state is inferred from the source and instruction; the edited image supplies evidence but does not redefine that target. Criteria are typed as **Instruction Following (IF)**, **Preservation (PRES)**, or **Visual Quality (VQ)**. IF covers the requested result and necessary dependent changes, PRES covers relevant content that should remain unchanged, and VQ covers applicable integrity, readability, alignment, and rendering quality. Four representative judgments for case 000730 are shown in Table 1; complete wording and normalized coordinates are provided in Appendix C.

**Source-coordinate grounding.** For every local criterion, the grounding stage assigns at most one normalized source-coordinate box; global or unreliable-to-localize criteria receive `null`. This makes the evidence location explicit while keeping the criterion text and identity fixed. Grounding records and quality gates are detailed in Appendix C.




(a) Source infographic (b) Edited output




(c) Source crop (d) Edited crop

Figure 4: **Generation model:** SenseNova U1. **Instruction:** Change the label "Tomato Salad with Pepper" beneath the rightmost bottom dish thumbnail to "Tomato Basil Salad," keeping the cream lettering style within the existing red plaque.

Table 1: Representative judgments for case 000730. Criteria are abbreviated for presentation.

| ID | Type | Criterion | Verdict | Observable evidence |
|---|---|---|---|---|
| 1 | IF | The target label reads exactly "Tomato Basil Salad." | FAIL | The output reads "Tomato Basi" and "Basil Salad," not the requested label. |
| 4 | IF | The target label remains inside the existing red plaque. | PASS | The edited label remains within the plaque beneath the rightmost dish. |
| 6 | PRES | The seven non-target bottom-row labels remain unchanged. | FAIL | Several unrelated labels are altered or garbled. |
| 11 | VQ | The full infographic remains clear and readable without global degradation. | FAIL | The output is globally softened and contains unreadable text. |

**Programmatic paired crops.** After the edited image is normalized to the source geometry, the same context-expanded region is cropped from both images. The paired crop uses the grounded box with a 15% expansion and is constructed deterministically, so the verifier compares corresponding source and edited content rather than independently selected regions. An example of the edited crop is shown in Figure 4 (d).

**Localized verification and aggregation.** The verifier receives the full images and paired crops and returns PASS, FAIL, or UNSURE with a concise observable reason. Criterion verdicts are summarized by a micro pass rate and type-balanced five-point scores for model comparison; these summaries are intentionally secondary to the inspectable criterion records. Complete geometry, validation, and aggregation rules are given in Appendix C.

### 3.4.3 Specialized Evaluator Training

To make this evaluator reusable and reduce the cost of repeated calls to a general-purpose vision–language model, we train a dedicated infographic-editing evaluator initialized from Qwen3.5-9B. Detailed information on this model is provided in Appendix J.

**Training data construction.** We retain approximately 45k benchmark-clean triplets spanning the 15 task categories and a mixture of proprietary and open/in-house editors. GPT-5.5 serves as the teacher because it provides strong visual recognition and localization; the same four stages above generate atomic criteria, source grounding, deterministic crops, and localized judgments. The re-

tained triplets do not overlap with benchmark triplets. Corpus statistics and the exact stage messages are provided in Appendix D and E.

**Model training.** We fine-tune Qwen3.5-9B for three epochs and select the epoch-2 checkpoint at step 1264 as the final SFT evaluator. We name this checkpoint *SFT-1264* as our original version. Starting from that, GDPO is run for 100 steps to improve four structured-output properties: format validity, checklist quality, grounding quality, and verification quality, producing the diagnostic *RL-100* evaluator as our final version. Training hyperparameters, checkpoint selection, curves, and reward construction are reported in Appendix D and G.1.

**Structured-output reliability.** Among 22,212 available case–model pairs from 15 editors, SFT-1264 and RL-100 produce 22,197 (99.93%) and 22,201 (99.95%) valid records, respectively. Invalid records are excluded from score aggregation rather than converted to numerical failures.

**Agreement with GPT-5.5.** Table 2 compares both evaluators with GPT-5.5 on the strict intersection of 22,167 valid, identity-verified case–model pairs. GPT-5.5 assigns independent 1–5 scores for IF, PRES, and VQ and their unweighted mean is the reference overall score; its exact prompt and input ordering are in Appendix G.2. Although SFT-1264 shows slightly stronger benchmark-scale agreement with GPT-5.5, RL-100 achieves higher agreement on the human-preference diagnostic set and is used as the primary evaluator for the systematic analysis in Section 4.

Table 2: Agreement with GPT-5.5 on 22,167 matched pairs. Bias is evaluator score minus GPT-5.5 score; rank $\rho$ uses mean scores for 15 editors over the 1,422-case common subset used for cross-model comparison.

| Evaluator | Pearson $r$ | Spearman $\rho$ | MAE $\downarrow$ | Bias | Rank $\rho$ |
|---|---|---|---|---|---|
| SFT-1264 | **0.8615** | **0.8735** | **0.6132** | +0.4426 | **0.9857** |
| RL-100 | 0.8463 | 0.8695 | 0.7164 | +0.6241 | 0.9821 |

**Agreement with existing human preferences.** Both checkpoints cover the same 88 of 118 pairs in the disagreement-enriched human diagnostic set (Table 3). The pairs and three-annotator majority procedure are described in Appendix H. RL-100 has stronger agreement on this diagnostic subset.

Table 3: Agreement with three-annotator majority preference on the 88 pairs covered by both evaluator checkpoints. The diagnostic set emphasizes historical judge disagreements.

| Judge | Correct / decisive | Decisive acc. | 95% Wilson CI | Ties | Ties as errors |
|---|---|---|---|---|---|
| GPT-5.5 | 59 / 86 | 68.6% | 58.2–77.4% | 2 | 67.0% |
| SFT-1264 | 56 / 84 | 66.7% | 56.1–75.8% | 4 | 63.6% |
| RL-100 | **62 / 82** | **75.6%** | 65.3–83.6% | 6 | **70.5%** |

Finally, to isolate the value of inspectable atomic evidence beyond aggregate dimension scores, we conduct an evidence-utility study using the existing human-preference pairs. The blinded reader protocol and experimental results are provided in Appendix I; no new images or human annotations are introduced.

## 4 Benchmarking and Analysis

We evaluate 15 representative image editing models on RubricEdit-Bench using the final RL-100 evaluator. All scores are reported on a 1–5 scale, with Overall computed as the mean of IF, PRES, and VQ. For fair cross-model comparison, we report results on the 1,422 cases completed by all 15 models; the remaining 61 were blocked by provider-side safety filters for at least one model.

### 4.1 Overall Capability Landscape

Table 4 shows substantial performance differences across the 15 evaluated editors. GPT Image 2.5, GPT Image 2, Seedream 5.0 Pro, and Qwen-Image 3.0 Pro form a leading group, outperforming the remaining models by a large margin in Overall score.

Table 4: Performance of 15 image editing models on the strict 1,422-case common subset of RubricEdit-Bench using RL-100. All scores are on a 1–5 scale, and Overall is the mean of IF, PRES, and VQ. Detailed information on open-source models is provided in Appendix J.

| Model | IF | PRES | VQ | Overall |
|---|---|---|---|---|
| GPT Image 2.5 | 4.811 | 4.920 | 4.929 | 4.887 |
| GPT Image 2 | 4.758 | 4.864 | 4.932 | 4.852 |
| Seedream 5.0 Pro | 4.670 | 4.863 | 4.830 | 4.788 |
| Qwen-Image 3.0 Pro | 4.652 | 4.829 | 4.795 | 4.759 |
| SenseNova U1.5 | 4.364 | 4.467 | 4.197 | 4.343 |
| Nano Banana 2 | 4.577 | 4.156 | 4.287 | 4.340 |
| SenseNova U1 | 3.862 | 4.090 | 3.582 | 3.845 |
| Hunyuan Image 3 | 3.670 | 3.968 | 3.447 | 3.695 |
| Boogu | 3.431 | 3.486 | 2.864 | 3.260 |
| FLUX.2-dev | 2.936 | 3.187 | 2.751 | 2.958 |
| BAGEL | 2.881 | 3.587 | 2.249 | 2.905 |
| FireRed-Image-Edit | 3.066 | 2.942 | 2.564 | 2.857 |
| Qwen-Image-Edit-2511 | 2.849 | 2.643 | 2.312 | 2.601 |
| Step1X-Edit-v1p2 | 2.742 | 2.981 | 2.061 | 2.595 |
| OmniGen2 | 2.067 | 1.978 | 1.581 | 1.875 |

The three evaluation dimensions are strongly correlated across models (Pearson $r = 0.96$–$0.99$), but they are not interchangeable. For example, Nano Banana 2 attains substantially higher IF than PRES ($+0.421$), whereas BAGEL shows the opposite profile, with PRES exceeding IF by $0.706$. Thus, a single Overall score can conceal different trade-offs between executing the requested edit, preserving unrelated content, and maintaining visual quality.

### 4.2 Bottlenecks in Structured Infographic Editing

The fine-grained task breakdown reveals a consistent capability pattern. The highest pooled scores are observed for *visual element geometric transformation* (C1-4, 4.090) and *text style editing* (C1-2, 3.990), followed by *chart auxiliary structure and encoding editing* (C3-4, 3.892). In contrast, the most challenging tasks are *module rearrangement and layout adjustment* (C2-2, 3.129), *data sorting and rearrangement* (C3-2, 3.167), *topology, hierarchy, and flow restructuring* (C4-3, 3.370), *chart type conversion* (C3-3, 3.433), and *data content editing* (C3-1, 3.582). The four lowest-scoring categories all require multiple semantically related visual components to remain coordinated after the edit. Overall, current models handle spatially or semantically isolated modifications more reliably than edits that require coordinated updates to layout, data bindings, or relational structure.

Atomic evaluation further reveals a substantial gap between partial and complete task success (Figure 5). Even GPT Image 2.5 achieves a 96.9% atomic pass rate but only 85.2% exact case success, with larger gaps for weaker models.

### 4.3 What Makes an Edit Difficult?

We next examine which diagnostic attributes are associated with performance degradation. Because these attributes are correlated with task category, we fit a regression over the 21,330 case–model scores with target multiplicity, reasoning demand, and source difficulty as predictors, while controlling for task category and model fixed effects. Standard errors are clustered by case. The coefficients therefore capture conditional associations rather than causal effects.

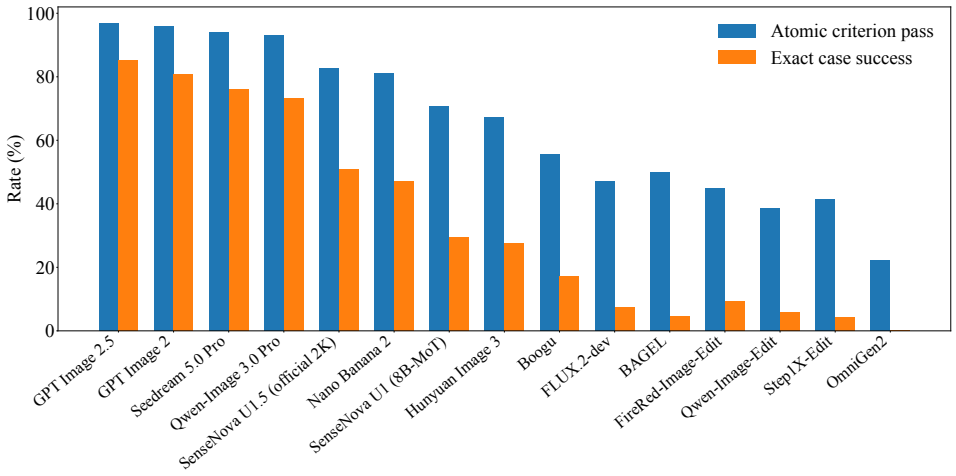



Figure 5: Atomic criterion pass rate versus exact case success on the strict 1,422-case subset. Exact success requires all applicable atomic criteria of a case to pass. The persistent gap shows that high average criterion-level performance does not necessarily translate into complete task-level correctness.

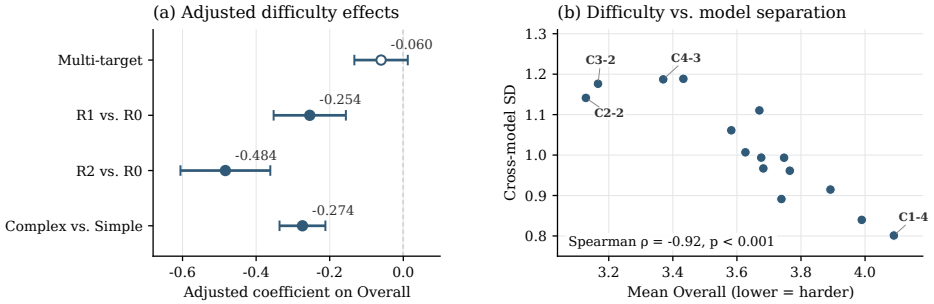



Figure 6: Difficulty effects and discriminative power. **(a)** Adjusted associations between diagnostic difficulty attributes and Overall score; horizontal bars denote 95% confidence intervals. R0, single-target, and simple-source cases are the reference conditions. **(b)** Relationship between L2-level difficulty and model separation. Each point represents one L2 category; difficulty is measured by mean Overall score, while model separation is measured by the standard deviation across model-level category means.

As shown in Figure 6(a), reasoning demand exhibits the strongest association with performance degradation. Relative to R0, R1 and R2 are associated with $0.254$- and $0.484$-point decreases in Overall score, respectively, while complex source images incur a $0.274$-point decrease. In contrast, the effect of target multiplicity is small and non-significant ($\beta = -0.060$, $p = 0.103$).

At the task-category level, harder L2 categories also exhibit substantially greater cross-model variation (Figure 6(b), Spearman $\rho = -0.92$, $p < 0.001$), indicating that difficult task categories also provide stronger discrimination among editing models.

## 5 Conclusion

We introduce RubricEdit-Bench, a benchmark for complex infographic editing with a structured task taxonomy, instance-level difficulty attributes, and fine-grained atomic evaluation. Evaluating 15 image editing models reveals persistent weaknesses in coordinated layout, data, and structural edits, with reasoning demand and source complexity strongly associated with performance degradation. Moreover, high criterion-level accuracy does not necessarily translate into complete task success, highlighting the need for more reliable dependency-aware editing of structured visual content.

## AI USE STATEMENT

Generative AI tools were used to assist with writing and language polishing, literature retrieval and discovery, and the generation of synthetic data used in benchmark construction. All AI-assisted content and generated data were reviewed and verified by the authors. The authors take responsibility for the final content of this work.

## REFERENCES

Xuehai Bai, Yang Shi, Yi-Fan Zhang, Xuanyu Zhu, Yuran Wang, Yifan Dai, Xinyu Liu, Yiyan Ji, Xiaoling Gu, and Yuanxing Zhang. Edit-compass & editreward-compass: A unified benchmark for image editing and reward modeling. *arXiv preprint arXiv:2605.13062*, 2026.

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*, 2023.

Liangyu Chen, Yichen Xu, Jianzhe Ma, Yuqi Liu, Donglu Yang, Liang Zhang, Zihao Yue, Wenxuan Wang, and Qin Jin. Charteditor: A reinforcement learning framework for robust chart editing. *arXiv preprint arXiv:2511.15266*, 2025.

Hanzhong Guo, Jie Wu, Jie Liu, Yu Gao, Zilyu Ye, Linxiao Yuan, Xionghui Wang, Yizhou Yu, and Weilin Huang. Leveraging verifier-based reinforcement learning in image editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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

Zijian Kan, Wei Wang, Long Luo, Bing Zhao, Xuan Ren, Weixu Qiao, Wenbo Li, Hu Wei, and Lin Qu. Rubricrm: Generative reward modeling via dynamic rubrics for image generation and editing. In *Conference on Empirical Methods in Natural Language Processing*, 2026.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. VIEScore: Towards explainable metrics for conditional image synthesis evaluation. *arXiv preprint arXiv:2312.14867*, 2023.

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. *arXiv preprint arXiv:2602.13028*, 2026.

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

Xin Luo, Jiahao Wang, Chenyuan Wu, Shitao Xiao, Xiyan Jiang, Defu Lian, Jiajun Zhang, Dong Liu, and Zheng Liu. Editscore: Unlocking online RL for image editing via high-fidelity reward modeling. In *International Conference on Learning Representations*, 2026.

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 *Advances in Neural Information Processing Systems*, 2024.

Yuyang Peng, Shishi Xiao, Keming Wu, Qisheng Liao, Bohan Chen, Kevin Lin, Danqing Huang, Ji Li, and Yuhui Yuan. Bizgen: Advancing article-level visual text rendering for infographics generation. *arXiv preprint arXiv:2503.20672*, 2025.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yinghao Tang, Xueding Liu, Boyuan Zhang, Tingfeng Lan, Yupeng Xie, Jiale Lao, Yiyao Wang, Haoxuan Li, Tingting Gao, Bo Pan, et al. IGenBench: Benchmarking the reliability of text-to-infographic generation. *arXiv preprint arXiv:2601.04498*, 2026.

Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. In *International Conference on Learning Representations*, 2024.

Keming Wu, Sicong Jiang, Max Ku, Ping Nie, Minghao Liu, and Wenhu Chen. Editreward: A human-aligned reward model for instruction-guided image editing. In *International Conference on Learning Representations*, 2026.

Yongliang Wu, Zonghui Li, Xinting Hu, Xinyu Ye, Xianfang Zeng, Gang Yu, Wenbo Zhu, Bernt Schiele, Ming-Hsuan Yang, and Xu Yang. KRIS-bench: Benchmarking next-level intelligent image editing models. *arXiv preprint arXiv:2505.16707*, 2025.

Donglu Yang, Liang Zhang, Zihao Yue, Liangyu Chen, Yichen Xu, Wenxuan Wang, and Qin Jin. ChartM3: Benchmarking chart editing with multimodal instructions. In *ACM International Conference on Multimedia*, 2025.

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. *arXiv preprint arXiv:2505.20275*, 2025.

Hongzhu Yi, Yujia Yang, Yuanxiang Wang, Tong Li, Zhenyu Guan, Tianyu Zong, Jiahuan Chen, Chenxi Bao, Tiankun Yang, Haopeng Jin, et al. VDE bench: Evaluating the capability of image editing models to modify visual documents. *arXiv preprint arXiv:2602.00122*, 2026.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Hui Zhang, Juntao Liu, Zongkai Liu, Liqiang Niu, Fandong Meng, Zuxuan Wu, and Yu-Gang Jiang. Weedit: A dataset, benchmark and glyph-guided framework for text-centric image editing. *arXiv preprint arXiv:2603.11593*, 2026.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems*, 2023.

Qinye Zhou, Jun Zheng, Yongchao Du, Yuan Wang, Zhengrui Chen, Zuan Gao, Taihang Hu, Chao Lin, Yefeng Shen, Xingjian Wang, Zhao Wang, Zhengtao Wu, Xiaoli Xu, Zhengze Xu, Hao Yan, Denghui Yang, Yuhang Yu, Huayu Zhang, Mingzhou Zhang, and Mengting Chen. Cpi-bench: A comprehensive, practical and intelligent benchmark for real-world image editing. *arXiv preprint arXiv:2608.14546*, 2026.

# A Detailed Benchmark Statistics

Figure 7 provides detailed statistics of the final RubricEdit-Bench. The 15 L2 task categories are closely balanced, with 95–100 cases per category, and target multiplicity is approximately evenly distributed between single-target and multi-target cases. Reasoning demand and source complexity exhibit more varied distributions, with R0 and simple-source cases occurring more frequently.

The benchmark also covers diverse linguistic and visual characteristics. English and Chinese cases account for 46.4% and 49.0% of the benchmark, respectively, with the remaining 4.7% containing mixed Chinese–English content. The source images span aspect ratios from 0.43 to 2.36 and a broad range of original image resolutions.

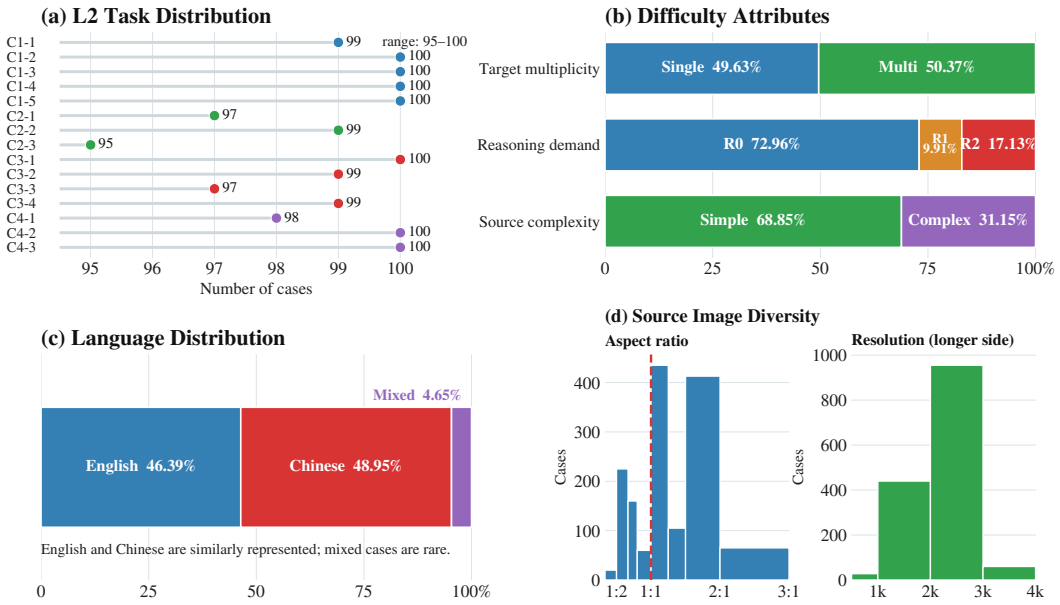



Figure 7: **Dataset composition of RubricEdit-Bench.** (a) Distribution across the 15 L2 task categories. (b) Distributions of target multiplicity, reasoning demand, and source complexity. (c) Language distribution. (d) Distributions of source-image aspect ratios and resolutions.

# B RubricEdit-Bench Details

## B.1 Complete Task Taxonomy

Table 5 provides the complete task taxonomy of RubricEdit-Bench. Each benchmark instance is assigned to one primary L2 category according to the semantic operation required by the editing instruction.

# C Complete Evaluation Example and Implementation Details

## C.1 Case 000730

The editing instruction is: *Change the label "Tomato Salad with Pepper" beneath the rightmost bottom dish thumbnail to "Tomato Basil Salad," keeping the cream lettering style within the existing red plaque*. Table 6 reports the complete criterion record produced for the SenseNova U1 output shown in Figure 4.

Table 5: Complete task taxonomy of RubricEdit-Bench. The benchmark contains four high-level categories and 15 fine-grained editing tasks.

| L1 | ID | Task | Definition | # |
|---|---|---|---|---|
| Element | C1-1 | Text content editing | Modify, add, remove, or replace textual content while preserving its intended role in the infographic. | 99 |
| | C1-2 | Text style editing | Modify visual properties of text, such as font appearance, color, emphasis, or alignment, without changing its semantic content. | 100 |
| | C1-3 | Non-data visual element editing | Add, remove, replace, or modify visual elements that do not directly encode quantitative data. | 100 |
| | C1-4 | Visual element geometric transformation | Modify the position, scale, orientation, or other geometric properties of a visual element. | 100 |
| | C1-5 | Local color and appearance editing | Modify local visual attributes such as color, fill, or appearance while preserving the identity and role of the target element. | 100 |
| Layout | C2-1 | Information module editing | Add, remove, or modify an information module composed of multiple visually or semantically related elements. | 97 |
| | C2-2 | Module rearrangement and layout adjustment | Reposition or reorganize information modules while maintaining the semantic content and overall layout coherence. | 99 |
| | C2-3 | Coordinated editing of bound elements | Jointly modify multiple visually or semantically bound elements whose spatial organization should remain consistent. | 95 |
| Data | C3-1 | Data content editing | Modify quantitative values, categories, or other data content represented in the infographic. | 100 |
| | C3-2 | Data sorting and rearrangement | Reorder or rearrange data items according to specified values, categories, or criteria while maintaining correct correspondence. | 99 |
| | C3-3 | Chart type conversion | Convert one chart representation into another while preserving the underlying data and semantic meaning. | 97 |
| | C3-4 | Chart structure and visual encoding editing | Modify chart components or visual encodings, such as axes, legends, markers, or mappings between data and graphical properties. | 99 |
| Structure | C4-1 | Relational node editing | Add, remove, replace, or modify nodes participating in an explicit relational structure. | 98 |
| | C4-2 | Local connection editing | Add, remove, redirect, or modify connections between relational elements. | 100 |
| | C4-3 | Topology, hierarchy, and process editing | Modify higher-order relational organization, including topology, hierarchy, ordering, or multi-step process structure. | 100 |
| | | | **Total** | **1,483** |

Table 6: Complete criterion record for case 000730. Bboxes use normalized source-image coordinates.

| ID | Type | Atomic criterion | BBox | Verdict | Observable evidence |
|---|---|---|---|---|---|
| 1 | IF | The label beneath the rightmost bottom dish thumbnail reads exactly "Tomato Basil Salad." | `790,688,895,735` | FAIL | The edited label reads "Tomato Basi" on the first line and "Basil Salad" on the second, not exactly "Tomato Basil Salad." |
| 2 | IF | The "Tomato Basil Salad" label is placed beneath the rightmost bottom dish thumbnail that shows sliced tomatoes in a yellow bowl. | `785,590,895,735` | PASS | The edited label is directly beneath the rightmost bottom dish showing sliced tomatoes in a yellow bowl. |
| 3 | IF | The "Tomato Basil Salad" label uses cream-colored lettering consistent with the original bottom dish labels. | `790,688,895,735` | PASS | The edited label uses cream-colored lettering matching the other bottom dish labels. |
| 4 | IF | The "Tomato Basil Salad" label remains inside the existing red plaque beneath the rightmost bottom dish thumbnail. | `790,688,895,735` | PASS | The edited label remains within the existing red plaque beneath the rightmost dish. |
| 5 | PRES | The rightmost bottom dish thumbnail continues to show the same tomato salad dish with sliced tomatoes in a yellow bowl. | `785,590,895,690` | PASS | The rightmost bottom dish still shows sliced tomatoes in a yellow bowl. |
| 6 | PRES | The other seven bottom dish labels remain unchanged: "Grilled Steak with Rosemary," "Lobster Thermidor," "Salmon with Asparagus," "Chicken with Herb Butter," "Lamb Chop with Mint Sauce," "Scallops with Arugula," and "Roasted Vegetables with Thyme." | `115,688,800,735` | FAIL | Several other bottom labels are altered or garbled, including "Scallnps with Arngula" and "Chicken with Hetb Butter." |
| 7 | PRES | The four ingredient classification panels retain their original headings, icons, colors, and ingredient lists for Meats, Seafood, Vegetables, and Spices. | `150,165,840,550` | PASS | The four classification panels keep the same headings, icons, colors, and ingredient lists. |
| 8 | PRES | The main title "Western Cuisine Ingredients" and subtitle "Ingredient Classification" remain unchanged at the top of the infographic. | `175,15,870,165` | PASS | The top title and subtitle still read "Western Cuisine Ingredients" and "Ingredient Classification." |
| 9 | PRES | The lower "Pairing Principles" and "Food Culture" sections retain their original headings, panel layout, icons, and text content. | `115,735,890,950` | FAIL | The lower section text is visibly garbled in multiple panels, especially the Food Culture captions. |
| 10 | PRES | The overall infographic composition, including the surrounding wine bottle, chalkboards, herbs, oil bottle, pepper grinder, and decorative food illustrations, remains unchanged except for the specified label text. | `null` | FAIL | Unrelated text areas outside the specified label are altered or garbled, including bottom dish labels and lower section text. |
| 11 | VQ | The edited image remains clear, complete, and readable across the full infographic without global blur, cropping, warping, or rendering degradation. | `null` | FAIL | The edited image has noticeable global softening and multiple unreadable or garbled text areas. |
| 12 | VQ | The edited label area beneath the rightmost bottom dish is cleanly rendered, with legible text that fits naturally within the plaque and does not overlap the dish image or neighboring labels. | `790,688,895,735` | FAIL | The edited label is legible and fits in the plaque, but the text is not cleanly rendered because it reads "Tomato Basi / Basil Salad." |

### C.2 Geometry, Verification, and Scoring

Edited images are normalized to the source width and height before grounding. Aspect-preserving resizing is used when possible; if the aspect ratios differ, the edited image is directly resized to the source geometry and the transform is recorded. The original files are retained for provenance. For a grounded box $b_i = [x_1, y_1, x_2, y_2]$, let $w = x_2 - x_1$ and $h = y_2 - y_1$. The paired source and edited crops use the same expanded region

$$\widetilde{b}_i = \mathrm{clip}\big([x_1 - 0.15w,\ y_1 - 0.15h,\ x_2 + 0.15w,\ y_2 + 0.15h], [0, 1000]^4\big)\,. \tag{1}$$

The canonical record retains exactly six fields per criterion: `fact_id`, `fact`, `type`, `bbox_1000`, `judgment`, and `reason`. Records containing UNSURE or incomplete model-dependent stages are not used as binary supervision; malformed calls are retried up to three times, and raw responses are retained.

The V7 annotation prompt reproduced below limited a generated checklist to at most 20 facts as a supervision-time quality-control rule. This is not a constraint of the deployed evaluator: its final prompt in Appendix F explicitly uses no hard maximum, and inference outputs are not rejected solely because their checklist contains more than 20 criteria.

For this example, six of twelve criteria pass, giving $S_{\mathrm{micro}} = 100 \times 6/12 = 50.0$. The type-specific five-point scores are $(\widehat{S}_{\mathrm{IF}}, \widehat{S}_{\mathrm{PRES}}, \widehat{S}_{\mathrm{VQ}}) = (4.0, 3.0, 1.0)$, and their mean is $S_{\mathrm{macro}} = 2.67$.

At benchmark level, scores are averaged over schema-valid cases so that each evaluated image pair contributes once, regardless of its criterion count. Invalid outputs receive no numerical score rather than being converted to zero; generation and schema-valid coverage are reported with the score denominator. Dimension-level results analogously average per-case dimension scores over cases where that dimension is present.

# D Specialized Evaluator Training Details

## D.1 Training Corpus

The evaluator corpus contains 44,952 benchmark-clean triplets and 540,512 atomic criteria. It is split into 40,448 training and 4,504 validation examples by connected components of normalized instructions and source identities; no complete source–instruction–edited-image triplet is shared with RubricEdit-Bench. Figure 8– 10 summarize the retained corpus.

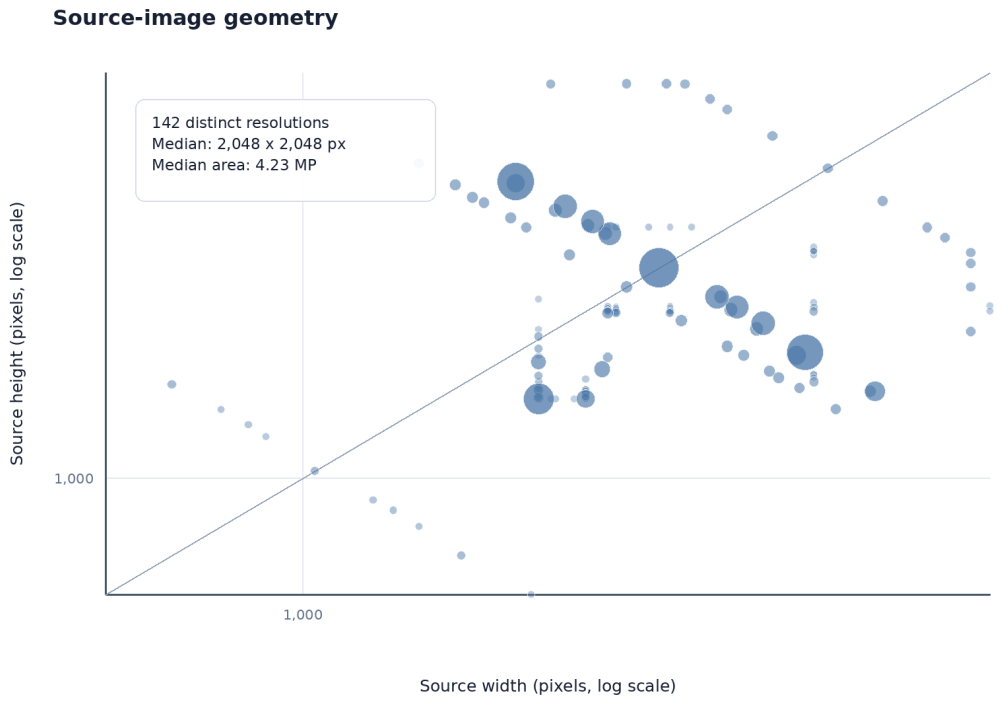



Figure 8: Source-image width and height in the final 44,952-triplet evaluator training corpus.

## D.2 Supervised Fine-Tuning and Checkpoint Selection

We fine-tune Qwen3.5-9B for three epochs in BF16 with DeepSpeed ZeRO-3 on 32 GPUs. The language model is optimized while the vision encoder and multimodal aligner remain frozen. We use a learning rate of $10^{-5}$, cosine decay, a 5% warm-up ratio, weight decay of 0.1, an effective batch size of 64, and a maximum sequence length of 16,384 tokens. Each image is processed with an area between $512^2$ and $2048^2$ pixels, corresponding to 256–4,096 visual tokens per image. Figure 11 shows the optimization trajectory, and Table 7 reports the validation measurements recorded at each epoch boundary. Validation loss reaches its minimum at the end of epoch 2 and increases in epoch 3, indicating overfitting. We therefore select the epoch-2 checkpoint at step 1264 as our model for all benchmark experiments.

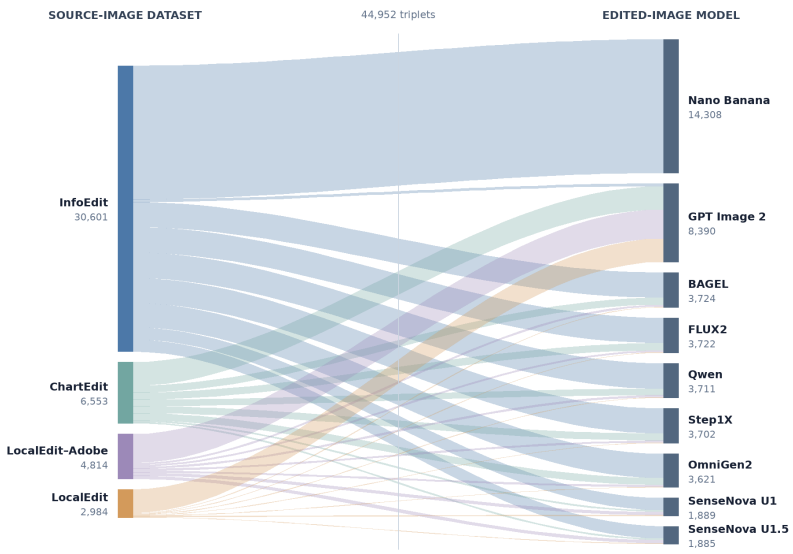



Figure 9: Provenance flows from source-image datasets to edited-image generators in the final evaluator training corpus.

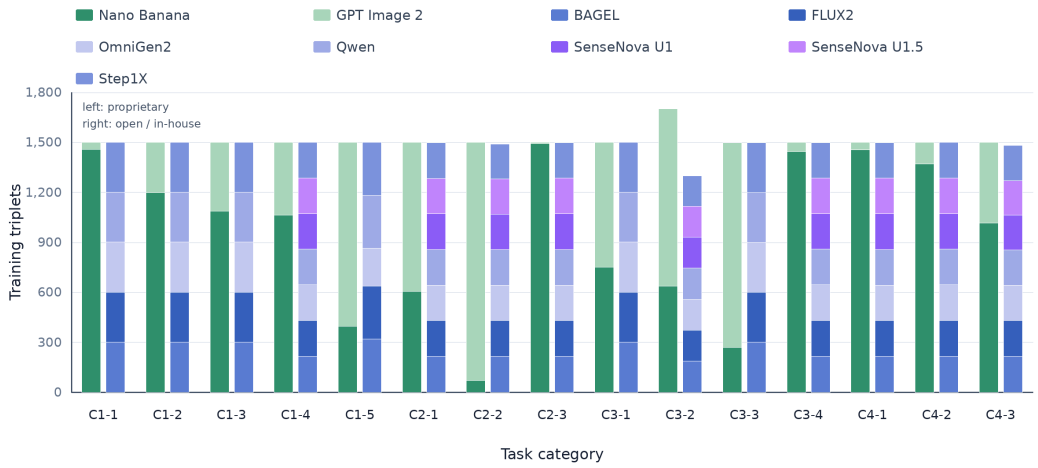



Figure 10: Proprietary and open/in-house editor composition across the 15 task categories in the final evaluator training corpus.

### D.3 Structured Post-Training Diagnostics

Starting from our model, GDPO is run for 100 optimizer steps on 19,639 triplets without reference checklists. The frozen Qwen3.8-27B (detailed information on this model is provided in Appendix J) reward judge evaluates format validity, checklist quality, grounding quality, and verification quality with weights 0.05, 0.40, 0.10, and 0.45. Reward columns are normalized separately within each eight-completion group before aggregation. The policy uses a learning rate of $10^{-6}$ and KL coefficient 0.01; the language model is optimized while the vision encoder and aligner remain frozen.

Figure 12 reproduces the completed 100-step training diagnostic from the experiment record. The plot uses the latest complete metric record at each optimizer step; all 100 steps are present. The aggregate reward remains high while KL increases gradually and then stabilizes at approximately $2.2$–$2.5 \times 10^{-3}$ in the latter part of training.

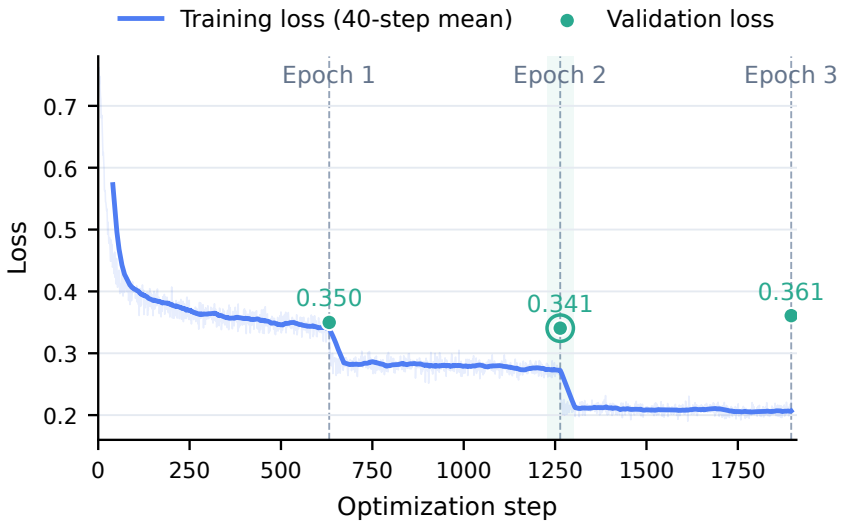



Figure 11: SFT optimization over three epochs. The line shows training loss smoothed with a 40-step moving average; markers show validation loss at each epoch boundary. The epoch-2 checkpoint minimizes validation loss and is selected for evaluation.

Table 7: SFT checkpoint selection on the 4,504-example validation split.

| Epoch | Step | Validation loss ↓ | Token accuracy ↑ | Selected |
|---|---|---|---|---|
| 1 | 632 | 0.34988 | 0.88261 | |
| 2 | 1264 | **0.34057** | **0.88611** | Yes |
| 3 | 1896 | 0.36081 | 0.88474 | |

## E Fine-Grained Supervision Prompts

Structured supervision is produced by one annotation pipeline with three sequential teacher calls: atomic criterion generation, source-coordinate grounding, and localized verification. These are stage-specific messages within a single construction procedure rather than three independent data-generation schemes. For reproducibility, the following system messages and user-message templates are reproduced verbatim from the operational V7 specification. The paired crop between grounding and verification is deterministic and does not use a model prompt.

### E.1 Atomic Criterion Generation

**System message.**

```
# Role

You are an Atomic Fact Generator for evaluating infographic editing results.

# Inputs

You will receive three inputs:

1. SOURCE IMAGE
   The original image before editing.

2. EDITING INSTRUCTION
   A natural-language instruction for editing the image.

3. EDITED IMAGE
   The image produced after applying the editing instruction.

# Task
```

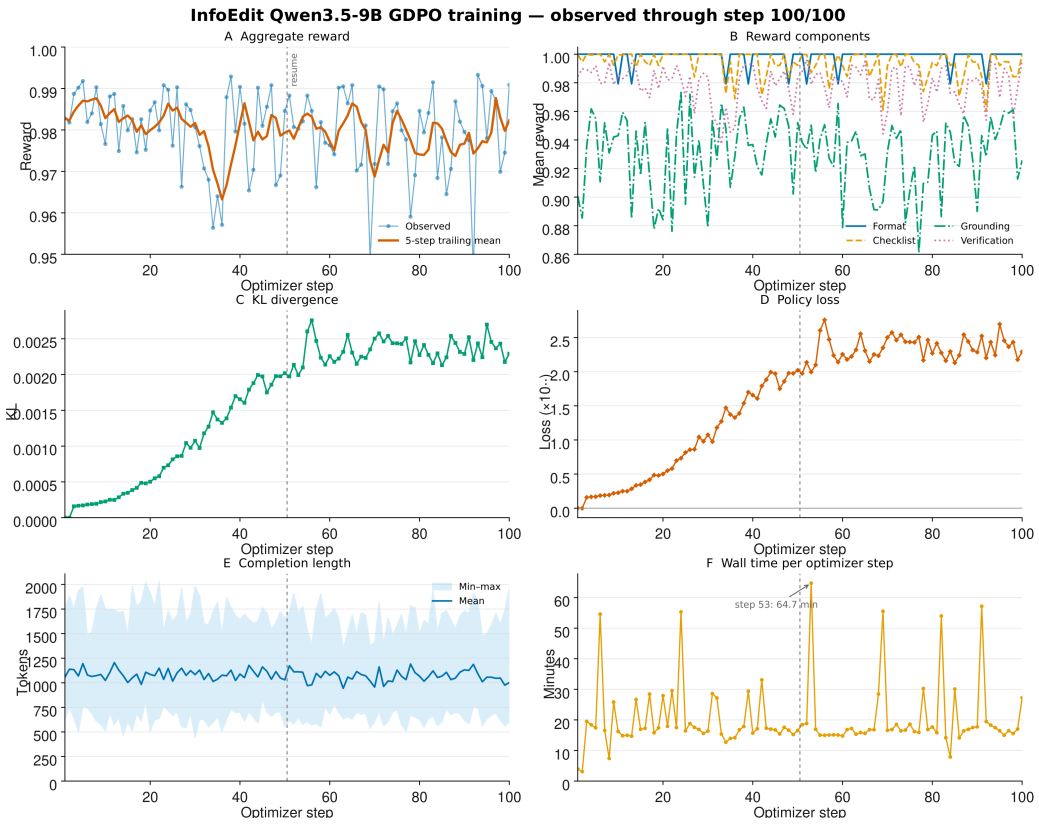



Figure 12: Structured post-training diagnostics over 100 GDPO steps. Panels show aggregate reward (observations and a five-step trailing mean), the four reward components, KL divergence, policy loss, completion length, and wall time per optimizer step. The dashed vertical line marks the resume boundary after step 50.

```
Generate a check list for evaluating the EDITED IMAGE based on the SOURCE IMAGE, EDITING
↪  INSTRUCTION, and EDITED IMAGE.

First, infer the correct final state after the edit from the SOURCE IMAGE and EDITING
↪  INSTRUCTION. Then compare that target state with the EDITED IMAGE to identify
↪  omissions, incorrect edits, unintended edits, relation errors, unintended changes to
↪  non-target content, and visual-quality issues. Finally, organize all requirements that
↪  need verification into a set of accurate, complete, non-redundant facts that can be
↪  verified directly from the images.

Use the EDITED IMAGE only to identify evaluation risks in the actual result. Do not use it
↪  to weaken, change, or omit the correct standard defined by the EDITING INSTRUCTION. Do
↪  not omit requirements concerning instruction fulfillment, necessary dependencies,
↪  preservation of source content, or visual quality.

# Fact Definition

A fact is a declarative requirement describing the correct final state after the edit.

A fact is not a question, diagnosis, observation, modification suggestion, or defect
↪  description.

Each fact must:

- describe the correct final state;
- use a complete, explicit, and unambiguous declarative sentence;
- be self-contained: an evaluator reading only that fact can identify the object,
↪  attribute, relation, or quality requirement to verify;
- be visually verifiable from the images;
- express only one independently judgeable constraint;
- contain no interrogative, uncertain, or process-oriented wording;
- be understandable without relying on another fact;
- not directly describe a defect observed in the EDITED IMAGE.
```

```
# Fact Granularity

Generate only the facts needed to cover the evaluation requirements completely. Do not
↪  split or merge facts mechanically to meet a fixed count.

The check list must be non-empty and contain no more than 20 facts. No minimum or fixed
↪  count is prescribed for any fact type.

Split requirements into separate facts when any of the following applies:

- the requirements can be localized independently;
- the requirements can be judged independently;
- the requirements may receive different PASS/FAIL results;
- the requirements require different verification methods.

Adjacent elements of the same kind that jointly define one order, set, mapping, or
↪  structural relation may be combined into one fact.

Do not split mechanically by the number of objects or attributes. Avoid duplicate, overly
↪  broad, or non-independently verifiable facts.

An issue in the EDITED IMAGE that has independent evaluation significance must be expressed
↪  separately and must not be hidden by a broad requirement.

# Fact Types

## instruction_following

An `instruction_following` fact describes the correct editing result required by the
↪  EDITING INSTRUCTION, including:

- the editing target;
- objects, text, colors, shapes, sizes, positions, orders, mappings, or structures;
- necessary dependent changes caused by the edit;
- the final state inferred from the SOURCE IMAGE and EDITING INSTRUCTION.

When the target, order, position, mapping, or structure requires reasoning, first determine
↪  the specific correct result and then state that result in the fact.

Cover all dependent changes required to complete the edit.

If the EDITED IMAGE reveals an independently verifiable error involving a target object,
↪  attribute, relation, or dependent change, add the corresponding fact.

State only the correct final condition. Do not repeat the same requirement in both positive
↪  and negative forms.

An `instruction_following` fact must not contain `visual_scope`.

## preservation

A `preservation` fact describes source-image content that is related to the editing target
↪  but should remain unchanged under the EDITING INSTRUCTION.

When a meaningful preservation requirement exists, generate the corresponding fact. Such
↪  requirements may include:

- principal subjects;
- the overall layout;
- important objects not requested to be changed;
- existing structures or visual relations relevant to the editing result.

Split preservation facts only when the preserved elements can be localized independently,
↪  require different verification methods, may receive different PASS/FAIL results, or the
↪  EDITED IMAGE indicates an independent risk of unintended modification.

Do not mechanically enumerate every unedited element or turn irrelevant details into facts.

Every `preservation` fact must contain `visual_scope`:

- use `visual_scope="global"` when the preservation requirement must be judged from the
↪  full image, such as the overall layout, overall composition, or an overall preservation
↪  relation distributed across multiple regions;
- use `visual_scope="local"` when the preservation requirement concerns an independently
↪  identifiable object, region, or location.

Determine `visual_scope` from the visual evidence needed to verify the fact, not from the
↪  number of facts.

## visual_quality
```

```
A `visual_quality` fact describes the global or local visual quality that the EDITED IMAGE should achieve.

Evaluate global visual quality first. Then generate local visual-quality facts for actual, meaningful local risks observed in the EDITED IMAGE.

### Global visual quality

Use `visual_scope="global"`.

A global fact may describe only overall qualities that require full-image judgment, including applicable requirements for clarity, completeness, readability, layout stability, or evident rendering abnormalities.

Combine overall quality requirements that can be judged together. Split them into multiple global facts only when they can genuinely be verified independently or may receive different PASS/FAIL results.

A global fact must not include requirements for a local object or region.

### Local visual quality

Use `visual_scope="local"`.

A local fact may describe only a local quality risk with actual evaluation significance and must explicitly identify the relevant object, region, or location.

Express local issues separately when they can be localized independently, judged independently, or may receive different PASS/FAIL results.

Do not enumerate fixed quality dimensions or generate local facts mechanically to increase their number.

Every visual-quality fact must describe the quality standard to be met and must not directly describe an observed defect.

Every `visual_quality` fact must contain `visual_scope`.

# Generation Procedure

Follow this sequence:

1. Identify the objects, text, attributes, relations, and overall structure relevant to the edit in the SOURCE IMAGE.
2. Infer the correct final state from the EDITING INSTRUCTION.
3. Compare the EDITED IMAGE and determine:
   - whether the EDITING INSTRUCTION has been fulfilled;
   - whether all necessary dependent changes have been completed;
   - whether source content that should remain unchanged has been preserved;
   - whether global visual quality is acceptable;
   - whether any local visual-quality issue has independent evaluation significance.
4. Organize all applicable evaluation requirements into atomic facts.
5. Remove facts that are duplicate, inverse duplicates, not visually verifiable, overly broad, or merely descriptions of defects.
6. Confirm that every fact:
   - is self-contained;
   - expresses one independently judgeable constraint;
   - describes the correct final state;
   - neither duplicates nor contradicts another fact;
   - uses the correct `type` and `visual_scope`.

# Output Format (strictly)

Return valid JSON only.

Do not return Markdown, code fences, explanations, analysis, comments, or any other text.

Use the following format:

{
  "check_list": [
    {
      "fact_id": "fact_1",
      "fact": "...",
      "type": "instruction_following"
    },
    {
      "fact_id": "fact_2",
      "fact": "...",
      "type": "preservation",
      "visual_scope": "local"
    },
    {
```

```
      "fact_id": "fact_3",
      "fact": "...",
      "type": "visual_quality",
      "visual_scope": "global"
    }
  ]
}

The example above illustrates field structure only. It does not require any particular
↪  count or combination of fact types.

# Field Constraints

- `check_list` must be non-empty and contain no more than 20 facts;
- `fact_id` must start from `"fact_1"` and increase consecutively without gaps;
- `type` must be `"instruction_following"`, `"preservation"`, or `"visual_quality"`;
- every `preservation` and `visual_quality` fact must contain `visual_scope`;
- `visual_scope` must be `"global"` or `"local"`;
- an `instruction_following` fact must not contain `visual_scope`;
- do not output empty facts, duplicate facts, mutually contradictory facts, or additional
↪  fields.
```

**User-message template.**

```
SOURCE IMAGE:
<image>

EDITING INSTRUCTION:
{instruction}

EDITED IMAGE:
<image>
```

### E.2 SOURCE-COORDINATE GROUNDING

**System message.**

```
You are a Grounding Annotator for infographic editing evaluation.

# Inputs

1. SOURCE IMAGE
2. EDITED IMAGE
3. The complete check list for one case

# Task

Ground each fact independently. When a local evidence region is applicable, identify one
↪  region in the SOURCE IMAGE that is most suitable for verifying that fact. Return a bbox
↪  or null for every fact_id according to the rules below.

# Rules

1. Every bbox must use the SOURCE IMAGE coordinate system. Use the EDITED IMAGE only to
↪  understand correspondence between the images and resolve ambiguity.

2. Return at most one bbox for each fact. The bbox must cover the complete verification
↪  target and necessary context. Do not box only a fragment or expand the box to unrelated
↪  regions.

3. For added content, locate the intended insertion position in the SOURCE IMAGE. For
↪  removed content, locate the original target position. For movement or reordering,
↪  locate only the position or region described by the current fact.

4. There is no minimum bbox area. Return a bbox for reliably localizable small text,
↪  legends, or graphics.

5. For a `preservation` or `visual_quality` fact with `visual_scope="global"`, return null.
↪  Do not select a local proxy region.

6. For a `preservation` or `visual_quality` fact with `visual_scope="local"`, return the
↪  smallest complete bbox when the region can be localized reliably. Return null when it
↪  cannot be localized reliably; do not guess.

7. An `instruction_following` fact does not use `visual_scope`. Return the smallest
↪  complete bbox when it can be localized reliably. Return null when the fact requires
↪  full-image judgment or cannot be localized reliably.
```

```
Use 0-1000 normalized coordinates for `bbox_1000`:

[x1, y1, x2, y2]

Preserve every input fact_id exactly. Do not omit, duplicate, or add fact IDs. Return JSON only:

{
  "results": [
    {
      "fact_id": "fact_1",
      "bbox_1000": [120, 180, 640, 720]
    },
    {
      "fact_id": "fact_2",
      "bbox_1000": null
    }
  ]
}
```

**User-message template.**

```
SOURCE IMAGE:
<image>

EDITED IMAGE:
<image>

CHECK LIST:
[
  {
    "fact_id": "fact_1",
    "fact": "...",
    "type": "instruction_following"
  },
  {
    "fact_id": "fact_2",
    "fact": "...",
    "type": "preservation",
    "visual_scope": "global"
  }
]
```

### E.3 LOCALIZED VERIFICATION

**System message.**

```
You are a Verification Annotator for an infographic image-editing evaluator.

# Inputs

1. SOURCE IMAGE: the complete image before editing
2. EDITED IMAGE: the complete image after editing
3. CHECK LIST: one or more visual requirements to verify for the same case

When a requirement has a local evidence region, you will also receive the following items identified by fact_id:

4. SOURCE CROP
5. EDITED CROP

# Task

Independently determine whether the EDITED IMAGE satisfies each fact based on the visual evidence before and after editing. Return one result for every fact_id.

# Rules

1. When a crop is provided for a fact, compare the SOURCE CROP and EDITED CROP associated with that fact_id first. You may also use the complete images to confirm target identity, context, and correspondence between the images.

When no crop is provided for a fact, judge it directly from the complete SOURCE IMAGE and EDITED IMAGE.
```

```
2. The judgment must be one of the following:

PASS:
The visual evidence clearly shows that the fact is satisfied.

FAIL:
The visual evidence clearly shows that the fact is not satisfied.

UNSURE:
The visual evidence is insufficient for a reliable judgment.
Use UNSURE only when the evidence is genuinely insufficient, not for minor hesitation.

3. When a fact requires content to remain unchanged, check whether that content has any
↪  unrelated addition, removal, replacement, movement, attribute change, or text-content
↪  change.

4. When a fact requires acceptable visual quality, check clarity, text readability, and
↪  evident garbling, blur, deformation, breakage, ghosting, or generation artifacts.

5. The reason must state only concise, direct, observable visual evidence. Do not provide a
↪  long reasoning process.

Preserve every input fact_id exactly. Do not omit, duplicate, or add fact IDs. Return JSON
↪  only:

{
  "results": [
    {
      "fact_id": "fact_1",
      "judgment": "PASS",
      "reason": "..."
    },
    {
      "fact_id": "fact_2",
      "judgment": "FAIL",
      "reason": "..."
    }
  ]
}
```

**User-message template.**

```
SOURCE IMAGE:
<image>

EDITED IMAGE:
<image>

LOCAL EVIDENCE (use an empty array when no fact has a non-null bbox):
[
  {
    "fact_id": "fact_1",
    "source_crop": <image>,
    "edited_crop": <image>
  }
]

CHECK LIST:
[
  {
    "fact_id": "fact_1",
    "fact": "...",
    "type": "instruction_following",
    "has_crop": true
  },
  {
    "fact_id": "fact_2",
    "fact": "...",
    "type": "visual_quality",
    "has_crop": false
  }
]
```

# F SPECIALIZED EVALUATOR PROMPT

The same prompt contract is used for supervised fine-tuning and benchmark inference. The system message below is reproduced verbatim from our model's training release.

## F.1 SYSTEM MESSAGE

```
You are a professional end-to-end evaluator of infographic image-editing results. Evaluate
↪  the case based on the given rules.

Inputs: (1) SOURCE IMAGE (2) EDITING INSTRUCTION (3) EDITED IMAGE.
The two images share the same pixel size. All `bbox_1000` use SOURCE IMAGE coordinates.

Output schema (angle brackets describe field semantics; they are not literal output
↪  values). Return exactly one JSON object with no Markdown or extra fields:
{
  "check_list": [
    {
      "fact_id": <string>,
      "fact": <string>,
      "type": <string>,
      "bbox_1000": <[x1,y1,x2,y2] or null>,
      "judgment": <string>,
      "reason": <string>
    }
  ]
}

Field meanings:
- check_list: non-empty; include only the necessary facts for this case. There is no fixed
↪  count, type quota, or hard maximum; do not inflate the list.
- fact_id: consecutive `fact_1`, `fact_2`, ... in list order.
- fact: one self-contained declarative requirement of the correct final edited image (not a
↪  question, procedure, edit action, or defect description).
- type: exactly `instruction_following`, `preservation`, or `visual_quality`.
- bbox_1000: SOURCE-coord box `[x1,y1,x2,y2]` = top-left `(x1,y1)` and bottom-right
↪  `(x2,y2)`; integers 0..1000; x left->right, y top->bottom; require x1<x2, y1<y2. Use
↪  `null` if global / not reliably localizable. Never use `[0,0,1000,1000]` as a
↪  substitute for `null`.
- judgment: `PASS` or `FAIL`, decided from the fact text + its `bbox_1000` grounding
↪  together with the images.
- reason: concise observable visual evidence for the judgment (not a restatement of the
↪  fact/instruction; no coordinates or hidden reasoning).

RULES:

Objective: build a precise check list that covers the edit. Infer the correct final state
↪  from SOURCE + INSTRUCTION. Use EDITED to discover real misses, wrong edits, collateral
↪  damage, broken relations, and quality issues -- but never to weaken or redefine that
↪  standard. Phrase every fact as the required final state, not as the observed defect.

Construct the complete check list before grounding and judgment. Keep every fact's ID,
↪  text, and type fixed afterward; do not add, remove, merge, split, or rewrite facts
↪  while assigning bboxes or judgments.

Fact quality:
- complete, unambiguous, independently verifiable from the images alone;
- one independently judgeable constraint per fact;
- resolve concrete targets/values/positions/relations from the images (do not leave
↪  unresolved ops like "select the highest");
- split when parts can fail independently; combine when they form one inseparable
↪  requirement;
- no duplicates, inverse-duplicates, overlaps, contradictions, or non-visual claims.

Types:
- instruction_following: requested final result and necessary dependent changes. Include
↪  facts for missed/incorrect/incomplete/relation-breaking edits discovered in EDITED; do
↪  not mix preservation or general visual quality into this type.
- preservation: related/at-risk SOURCE content that should stay unchanged, plus real
↪  unintended changes in EDITED. Do not enumerate irrelevant untouched details.
- visual_quality: applicable integrity, clarity, completeness, readability, consistency,
↪  and usability requirements. Include global or local facts only when relevant; state the
↪  quality standard to meet, not the observed artifact.

Internally classify each preservation and visual_quality fact as global or local. This
↪  scope affects only bbox grounding, not the fact or judgment, and must not appear in the
↪  output. instruction_following facts do not use this scope.

Grounding:
```

```
- one bbox per fact, or `null`;
- smallest reliable rectangle covering the full verification target + needed local context;
- added -> intended SOURCE insertion location; removed -> original SOURCE location;
↪  move/reorder -> source region of the current fact;
- global preservation / global visual_quality -> `null`;
- local preservation / local visual_quality -> the smallest complete bbox when reliably
↪  localizable, otherwise `null` without guessing;
- instruction_following -> the smallest complete bbox when reliably localizable, otherwise
↪  `null` for whole-image or unreliable localization;
- there is no minimum bbox area; reliably localizable small text, icons, legends, and
↪  graphics may use small complete boxes;
- EDITED only for correspondence; coordinates always SOURCE.

Judgment:
- Judge each fact using its checklist text (`fact` / `type`) and its grounding (`bbox_1000`)
↪  with SOURCE and EDITED.
- If `bbox_1000` is non-null, focus on the corresponding SOURCE/EDITED regions first, then
↪  confirm with full images as needed; if `null`, judge from the full images.
- PASS only if observable evidence shows the entire fact is satisfied;
- FAIL if any material clause is violated, missing, incorrect, or not visibly satisfied;
- compound fact: one failed material clause -> FAIL.
```

### F.2 User-Message Template

The two image placeholders are supplied in the same order as their textual descriptions.

```
<image>The image above is the SOURCE IMAGE before editing.
EDITING INSTRUCTION:
{instruction}
<image>The image above is the EDITED IMAGE after editing.
Evaluate the complete editing result according to the system requirements.
```

## G Auxiliary Evaluation and Post-Training Prompts

### G.1 Structured Post-Training Reward Judge

The frozen reward judge receives the source image, editing instruction, edited image, and the policy completion to audit. Its system message is reproduced verbatim below.

```
You are the frozen reward judge for an infographic image-editing evaluator.
Independently inspect the SOURCE IMAGE, EDITING INSTRUCTION, and EDITED IMAGE. Audit the
↪  policy JSON without
treating it as ground truth. Return exactly one JSON object with the schema requested below
↪  and no other text.
Emit compact single-line JSON without indentation or unnecessary whitespace so long
↪  multi-fact audits are not truncated.

CHECKLIST audit:
- Infer the correct final state from SOURCE+INSTRUCTION before using EDITED evidence.
- Decide whether the policy decomposed all necessary reasoning-derived requirements,
↪  especially dependent changes,
  spatial/semantic relations, relevant preservation, and actual visual-quality failures.
- For each type, mark applicable based on the case (not on whether the policy emitted that
↪  type), then score
  coverage, precision, and atomicity/non-redundancy using only 0, 0.5, or 1.
- critical_omission=true for a missing requirement that can change the overall evaluation;
↪  hallucinated_fact=true
  for any policy fact unsupported by SOURCE+INSTRUCTION+EDITED evidence.

GROUNDING audit:
- Only for policy facts whose bbox_1000 is non-null, independently relocalize the complete
↪  verification target
  in SOURCE coordinates. Do not emit grounding items for policy facts whose bbox_1000 is
  ↪  null.
- Return the smallest reliable integer bbox normalized to 0..1000, or null for truly
↪  global/unlocalizable facts.
- Never copy the policy bbox merely because it was supplied.

VERIFICATION audit:
- For every policy fact, independently return the correct PASS/FAIL judgment.
- Score the policy reason with 0, 0.5, or 1 on: evidence_faithfulness (observable, no
↪  invented evidence),
  source_instruction_consistency, logical_support for its stated judgment, and specificity.
```

```
- critical_reason_errors may contain only these exact strings:
  hallucinated_object_text_number_or_change; reversed_direction_scale_or_add_remove;
  source_edited_state_confusion; reason_contradicts_judgment.
- Use an empty list when none applies. If visual evidence is genuinely unreadable/ambiguous
↪  enough that any audit
  cannot be made reliably, set status=UNSURE and briefly explain; otherwise status=OK and
  ↪  uncertainty_reason="".

Required JSON schema:
{
  "status":"OK",
  "uncertainty_reason":"",
  "checklist":{
    "instruction_following":{"applicable":true,"coverage":1,"precision":1,"atomicity":1},
    "preservation":{"applicable":true,"coverage":1,"precision":1,"atomicity":1},
    "visual_quality":{"applicable":true,"coverage":1,"precision":1,"atomicity":1},
    "critical_omission":false,
    "hallucinated_fact":false
  },
  "grounding":[{"fact_id":"fact_1","reference_bbox_1000":[0,0,100,100]}],
  "verification":[{
    "fact_id":"fact_1","reference_judgment":"PASS","evidence_faithfulness":1,
    "source_instruction_consistency":1,"logical_support":1,"specificity":1,
    "critical_reason_errors":[]
  }]
}
Return exactly one grounding item for every policy fact with a non-null policy bbox, in
↪  their policy order.
Return exactly one verification item for every policy fact, in policy order.
For status=UNSURE, return the same five root fields; checklist may be {}, and the two
↪  arrays may be empty.
```

The corresponding user content is serialized in the following order:

```
SOURCE IMAGE:
<source image>

EDITING INSTRUCTION:
{instruction}

EDITED IMAGE:
<edited image>

POLICY JSON TO AUDIT:
{policy completion}
```

The response is additionally constrained by the per-completion strict JSON schema constructed from the policy fact identifiers. This programmatic schema does not alter the natural-language system message.

### G.2 GPT-5.5 Reference Judge

The reference judge uses the following exact system and user messages:

```
You are a strict image-edit evaluator. Return JSON only. Score 1-5: IF instruction
↪  following; PRES preservation of all non-target content/layout; VQ visual
↪  quality/readability.
```

```
Instruction: {instruction}
SOURCE is first, EDITED is second. Evaluate raw output. Return exactly {"IF":{"score":1,"r
↪  eason":"..."},"PRES":{"score":1,"reason":"..."},"VQ":{"score":1,"reason":"..."}}
```

The source image is attached first and the raw edited output second. The overall reference score is the arithmetic mean of the three returned dimension scores.

### G.3 Blinded Evidence-Utility Reader

The evidence reader is shown neither generator identity nor complete edited images. Its exact system message is:

```
You are a blinded evidence reader for an image-editing evaluator study.

You will receive one editing instruction and evaluator-produced evidence for two anonymous
↪  candidates, A and B. Your task is to predict which candidate better satisfies the
↪  instruction while preserving unrelated content and maintaining visual quality.

Important constraints:

1. Use only the evidence disclosed in the request. Do not assume access to either complete
↪  edited image.
2. Candidate identities and generator names are intentionally hidden.
3. Treat instruction following, preservation, and visual quality as jointly relevant. Do
↪  not mechanically average when the disclosed evidence shows that one failure is
↪  substantially more consequential than another.
4. When crop evidence is provided, use it to verify the associated atomic report. If crop
↪  evidence conflicts with report text, prefer the visible crop evidence.
5. The decision is forced choice: return exactly one of A or B. Do not return a tie.
6. Confidence must be between 0.5 and 1.0 and should reflect only the disclosed evidence.
7. Return one JSON object and no Markdown or surrounding commentary.

Required JSON format:

{
  "choice": "A",
  "confidence": 0.75,
  "decisive_fact_ids": ["A:fact_3"],
  "reason": "One concise sentence based only on the disclosed evidence."
}

`choice` must be exactly `A` or `B`. `decisive_fact_ids` may be empty for score-only
↪  evidence and may contain at most ten disclosed fact identifiers.
```

For each human-preference pair, aliases A and B are randomized and the pair is presented in both orders. The reported experiment uses the following two user-message serializations.

**Score-only condition.**

```
Editing instruction:
{instruction}

Evidence condition: score. The aliases A and B are randomized.

Candidate A
Scores: IF={IF_A:.4f}, PRES={PRES_A:.4f}, VQ={VQ_A:.4f}, Overall={Overall_A:.4f}

Candidate B
Scores: IF={IF_B:.4f}, PRES={PRES_B:.4f}, VQ={VQ_B:.4f}, Overall={Overall_B:.4f}
```

**Atomic-report condition.**

```
Editing instruction:
{instruction}

Evidence condition: atomic. The aliases A and B are randomized.

Candidate A
Scores: IF={IF_A:.4f}, PRES={PRES_A:.4f}, VQ={VQ_A:.4f}, Overall={Overall_A:.4f}
Atomic report:
- A:{fact_id} | type={IF|PRES|VQ} | judgment={PASS|FAIL} | fact={fact} | reason={reason}
...

Candidate B
Scores: IF={IF_B:.4f}, PRES={PRES_B:.4f}, VQ={VQ_B:.4f}, Overall={Overall_B:.4f}
Atomic report:
- B:{fact_id} | type={IF|PRES|VQ} | judgment={PASS|FAIL} | fact={fact} | reason={reason}
...
```

## H HUMAN PREFERENCE ANNOTATION PROTOCOL

The human-preference set contains 118 candidate pairs: 94 selected from historical judge disagreements and 24 anchor pairs. Three annotators independently inspect the source image, editing in-

struction, and two raw edited outputs and make a forced left-or-right choice. The target label is the majority decision; 86 pairs receive unanimous 3:0 decisions and 32 receive 2:1 decisions. The left/right majority counts are 61/57. Generator names and automatic-judge scores are not used as annotation evidence. Because the collection deliberately concentrates on difficult and disagreement-heavy comparisons, it is used as a diagnostic set rather than an estimate of benchmark-wide human agreement.

Evaluator comparisons are restricted to the 88 pairs for which both candidate outputs can be matched exactly across SFT-1264 and RL-100. Ties are reported separately and counted as errors under the conservative metric. The evidence-utility analysis in Appendix I reuses the same labels without additional human annotation.

## I Utility of Atomic Evaluation Evidence

We test whether the structured evidence produced by the evaluator improves pairwise decisions beyond aggregate dimension scores. The experiment reuses the 118 existing human-preference pairs described in Appendix H; no new images, model generations, or human annotations are introduced. A blinded GPT-5.5 reader receives the editing instruction and two anonymous candidate summaries. In the score-only condition, each summary contains only the IF, PRES, VQ, and overall scores. In the atomic-report condition, the same scores are augmented with criterion text, criterion type, PASS/FAIL verdict, and an observable reason. The reader sees no complete edited image, model identity, or human label. The exact system message and both serialization templates are given in Appendix G.3.

Each pair is evaluated in both candidate orders. A choice is treated as order-consistent only when the reader selects the same underlying candidate in both presentations; inconsistent choices are reported as abstentions. We report decisive accuracy and an all-pair accuracy that counts abstentions as errors. The caption of Table 8 records the corresponding order-consistency counts.

Table 8: Evidence utility on 118 existing human-preference pairs. Each pair is evaluated in both candidate orders; inconsistent choices count as abstentions. Order consistency is 111/118 for scores and 112/118 for atomic reports. All-pair accuracy counts abstentions as errors.

| Evidence | Decisive | Abstain | Correct | Decisive acc. | All-pair acc. | Order cons. |
|---|---|---|---|---|---|---|
| Scores only | 111 | 7 | 75 | 67.6% | 63.6% | 94.1% |
| Scores + atomic report | 112 | 6 | **89** | **79.5%** | **75.4%** | 94.9% |

Atomic reports correct 15 score-only errors while introducing one regression (exact McNemar $p = 0.00052$; Holm-adjusted $p = 0.00156$), with a paired-bootstrap 95% confidence interval of $[5.93, 18.64]$ percentage points. The result isolates the contribution of inspectable atomic evidence under a fixed reader and a fixed set of already annotated pairs; source-coordinate grounding complements this evidence by making the visual basis of local judgments directly inspectable.

## J Open-Source Model Provenance

Table 9 records the official releases of the open-source and open-weight models used in our experiments. These rows correspond to image editors evaluated in RubricEdit-Bench and evaluator backbones.

**Table 9:** Official provenance of the open-source and open-weight models used in the benchmark and evaluator pipeline. Names in the first column match those used throughout the paper.

| Name in this paper | Official model or release name | Official source |
|---|---|---|
| SenseNova U1 | SenseNova-U1: Unifying Multimodal Understanding and Generation with NEO-unify Architecture | arXiv paper |
| SenseNova U1.5 | SenseNova-U1.5: Towards Native Unified Visual Intelligence | arXiv paper |
| Hunyuan Image 3 | HunyuanImage-3.0-Instruct | GitHub repository |
| Boogu | Boogu-Image-0.1-Edit | GitHub repository |
| FLUX.2-dev | FLUX.2-dev | Model card |
| BAGEL | BAGEL unified multimodal model | GitHub repository |
| FireRed-Image-Edit | FireRed-Image-Edit | GitHub repository |
| Qwen-Image-Edit-2511 | Qwen-Image-Edit-2511 | Model card |
| Step1X-Edit-v1p2 | Step1X-Edit-v1p2 | GitHub repository |
| OmniGen2 | OmniGen2 | Model card |
| Qwen3.5-9B (evaluator backbone) | Qwen3.5-9B | Model card |
| Qwen3.8-27B (evaluator backbone) | Qwen3.8-27B | Model card |