# REFLOWTEXT: BEYOND LOCAL TEXT REPLACEMENT FOR COHERENT VISUAL CONTENT REVISION

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

## ABSTRACT

Image editing models have become effective at replacing short words and phrases, but real updates often involve long text, source-to-target length changes, and multiple jointly edited regions. Beyond accurate glyphs, models must adapt line breaks, text scale, occupied regions, and surrounding layout while preserving style and non-edited content. Existing benchmarks do not systematically isolate these challenges. We introduce **ReflowText**, an English-to-English, replacement-only benchmark with 2,002 full images and 9,197 regions across four categories and 15 subcategories. It controls three coupled difficulty axes—long-form content, multi-region replacement, and length mismatch—and combines OCR metrics with a task-specific VLM judge, **ReflowJudge**, to evaluate text fidelity, font style, layout and reflow, and background preservation. ReflowJudge reaches 79.1% agreement with human annotations. We further propose **GlyphPlan**, a plug-and-play layout-first strategy in which a fine-tuned VLM predicts line-level layouts, a deterministic renderer creates spatially aligned glyph conditions, and the image editor restores visual appearance. Across two open-weight backbones, GlyphPlan consistently improves aggregate quality, especially font style and layout. Together, ReflowText and GlyphPlan advance reliable long-form, multi-region visual text editing with coherent layout adaptation.

## 1 INTRODUCTION

Text in images often needs to be updated without regenerating the entire design. Advertisements, menus, product packaging, posters, and digital interfaces contain titles, prices, descriptions, and buttons that change over time. Such updates are rarely limited to replacing one short word with another of similar length: "SALE" may become "END-OF-SEASON SALE — UP TO 50% OFF," while several headings, paragraphs, and buttons may change together. A useful editor must therefore render the requested text accurately, preserve its visual style, and maintain a coherent design after the content changes.

Existing scene-text editing evaluations largely emphasize short text, localized regions, and near-fixed layouts. Editing Text in the Wild (SRNet) (Wu et al., 2019) evaluates word-level editing on cropped text images, while ScenePair (Zeng et al., 2024) deliberately pairs regions with similar lengths and visual properties. Recent benchmarks broaden evaluation to full-image editing, multilingual generation, visual documents, and diverse operations (Tuo et al., 2024; Gui et al., 2025; Lin et al., 2026; Zhang et al., 2026; Yi et al., 2026). However, replacement remains one operation among many and is not systematically stratified by target length, source-to-target mismatch, or the number of jointly edited regions. Existing protocols may therefore verify a local word change while bypassing the coupled rendering and layout decisions required when new content no longer fits the original design. Table 1 shows that prior benchmarks do not jointly cover long-form, length-mismatched, multi-region replacement with autonomous layout adaptation.

Figure 1 makes this blind spot concrete. The left shows short, approximately matched-length replacements for which the source layout can be reused; the ReflowText example combines long-form content, multiple jointly edited regions, and heterogeneous source-to-target length changes. Long targets amplify character-level rendering errors, multiple regions test coverage and region–target binding, and length mismatch requires adaptive spatial organization. These challenges expose garbled text, omitted or misplaced regions, overflow and collisions, abnormal text scaling, style drift,



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.

Table 1: Comparison of representative scene-text editing evaluation protocols and benchmarks.

| Benchmark | Full-image replacement | Long-form replacement | Controlled length mismatch | Multi-region joint replacement | Autonomous layout adaptation |
|---|---|---|---|---|---|
| Editing Text in the Wild (SRNet) (Wu et al., 2019) | ✗ | ✗ | ✗ | ✗ | ✗ |
| TextCtrl (ScenePair) (Zeng et al., 2024) | ✗ | ✗ | ✗ | ✗ | ✗ |
| AnyText-Benchmark (Tuo et al., 2024) | ✗ | ✗ | ✗ | ✗ | ✗ |
| TextEditBench (Gui et al., 2025) | ✓ | ✗ | ✗ | ✗ | ✗ |
| TextSculpt-Bench (Lin et al., 2026) | ✓ | ✓ | ✗ | ✓ | ✗ |
| WeEdit (Zhang et al., 2026) | ✓ | ✗ | ✗ | ✓ | ✗ |
| VDE Bench (Yi et al., 2026) | ✓ | ✓ | ✗ | ✓ | ✗ |
| **ReflowText** | ✓ | ✓ | ✓ | ✓ | ✓ |

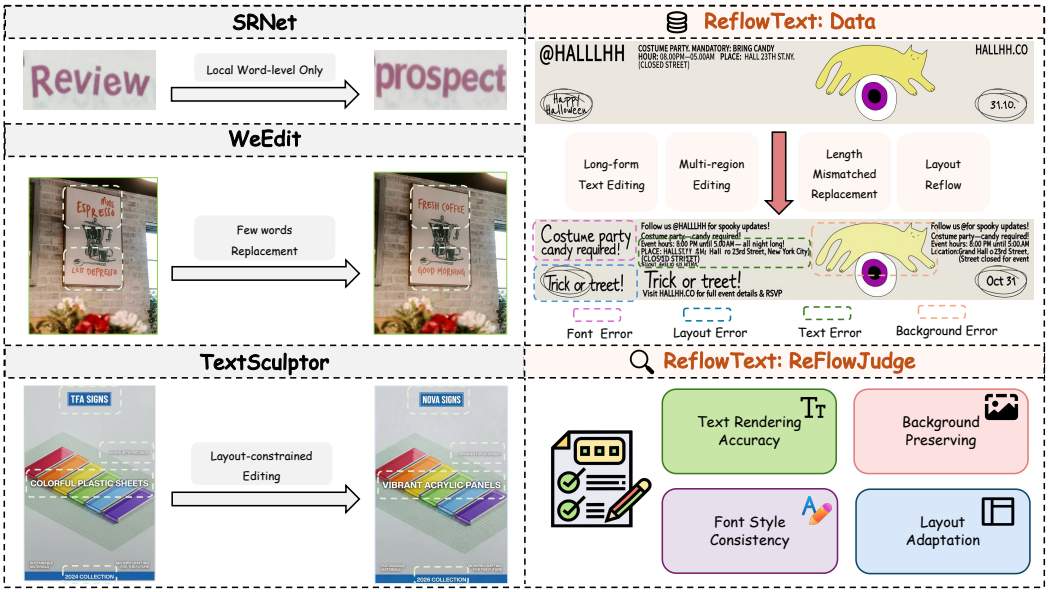


Figure 1: Motivation and scope of ReflowText. Existing benchmarks emphasize local word-level or matched-length replacement under fixed layouts, whereas ReflowText evaluates long-form, multi-region, and length-mismatched editing and separately assesses text accuracy, layout adaptation, font style, and background preservation.

and unintended background changes. Although strong editors handle short fixed-layout substitutions, reliability degrades when replacement becomes a joint content-and-layout revision—a regime prior benchmarks were not designed to isolate. We therefore ask whether an editor can render every target while preserving visual style, non-edited content, and a coherent global layout.

We introduce **ReflowText**, an English-to-English, replacement-only benchmark with 2,002 full images and 9,197 regions across four top-level categories and 15 subcategories. It controls three coupled difficulty axes: targets average 68.9 characters and span words to paragraphs; each image contains 1–8 edited regions (4.59 on average), stressing coverage and region–target binding; and targets may be shorter than, similar to, or longer than their sources, requiring adaptive line breaks, text scale, region extent, and surrounding layout. ReflowText thus tests accurate rendering and coherent reflow under compound difficulty rather than broadening the set of edit operations. ReflowText uses full-image OCR from TextPecker (Zhu et al., 2026) to measure target-text accuracy and preservation of non-edited text. Our task-specific VLM judge, **ReflowJudge**, separately evaluates style consistency, Layout & Reflow, and background preservation. Trained independently of the benchmark test set, it achieves 79.1% agreement with human annotations. Together, these complementary metrics distinguish character errors, layout failures, and unintended changes to the rest of the image.

We further propose **GlyphPlan**, a layout-first approach that explicitly separates spatial planning from visual rendering. A VLM predicts the target text line by line, including line breaks, positions,

and approximate scale; a deterministic renderer converts this plan into a spatially aligned glyph condition; and an image editor restores typography, texture, lighting, and background integration. Controlled comparisons among no glyph, naive glyph, and GlyphPlan glyph show that glyph conditioning alone is insufficient: a glyph prior conflicting with the target layout can be worse than no glyph, whereas the planned condition consistently improves text accuracy, style, and layout.

In summary, our contributions are threefold:

1. **ReflowText**, a full-image benchmark that isolates long text, length mismatch, and multi-region replacement as controlled difficulty axes across 2,002 images and 9,197 regions.
2. **ReflowJudge**, a human-validated VLM judge that, together with OCR-based metrics, enables accurate multidimensional evaluation of glyph accuracy, font-style consistency, layout and reflow quality, and background preservation.
3. **GlyphPlan**, a plug-and-play layout-planning strategy applicable to multiple image editing models, improving glyph accuracy, font-style consistency, and visual layout quality.

## 2 Related Work

### 2.1 Benchmarks for Visual Text Editing

Early scene-text editing evaluations were commonly introduced with task-specific methods. Editing Text in the Wild (Wu et al., 2019), for example, evaluates SRNet primarily on cropped word-level text images, while ScenePair, released with TextCtrl (Zeng et al., 2024), constructs matched real-image text pairs with similar lengths, styles, and backgrounds. These protocols enable controlled measurement of local text accuracy and style transfer but largely exclude full-image context, substantial content expansion, and interactions among multiple edited regions.

Later benchmarks broaden the scope of visual text generation and editing. AnyText-Benchmark (Tuo et al., 2024) emphasizes multilingual text rendering and generation quality. TextEditBench (Gui et al., 2025) evaluates diverse editing operations and reasoning requirements, while TextSculptBench (Lin et al., 2026) covers addition, replacement, removal, and mixed editing. WeEdit (Zhang et al., 2026) further expands task and language coverage, and VDE Bench (Yi et al., 2026) focuses on text-dense visual documents. Yet replacement is typically evaluated as one operation among many. In contrast, **ReflowText** holds the operation fixed and decomposes difficulty by long-form content, source-to-target length mismatch, joint multi-region editing, and autonomous layout adaptation, enabling targeted analysis of character errors, incomplete region coverage, layout conflicts, and unintended changes outside edited regions.

### 2.2 Image Editing Models

Scene-text editing has evolved from specialized pipelines to general-purpose image editors. Early methods such as SRNet (Wu et al., 2019) separate text synthesis, background reconstruction, and fusion, while TextCtrl (Zeng et al., 2024), AnyText (Tuo et al., 2024), and TextSculptor (Lin et al., 2026) introduce diffusion-based editing and stronger text or glyph conditioning. Recent open-weight models, including Qwen-Image-Edit (Wu et al., 2025), BAGEL (Deng et al., 2025), FLUX.1 Kontext [dev] (Labs et al., 2025), and SenseNova-U1.5 (Diao et al., 2026a), broaden editing to general semantic and structural changes, while proprietary systems such as GPT-Image-2 (OpenAI, 2026b), Nano Banana (Google, 2026c; 2025), and Seedream (ByteDance, 2026) further improve instruction following and short-text rendering. Open-weight editors nevertheless retain a noticeable gap on complex text replacement. **GlyphPlan** narrows this gap through an explicit, plug-and-play layout-planning stage before visual rendering.

## 3 ReflowText Benchmark and Evaluation

### 3.1 Benchmark Construction

Figure 2 summarizes the four-stage construction pipeline. We collect English-rich images from public datasets and real applications, manually remove low-quality, non-English, duplicate, or un-

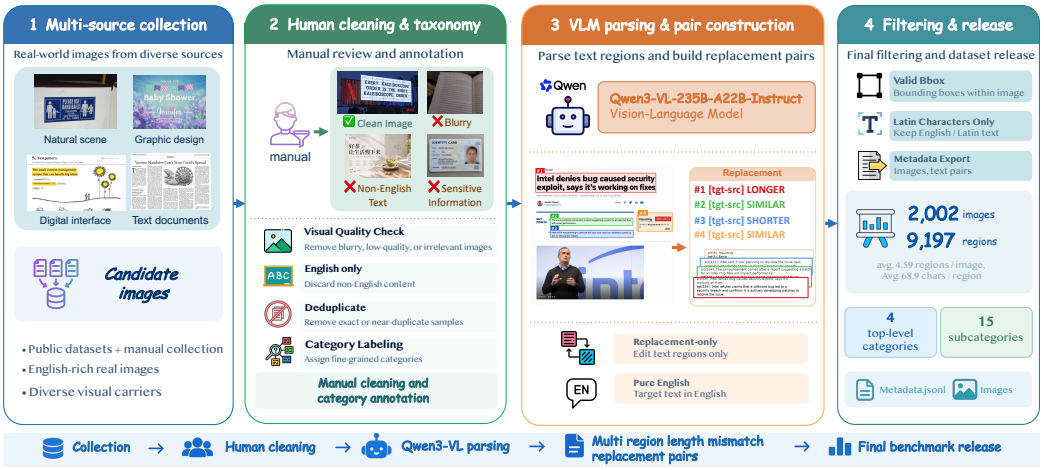


Figure 2: The ReflowText construction pipeline. Images are collected from diverse real-world sources, manually cleaned and categorized, parsed by a VLM to identify text regions and construct length-controlled replacement pairs, and finally filtered before release.

suitable samples, and assign scene categories. Qwen3-VL-235B-A22B-Instruct (Team, 2025) then parses the retained full images, localizes candidate text regions, and constructs ordered source-to-target replacement pairs. Targets are shorter than, similar to, or longer than their sources while remaining semantically consistent with the scene.

We retain only valid English-to-English replacement pairs with well-formed in-image regions. Each sample contains a full source image, scene metadata, and an ordered list of source text, target text, source-region coordinates, and length relationships. The source boxes support construction and evaluation matching but are not treated as desired output boxes, since valid longer replacements may expand beyond the original regions. During standard inference, models receive only the source image and replacement instructions—not target boxes or a target image—and must determine new line breaks, text scale, and layout autonomously.

### 3.2 Dataset Composition

The final benchmark contains 2,002 full images and 9,197 replacement regions across four top-level categories and 15 subcategories. Figure 3 shows their complementary challenges: text-dense documents stress sustained glyph accuracy, posters and interfaces require coherent layout adaptation, and natural scenes and graphic designs demand font-style preservation across diverse backgrounds and perspectives. Appendix B provides detailed category statistics.

ReflowText concentrates difficulty along three axes: long-form targets averaging 68.9 characters; multi-region editing with 1–8 regions per image (4.59 on average); and controlled source-to-target length mismatch, with 43.1% of targets longer than their sources and a median ratio of 2.08 in this subset. Together, these axes test accurate rendering, complete region coverage, and adaptive reflow under compound replacement difficulty. Holding the editing operation fixed enables controlled analysis of each axis without conflating it with operation diversity.

### 3.3 Evaluation Protocol

No single metric captures long-form replacement: an output may contain the requested words but fail in style or layout, or appear plausible while omitting text or altering unrelated content. ReflowText combines OCR metrics with visual assessment by **ReflowJudge**. We report six metrics—FG-NED, FG-Recall, BG-Text F1, Style, Layout, and Background Preservation—and use this aggregate only for compact ranking:

$$S_{\text{final}} = 100\,(0.60S_{\text{text}} + 0.25S_{\text{visual}} + 0.15S_{\text{pres}})\,, \tag{1}$$

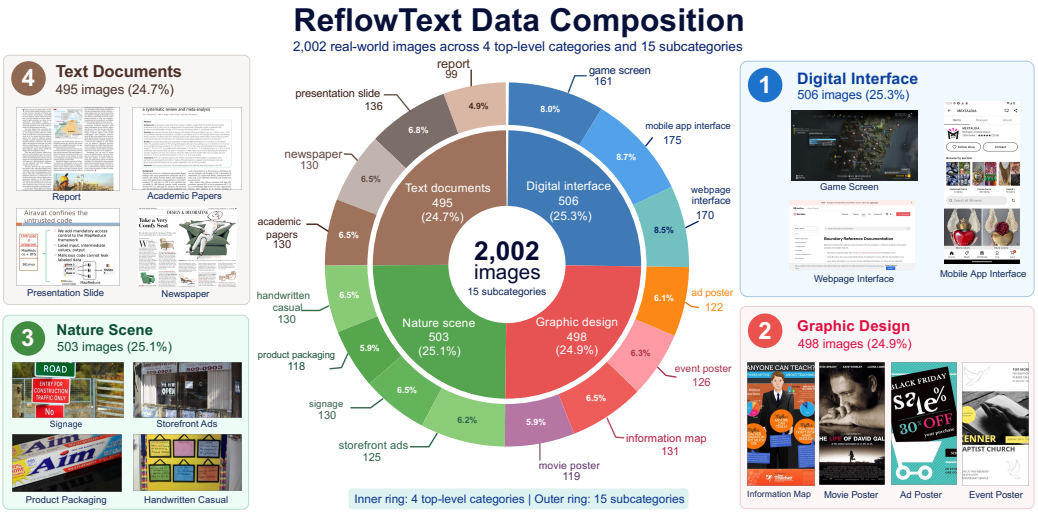


Figure 3: Composition of ReflowText across four top-level categories and 15 subcategories. The benchmark balances natural scenes, graphic designs, digital interfaces, and text-centric documents rather than concentrating on a single visual carrier.

where $S_{\text{text}} = ((1 - \text{FG-NED}) + \text{FG-Recall})/2$, $S_{\text{visual}} = (\text{Style} + \text{Layout})/2$, and $S_{\text{pres}} = (\text{BG-TextF1} + \text{Background})/2$. Thus, 60% measures target-text correctness, 25% visual quality, and 15% non-edited content preservation. The aggregate favors fidelity for ranking, but dimension-specific metrics remain primary so failures with different practical consequences remain visible.

### 3.3.1 Text Fidelity

We run TextPecker-8B-Qwen3VL (Zhu et al., 2026) on each source and edited image. **FG-NED** is the normalized edit distance between each target and its best OCR match, capturing misspellings, omissions, repetitions, and truncation; lower is better. **FG-Recall** is the fraction recovered exactly, emphasizing complete rendering over approximate transcription. **BG-Text F1** compares OCR text remaining after excluding requested replacements, measuring preservation of non-edited text. Appendix B.4 gives implementation details.

### 3.3.2 Visual Quality

OCR cannot assess visual coherence. We fine-tune **ReflowJudge**, based on Qwen3.6-VL-27B (Qwen Team, 2026a), for three binary dimensions: **Style**, covering font appearance, color, outline, and effects; **Layout**, covering position, line breaking, scale, containment, and neighboring elements; and **Background Preservation**, covering unintended changes outside requested regions. Style and Layout are judged per requested block, whereas Background Preservation is judged once over the full image. Scores are mean pass rates; training data are disjoint from the test set.

We validate ReflowJudge against 2,100 adjudicated human judgments from 50 images and 14 systems. Table 2 compares its base and combined variants, the GPT-5.5 teacher (OpenAI, 2026a), and Gemini baselines (Google, 2026b;a).

ReflowJudge reaches 79.1% agreement with human labels and Cohen's $\kappa = 0.558$, which measures agreement beyond chance, versus 65.6% for the base model. Unlike accuracy alone, $\kappa$ accounts for chance agreement when pass and fail labels are imbalanced. Split uses separate dimension-specific weights rather than one shared set, improving over the combined variant overall (79.1% vs. 78.2%) and on Layout (77.7% vs. 74.8%). We therefore use this locally deployable variant for visual evaluation; Appendix F provides further details.

Table 2: Binary-label agreement (%) between visual judgers and human annotations.

| Judger | Style | Layout | Background | Overall | $\kappa$ |
|---|---|---|---|---|---|
| GPT-5.5 teacher | **86.4** | 76.6 | **82.4** | **81.8** | **0.619** |
| ReflowJudge (split) | 80.8 | **77.7** | 78.9 | 79.1 | 0.558 |
| Qwen3.6-VL-27B (combined) | 80.9 | 74.8 | 78.9 | 78.2 | 0.551 |
| Gemini 3.1 Pro | 74.0 | 75.9 | 80.5 | 76.9 | 0.543 |
| Gemini 3.1 Flash Lite | 61.5 | 70.8 | 65.9 | 66.0 | 0.355 |
| Qwen3.6-VL-27B (base) | 64.7 | 69.8 | 62.4 | 65.6 | 0.356 |

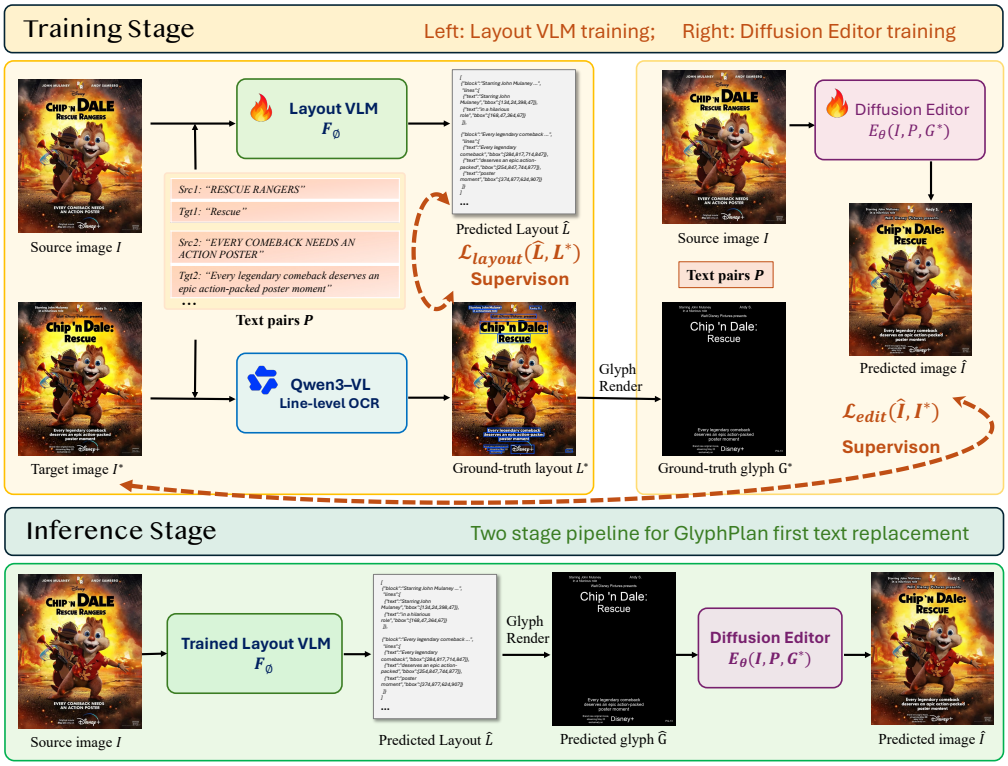


Figure 4: GlyphPlan training and inference. Target images provide line-level supervision and aligned glyph conditions during training. At inference, the VLM predicts a layout from the source image and replacement pairs, which a deterministic renderer converts into a glyph condition for editing.

## 4 GlyphPlan: Layout-First Text Replacement

ReflowText exposes a bottleneck beyond character generation: long or length-mismatched targets require new line breaks, text scales, positions, and interactions with neighboring elements. Asking one generative model to determine this layout while rendering accurate, styled text creates competing objectives. We therefore propose **GlyphPlan**, which explicitly separates layout planning from visual rendering.

### 4.1 Plan–Render–Edit Inference

Figure 4 shows GlyphPlan's *plan–render–edit* pipeline. Given source image $I$ and ordered pairs $\mathcal{P} = \{(s_i, t_i)\}_{i=1}^{K}$, a fine-tuned VLM predicts $\hat{\mathcal{L}} = F_\phi(I, \mathcal{P})$: lines with block identities, exact strings, and normalized boxes. It decides line breaks, positions, and approximate scale before generation. The plan specifies both what to render and how each target should occupy the image.

Table 3: Main results on ReflowText. All metrics are reported on a 0–100 scale. Lower FG-NED is better; all other metrics are higher-is-better. Colored columns group glyph fidelity (blue), font style (purple), layout (green), and background preservation (orange); the aggregate Final score is highlighted in yellow. Best and second-best results are shown in **bold** and underlined, respectively.

| Model | Access | **Final** ↑ | FG-NED ↓ | FG-Recall ↑ | Style ↑ | Layout ↑ | Background ↑ | BG-Text F1 ↑ |
|---|---|---|---|---|---|---|---|---|
| GPT-Image-2 | Closed-source | **72.84** | **15.80** | **64.23** | 84.85 | 86.17 | 29.09 | 63.34 |
| Qwen-Image-3-Pro | Closed-source | 70.40 | 18.28 | 59.16 | **85.56** | **87.70** | 25.14 | 61.30 |
| Nano-Banana-2 | Closed-source | 68.49 | 18.24 | 58.68 | 80.47 | 85.47 | 19.96 | 54.96 |
| Seedream-5-Pro | Closed-source | 67.78 | 18.56 | 56.16 | 73.46 | 80.68 | **32.12** | **64.30** |
| Nano-Banana-Pro | Closed-source | 66.10 | 21.56 | 54.82 | 80.64 | 83.90 | 20.90 | 53.23 |
| SenseNova-U1.5-8B-MoT | Open-source | 65.23 | 19.42 | 57.24 | 63.35 | 84.22 | 18.49 | 54.01 |
| Boogu-Image-0.1-Edit | Open-source | 62.33 | 23.06 | 52.34 | 66.11 | 82.27 | 17.89 | 48.82 |
| MAI-Image-2d6 | Closed-source | 62.04 | 22.47 | 52.38 | 67.47 | 70.52 | 25.82 | 51.80 |
| SenseNova-U1-8B-MoT | Open-source | 53.59 | 26.86 | 44.10 | 40.03 | 72.34 | 13.56 | 44.79 |
| Flux2 | Open-source | 53.16 | 36.53 | 34.72 | 75.27 | 74.51 | 22.42 | 43.98 |
| Qwen-Image-Edit-2511 | Open-source | 50.86 | 31.78 | 34.21 | 59.51 | 70.02 | 15.19 | 37.31 |
| FireRed-Image-Edit-1.1 | Open-source | 49.13 | 33.68 | 31.06 | 57.62 | 69.64 | 15.08 | 38.43 |
| LongCat-Image | Open-source | 46.60 | 38.39 | 29.96 | 56.74 | 71.15 | 9.35 | 32.54 |
| BAGEL | Open-source | 24.17 | 67.65 | 5.25 | 29.90 | 53.86 | 11.32 | 20.98 |

A deterministic renderer produces glyph condition $\hat{G} = R(\hat{\mathcal{L}})$ without introducing new wrapping. It encodes the target characters and their spatial organization but deliberately omits font identity, color, texture, and lighting. The editor generates $\hat{I} = E_\theta(I, \mathcal{P}, \hat{G})$, using the source image to recover visual appearance and integrate the rendered text. Appendix E details rendering and collision repair.

### 4.2 Training

GlyphPlan uses the line-level representation for planner and editor training. For each target image $I^*$, Qwen3-VL-235B-A22B-Instruct (Team, 2025) extracts rendered lines and boxes as ground-truth layout $\mathcal{L}^*$. These annotations capture the layout actually present in the target, including changes in line count, position, width, and scale, rather than fitting text heuristically into source regions.

The layout VLM predicts $\hat{\mathcal{L}}$ from $(I, \mathcal{P})$. During editor training, glyphs come directly from the ground-truth target layout: $G^* = R(\mathcal{L}^*)$ matches the text, line breaks, positions, and approximate scale in $I^*$ line by line. This one-to-one correspondence teaches the editor to follow glyph-defined line layouts rather than compensate for systematic spatial errors. The editor learns $(I, \mathcal{P}, G^*) \to I^*$ with its native objective, learning to follow this spatially aligned line-level condition. At inference, predicted $\hat{\mathcal{L}}$ replaces the annotation while the representation and renderer remain unchanged. Using the same renderer in both stages keeps the conditioning format identical, leaving planner prediction error as the primary train–test gap. Appendix C details the setup.

## 5 Experiments

**Evaluated systems.** We evaluate 22 configurations: six proprietary editors—GPT-Image-2, Qwen-Image-3-Pro, Nano-Banana-2, Nano-Banana-Pro, Seedream-5-Pro, and MAI-Image-2d6 (OpenAI, 2026b; Qwen Team, 2026b; Google, 2026c; 2025; ByteDance, 2026; Microsoft, 2026)—and eight open-weight editors—BAGEL, Boogu-Image-0.1-Edit, FireRed-Image-Edit-1.1, Flux2, LongCat-Image, Qwen-Image-Edit-2511, SenseNova-U1-8B-MoT, and SenseNova-U1.5-8B-MoT (Deng et al., 2025; Chen et al., 2026; Team et al., 2026; Labs, 2025; Team et al., 2025; Wu et al., 2025; Diao et al., 2026b;a)—plus zero-shot glyph guidance and six controlled supervised fine-tuning (SFT) variants on Qwen and SenseNova-U1.5.

**Evaluation protocol.** We evaluate all systems on 2,002 images. Models receive only the source image and ordered instructions, without target boxes, references, or prescribed layouts; glyph variants also receive a glyph condition. Proprietary systems use official interfaces, and open-weight models follow recommended configurations. Controlled SFT comparisons match training data, optimization budgets, and inference settings within each family, so differences isolate the conditioning strategy rather than data or compute. We report six metrics and the aggregate score on a 0–100 scale; Appendix C details the setup.

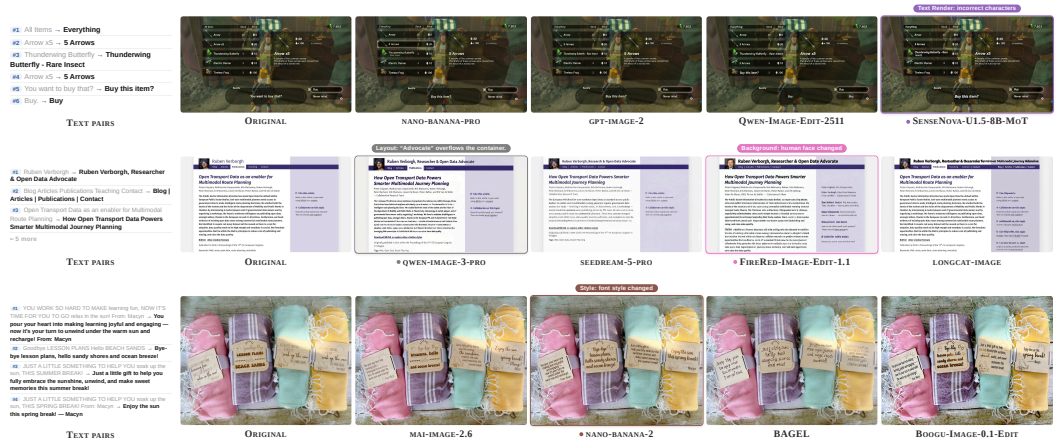


Figure 5: Qualitative comparison of representative models from Table 3. The highlighted examples illustrate complementary failure modes captured by ReflowJudge, including incorrect characters, text overflowing its visual container, unintended background changes, and inconsistent font style.

Table 4: Zero-shot glyph guidance. The better result within each backbone is shown in **bold**. Δ Final is measured relative to the corresponding no-glyph baseline.

| Base model | Condition | FG-NED ↓ | FG-Recall ↑ | BG-Text F1 ↑ | Style ↑ | Layout ↑ | Background ↑ | Final ↑ | Δ Final |
|---|---|---|---|---|---|---|---|---|---|
| Qwen-Image-Edit-2511 | No glyph | 31.78 | 34.21 | **37.31** | **59.51** | **70.02** | **15.19** | **50.86** | — |
| Qwen-Image-Edit-2511 | + GlyphPlan glyph | **26.64** | **44.49** | 26.02 | 13.64 | 43.44 | 2.56 | 44.63 | −6.23 |
| SenseNova-U1.5-8B-MoT | No glyph | 19.42 | 57.24 | 54.01 | **63.35** | **84.22** | **18.49** | **65.23** | — |
| SenseNova-U1.5-8B-MoT | + GlyphPlan glyph | **18.49** | **58.26** | **55.20** | 52.09 | 80.94 | 17.80 | 64.04 | −1.19 |

### 5.1 Main Results

Table 3 compares 14 editors. Proprietary models occupy the top five aggregate positions, but none dominates all dimensions. GPT-Image-2 leads text fidelity and overall score, Qwen-Image-3-Pro leads Style and Layout, and Seedream-5-Pro leads BG-Text F1 and Background Preservation. These complementary strengths motivate reporting of accuracy, visual coherence, and preservation.

The strongest open-weight model, SenseNova-U1.5-8B-MoT, reaches 65.23 Final and 84.22 Layout, approaching proprietary systems in spatial coherence. The 7.61-point gap between the best proprietary and open-weight systems shows that complex replacement remains challenging for local models despite competitive layout scores. The gap is largest in text fidelity, where long targets amplify character errors across edited regions. Editors differ across dimensions: Flux2 scores 75.27 on Style but only 34.72 on FG-Recall, showing that natural typography does not imply faithful rendering. Background Preservation remains low for leading systems, indicating that joint replacement often regenerates unrelated image content. Figure 5 exposes incorrect characters, style drift, overflow, and unintended changes, illustrating failures obscured by one aggregate score.

### 5.2 Zero-Shot Glyph Guidance

We test zero-shot compatibility by supplying predicted GlyphPlan conditions to pretrained Qwen-Image-Edit-2511 and SenseNova-U1.5 without training. Table 4 shows improved glyph fidelity but generally worse Style, Layout, and preservation: for Qwen, FG-NED improves from 31.78 to 26.64 and FG-Recall from 34.21 to 44.49, yet Final drops by 6.23 points; SenseNova shows the tradeoff with a 1.19-point Final decrease. Visual inspection reveals typography and layout corruption or local overwrite, with models sometimes copying the glyph image's text appearance instead of integrating it into the original style and background. The editors can extract character information from the condition but lack a learned mechanism for reconciling it with source appearance. Glyph conditions are therefore ineffective when attached directly to an unadapted editor.

Table 5: Controlled SFT ablation. Best and second-best results within each backbone are shown in **bold** and underlined, respectively. Δ Final is measured relative to SFT without glyphs.

| Base model | Condition | FG-NED ↓ | FG-Recall ↑ | BG-Text F1 ↑ | Style ↑ | Layout ↑ | Background ↑ | Final ↑ | Δ Final |
|---|---|---|---|---|---|---|---|---|---|
| Qwen-Image-Edit-2511 | SFT no glyph | 28.97 | 37.53 | 26.57 | 54.87 | 69.83 | **15.31** | 51.30 | — |
| Qwen-Image-Edit-2511 | + Naive Glyph | 40.31 | 24.50 | 27.20 | 43.70 | 65.69 | 13.62 | 41.99 | −9.31 |
| Qwen-Image-Edit-2511 | + GlyphPlan Glyph | **28.46** | **38.29** | **30.32** | **60.02** | **75.31** | 14.06 | **53.19** | +1.89 |
| SenseNova-U1.5-8B-MoT | SFT no glyph | 19.70 | **57.91** | 51.93 | 62.23 | 80.36 | 14.94 | 64.30 | — |
| SenseNova-U1.5-8B-MoT | + Naive Glyph | 22.14 | 50.89 | 52.18 | 67.35 | 73.03 | 15.00 | 61.21 | −3.09 |
| SenseNova-U1.5-8B-MoT | + GlyphPlan Glyph | **18.57** | 57.10 | **53.09** | **73.06** | **83.05** | **16.30** | **66.28** | +1.98 |

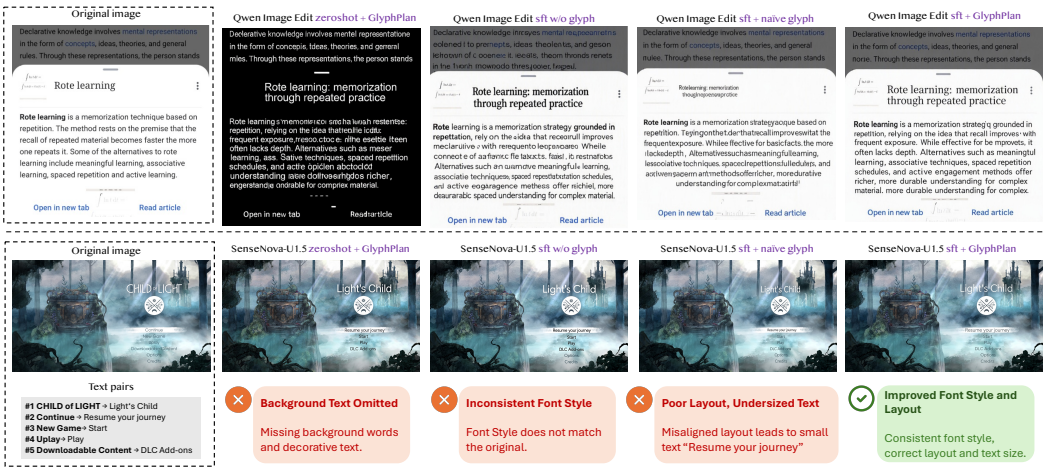


Figure 6: Qualitative comparison of glyph-conditioning strategies on SenseNova-U1.5 and Qwen Image Edit 2511. Zero-shot glyph guidance may omit preserved background text, no-glyph SFT can retain inconsistent typography, and naive glyph supervision can produce undersized or misaligned text. GlyphPlan provides a spatially aligned condition, improving font-style consistency and layout.

### 5.3 SFT Ablation: No Glyph vs. Naive Glyph vs. GlyphPlan

We ask whether glyph-conditioned training benefits from character information alone or a spatial plan aligned with the desired result. For Qwen-Image-Edit-2511 and SenseNova-U1.5-8B-MoT, we compare three matched SFT settings. *No Glyph* uses the source image and instructions. *Naive Glyph* renders each target into its source region with fixed wrapping and scaling. *GlyphPlan Glyph* uses target-aligned line supervision during training and planner predictions during inference. Within each backbone, variants share training data and optimization settings.

Table 5 and Figure 6 show that naive glyph supervision degrades both backbones because fixed wrapping and scaling conflict with target layouts, reducing Final by 9.31 points for Qwen and 3.09 for SenseNova. FG-Recall falls from 37.53 to 24.50 for Qwen and from 57.91 to 50.89 for SenseNova, confirming that inconsistent spatial supervision is harmful rather than merely ignored. GlyphPlan yields consistent gains (+1.89 and +1.98 Final), with Style improving by 5.15 and 10.83 points. Assigning line breaking, positioning, and scaling to the layout VLM offloads spatial planning from the diffusion editor, freeing capacity for font style, appearance, and image integration. The benefit therefore comes from a reliable spatial plan rather than glyph conditioning alone.

## 6 Conclusion

We study visual text replacement with long content, length mismatch, and multi-region edits, challenging text accuracy and layout coherence. ReflowText is a replacement-only benchmark of 2,002 images and 9,197 regions, evaluating text fidelity, font style, layout, and preservation. GlyphPlan separates line-level planning from visual rendering. Experiments show spatially aligned planning consistently improves quality, whereas glyph conditioning can be ineffective or harmful. ReflowText and GlyphPlan provide a foundation for improving coherent content-and-layout revision.

## AI USE STATEMENT

We used generative AI tools to support code development, figure preparation, and language editing during this project. The authors inspected all generated text, code, figures, and experimental outputs, verified factual claims and numerical results against the underlying data and evaluation records, and remain fully responsible for the manuscript, benchmark, models, and released artifacts.

## ETHICS STATEMENT

ReflowText combines images from public research datasets with a small collection of manually gathered public-facing interface and game screenshots. We screen for and exclude samples containing identifiable personal information, sensitive or unsafe content, prominent watermarks, or unclear suitability for research use. Source materials remain subject to their original licenses and terms; redistribution will follow the applicable permissions, and restricted samples will be released only as metadata or excluded.

## REPRODUCIBILITY STATEMENT

The main paper and appendices document benchmark construction, source-region annotation, OCR and visual metrics, model and inference settings, supervised fine-tuning schedules, ReflowJudge validation, GlyphPlan rendering, and all evaluation prompts. Subject to source licenses, our release will include the 2,002 benchmark images and metadata for 9,197 replacement regions, ordered replacement instructions, source-region annotations, length-policy labels, ReflowJudge and GlyphPlan prompts and training configurations, human-validation materials, evaluation records, and code for constructing instructions and computing all reported metrics.

## REFERENCES

Zoya Bylinskii, Sami Alsheikh, Spandan Madan, Adria Recasens, Kimberli Zhong, Hanspeter Pfister, Fredo Durand, and Aude Oliva. Understanding infographics through textual and visual tag prediction. *arXiv preprint arXiv:1709.09215*, 2017.

ByteDance. Seedream 5.0 pro. Online, 2026. URL https://seed.bytedance.com/en/seedream5_0_pro. Accessed: 2026-09-23.

Yuxiang Chai, Siyuan Huang, Yazhe Niu, Han Xiao, Liang Liu, Dingyu Zhang, Peng Gao, Shuai Ren, and Hongsheng Li. Amex: Android multi-annotation expo dataset for mobile gui agents, 2024. URL https://arxiv.org/abs/2407.17490.

Guoxuan Chen, Chufeng Xiao, Haoran Yang, Siyue Xie, Binxiao Huang, Ming Zhang, Cheuk Him Chau, Xinyu Fu, Yingzhao Lian, Tom SY Li, et al. Boogu-image-0.1: Boosting open-source unified multimodal understanding and generation. *arXiv preprint arXiv:2607.13125*, 2026.

Haoyu Chen, Xiaojie Xu, Wenbo Li, Jingjing Ren, Tian Ye, Songhua Liu, Ying-Cong Chen, Lei Zhu, and Xinchao Wang. Posta: A go-to framework for customized artistic poster generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28694–28704, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Chen_POSTA_A_Go-to_Framework_for_Customized_Artistic_Poster_Generation_CVPR_2025_paper.html.

Hiuyi Cheng, Peirong Zhang, Sihang Wu, Jiaxin Zhang, Qiyuan Zhu, Zecheng Xie, Jing Li, Kai Ding, and Lianwen Jin. M6doc: A large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15138–15147, June 2023.

Creative Graphic Design Lab and Shunsuke Kitada. Genposter100k dataset. Hugging Face dataset, 2025. URL: https://huggingface.co/datasets/creative-graphic-design/GenPoster100K.

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

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

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

Google. Introducing nano banana pro. Online, 2025. URL https://blog.google/innovation-and-ai/products/nano-banana-pro/. Accessed: 2026-09-23.

Google. Gemini 3.1 flash-lite: Built for intelligence at scale. Online, 2026a. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-flash-lite/. Accessed: 2026-09-23.

Google. Gemini 3.1 pro: A smarter model for your most complex tasks. Online, 2026b. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/. Accessed: 2026-09-23.

Google. Nano banana 2: Combining pro capabilities with lightning-fast speed. Online, 2026c. URL https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/. Accessed: 2026-09-23.

Rui Gui, Yang Wan, Haochen Han, Dongxing Mao, Fangming Liu, Min Li, and Alex Jinpeng Wang. Texteditbench: Evaluating reasoning-aware text editing beyond rendering. *arXiv preprint arXiv:2512.16270*, 2025.

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, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

VI Lcvenshtcin. Binary coors capable or 'correcting deletions, insertions, and reversals. In *Soviet physics-doklady*, volume 10, pp. 707–710, 1966.

Yiheng Lin, Siyu Jiao, Xiaohan Lan, Wei Zhou, Qi She, Fei Yu, Heyun Chen, Zhengwei Wang, Jinghuan Chen, Moran Li, et al. Textsculptor: Training and benchmarking scene text editing. *arXiv preprint arXiv:2605.21090*, 2026.

Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. Towards end-to-end unified scene text detection and layout analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. Icdar 2023 competition on hierarchical text detection and recognition. *arXiv preprint arXiv:2305.09750*, 2023.

Spandan Madan, Zoya Bylinskii, Matthew Tancik, Adrià Recasens, Kimberli Zhong, Sami Alsheikh, Hanspeter Pfister, Aude Oliva, and Fredo Durand. Synthetically trained icon proposals for parsing and summarizing infographics. *arXiv preprint arXiv:1807.10441*, 2018.

Microsoft. Mai image 2.6 generator: Features, access and arena rank. Online, 2026. URL https://mai-image2.com/mai-image-2-6. Accessed: 2026-09-23.

Aoxiang Ning, Minglong Xue, Yiting Wei, Mingliang Zhou, and Senming Zhong. Artistic-style text detector and a new movie-poster dataset. *Expert Systems with Applications*, 261:125544, 2025.

OpenAI. Introducing gpt-5.5. Online, 2026a. URL https://openai.com/index/introducing-gpt-5-5/. Accessed: 2026-09-23.

OpenAI. Introducing chatgpt images 2.0. Online, 2026b. URL https://openai.com/index/introducing-chatgpt-images-2-0. Accessed: 2026-09-23.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026a. URL https://qwen.ai/blog?id=qwen3.6-27b.

Qwen Team. Qwen-image-3.0: Rich content, authentic details, deep knowledge. Online, 2026b. URL https://qwen.ai/blog?id=qwen-image-3.0. Accessed: 2026-09-23.

Shohei Tanaka, Hao Wang, and Yoshitaka Ushiku. Scipostlayout: A dataset for layout analysis and layout generation of scientific posters. *arXiv preprint arXiv:2407.19787*, 2024.

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

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Super Intelligence Team, Changhao Qiao, Chao Hui, Chen Li, Cunzheng Wang, Dejia Song, Jiale Zhang, Jing Li, Qiang Xiang, Runqi Wang, et al. Firered-image-edit-1.0 technical report. *arXiv preprint arXiv:2602.13344*, 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*, volume 2024, pp. 56783–56799, 2024.

Alex Jinpeng Wang, Dongxing Mao, Jiawei Zhang, Weiming Han, Zhuobai Dong, Linjie Li, Yiqi Lin, Zhengyuan Yang, Libo Qin, Fuwei Zhang, et al. Textatlas5m: A large-scale dataset for dense text image generation. *arXiv preprint arXiv:2502.07870*, 2025.

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

Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. Webui: A dataset for enhancing visual ui understanding with web semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2023.

Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Editing text in the wild. In *Proceedings of the 27th ACM international conference on multimedia*, pp. 1500–1508, 2019.

Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. *ICCV*, 2021.

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.

Weichao Zeng, Yan Shu, Zhenhang Li, Dongbao Yang, and Yu Zhou. Textctrl: Diffusion-based scene text editing with prior guidance control. *Advances in Neural Information Processing Systems*, 37:138569–138594, 2024.

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. In *European Conference on Computer Vision*, pp. 488–507. Springer, 2026.

Hanshen Zhu, Yuliang Liu, Xuecheng Wu, An-Lan Wang, Hao Feng, Dingkang Yang, Chao Feng, Can Huang, Jingqun Tang, and Xiang Bai. Textpecker: Rewarding structural anomaly quantification for enhancing visual text rendering. *arXiv preprint arXiv:2602.20903*, 2026.

# A Additional Qualitative Results

Figures 7 and 8 provide additional qualitative results for the 14 editors in Table 3. The five examples cover product-page editing, a text-dense infographic, a game interface, natural-scene signage, and document-style content, further illustrating differences in text fidelity, font style, layout adaptation, and preservation across models.

# B ReflowText Benchmark Details

## B.1 Data Sources and Construction

ReflowText contains 2,002 full images and 9,197 replacement regions across four domains and 15 subcategories: Natural Scene, Graphic Design, Digital Interface, and Text Documents. They cover irregular surfaces, designed graphics, structured interfaces, and dense documents. Images come from public scene-text, design, interface, and document datasets, plus manually collected interface screenshots. We retain complete images rather than text crops so typography, containers, neighboring elements, and global layout remain available.

Natural-scene images are collected primarily from HierText and its TextScenesHQ subset (Long et al., 2022; 2023). Graphic-design images draw on Visually29K (Madan et al., 2018; Bylinskii et al., 2017), Movie-Poster (Ning et al., 2025), Crello (Yamaguchi, 2021), POSTA/PosterArt (Chen et al., 2025), and GenPoster100K (Creative Graphic Design Lab & Kitada, 2025). Interface images use WebUI (Wu et al., 2023), AMEX (Chai et al., 2024), and manually collected game screens. Document images use M6Doc (Cheng et al., 2023), SciPostLayout (Tanaka et al., 2024), and TextAtlas5M together with its PPT2Structured subset (Wang et al., 2025). Table 6 reports the audited per-subcategory counts.

Table 6: Audited category composition and source datasets.

| Domain | Subcategory | Images | Regions | Main source |
|---|---|---|---|---|
| Natural Scene | Signage | 130 | 560 | HierText, TextScenesHQ |
| | Storefront Ads | 125 | 591 | HierText, TextScenesHQ |
| | Product Packaging | 118 | 583 | HierText, TextScenesHQ |
| | Handwritten Casual | 130 | 524 | HierText, TextScenesHQ |
| Graphic Design | Information Map | 131 | 660 | Visually29K |
| | Movie Poster | 119 | 514 | Movie-Poster |
| | Advertisement Poster | 122 | 540 | Crello, POSTA, GenPoster100K |
| | Event Poster | 126 | 561 | Crello, POSTA/PosterArt |
| Digital Interface | Game Screen | 161 | 723 | Manually collected |
| | Webpage Interface | 170 | 844 | WebUI |
| | Mobile App Interface | 175 | 886 | AMEX |
| Text Documents | Reports | 99 | 378 | M6Doc |
| | Academic Papers | 130 | 604 | SciPostLayout |
| | Presentation Slides | 136 | 593 | TextAtlas5M, PPT2Structured |
| | Newspaper | 130 | 636 | M6Doc |
| **Total** | 15 subcategories | **2,002** | **9,197** | |

**Construction pipeline.** Candidate images undergo manual screening for legibility, spatial complexity, resolution, language, and replacement suitability. We exclude images with unreadable principal text, predominantly non-English content, or no meaningful opportunity to study reflow. Retained images are categorized before automatic parsing, so labels do not depend on the OCR/VLM pipeline.

Qwen3-VL-235B-A22B-Instruct (Team, 2025) parses each image and returns English strings with spatial locations. Using these candidates and full-image context, it selects one to eight identifiable

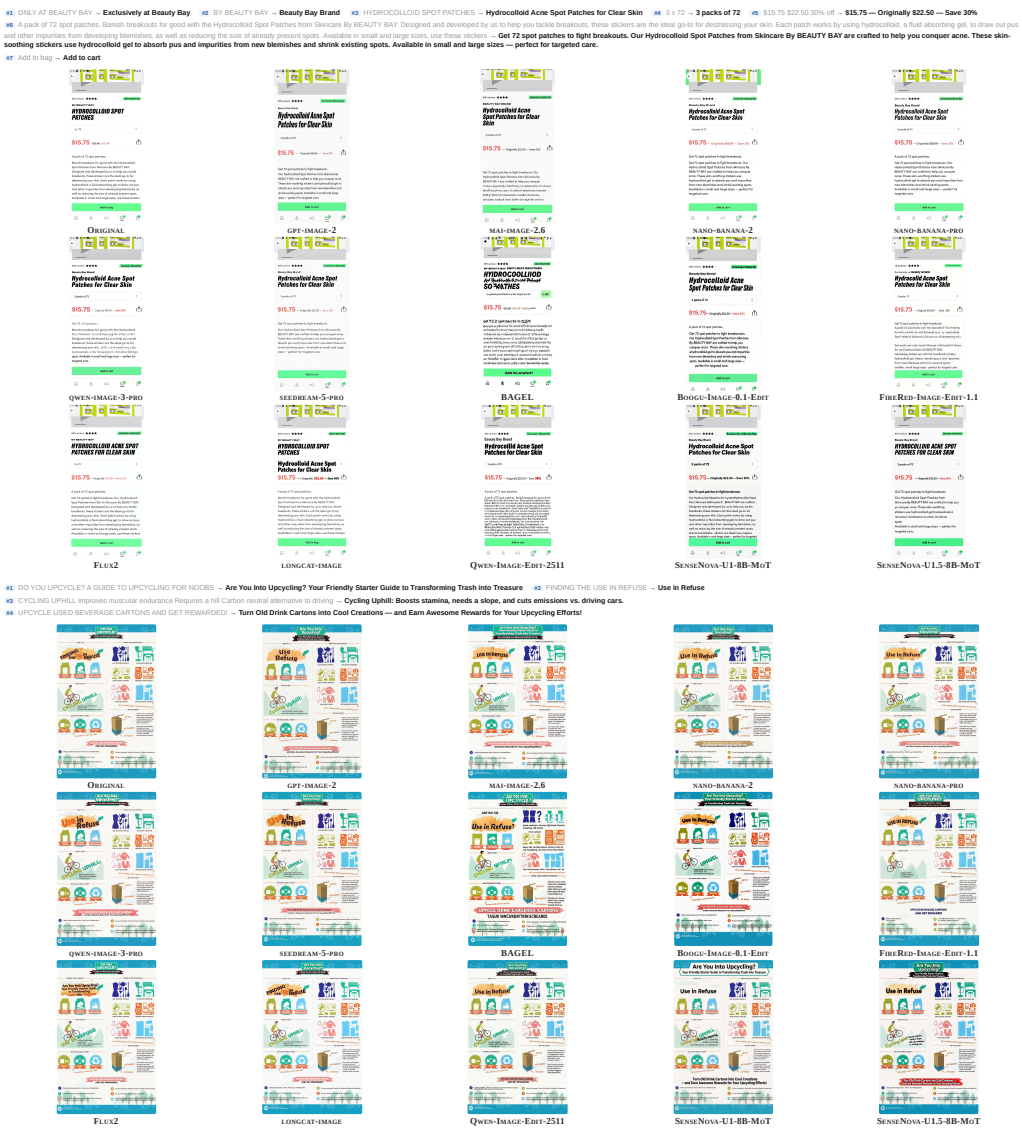


Figure 7: Additional qualitative comparison of the 14 editors in Table 3 (Part 1).

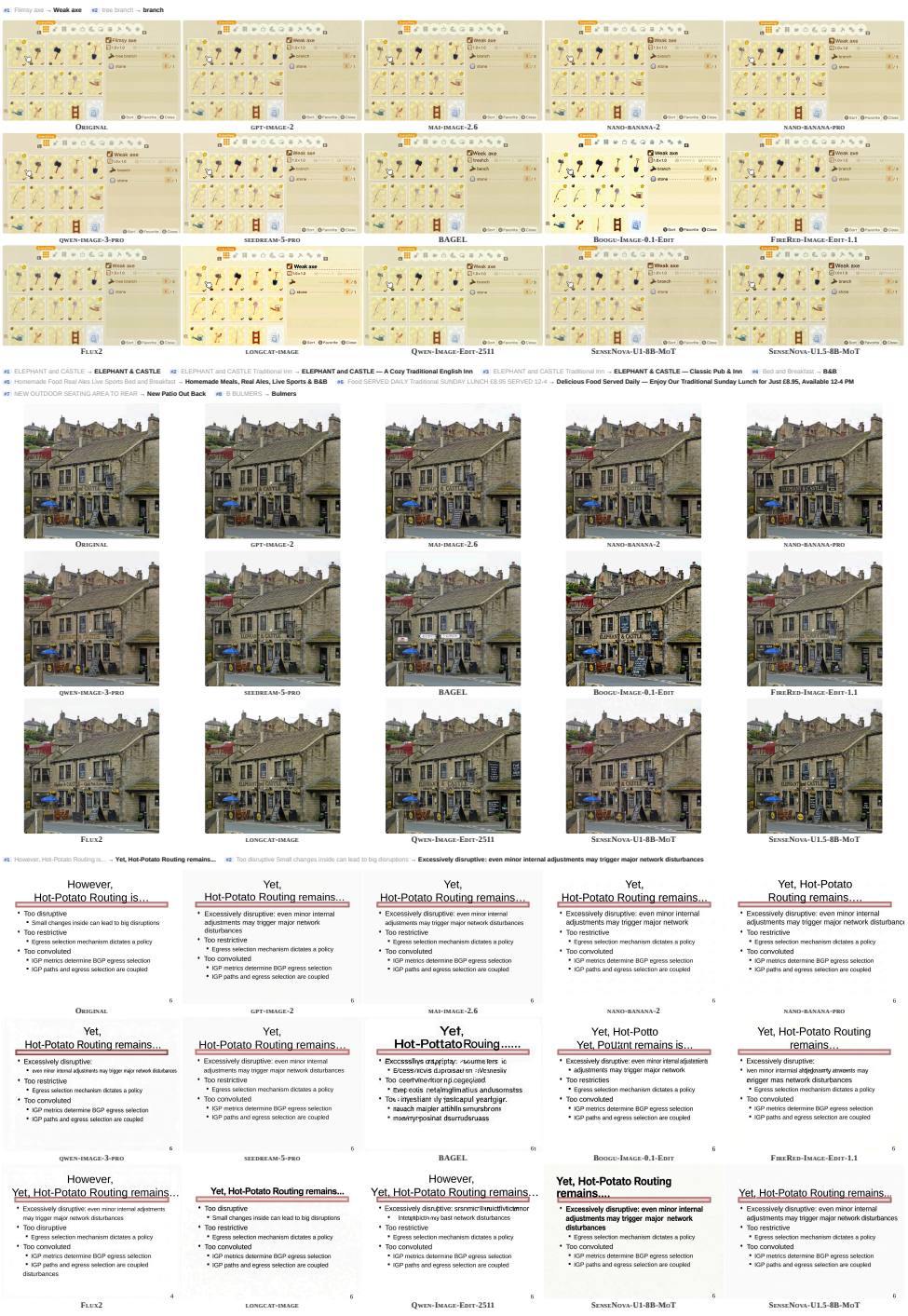


Figure 8: Additional qualitative comparison of the 14 editors in Table 3 (Part 2).

regions and generates replacements. Each region receives one length policy: SHORTER requests a concise target, SIMILAR approximately preserves source length, and LONGER expands content while remaining plausible. The prompt requires English-to-English replacement and discourages translation, deletion, or unrelated additions.

**Quality control.** We apply rule-based and manual validation after target generation. Rules reject empty strings, duplicate pairs, out-of-image or degenerate boxes, non-Latin targets, and unparseable instructions. We normalize whitespace and punctuation for length statistics while retaining the original display string. Manual inspection checks source grounding, target grammaticality and plausibility, and whether the change remains replacement. Released source boxes support construction, matching, and stratified analysis but are not target-layout constraints at inference.

### B.2 Composition and Statistics

The four domains are nearly balanced: Digital Interface contains 506 images and 2,453 regions, Natural Scene 503 and 2,258, Graphic Design 498 and 2,275, and Text Documents 495 and 2,211. Shares range from 24.7% to 25.3%, preventing one visual carrier from dominating aggregate results.

Multi-region cases dominate: the mean and median edited-region counts are 4.59 and 4, and only 43 images (2.1%) contain one region. Counts for one through eight regions are 43, 352, 345, 294, 255, 260, 220, and 233. Thus, 97.9% require editing at least two regions, enabling analysis of misses, incorrect binding, cross-region interference, and accumulated layout conflicts.

Table 7 summarizes the controlled source-to-target length policies. LONGER examples form the largest subset and have a median target-to-source ratio of 2.08. The nominal generation intervals (approximately 0.4–0.7×, 0.8–1.2×, and 1.3–1.8×) guide target construction but are not hard post-generation filters; consequently, the observed ratios may extend beyond these ranges.

Table 7: Distribution of source-to-target length policies.

| Policy | Regions | Share | Median ratio | Mean ratio |
|---|---|---|---|---|
| SHORTER | 2,409 | 26.2% | 0.67 | 0.71 |
| SIMILAR | 2,825 | 30.7% | 1.00 | 1.09 |
| LONGER | 3,963 | 43.1% | 2.08 | 2.61 |

Text lengths are strongly long-tailed. Sources average 68.2 characters, with median 23, P10 7, P90 116, and maximum 2,431. Targets average 68.9, with median 33, P10 7, P90 132, and maximum 1,883. The higher target median and P90 show that many examples involve sentence- or paragraph-level replacement rather than isolated words.

Images retain their original aspect ratios and resolutions. The median resolution is approximately $1506 \times 1600$, while the collection ranges from small interface crops to images as large as $9600 \times 6240$. This variation prevents evaluation from collapsing to a single canvas geometry and preserves the spatial constraints of webpages, posters, documents, and natural scenes.

### B.3 Annotation and Release Format

Each example contains a source image, scene labels, dimensions, and an ordered replacement-region list. Region order is preserved in instructions and metrics, making target swaps and skipped blocks traceable. A simplified record follows:

```
{
  "image": "images/0003.jpg",
  "category": "digital_interface",
  "image_size": [1280, 720],
  "coord_scale": 1000,
  "regions": [{
    "policy": "SIMILAR",
    "bbox": [x1, y1, x2, y2],
```

```
    "source_text": "Inventory",
    "target_text": "Items"
  }]
}
```

Coordinates are normalized to $[0, 1000]$ and map linearly to image dimensions. The `bbox` denotes the source region, not a required output box: valid LONGER edits may add line breaks, change scale, or expand the occupied region. The `policy` field supports stratified reporting but is not given to the editor. The release contains images, a manifest, per-image metadata, and instruction-construction scripts; internal OCR caches and construction outputs are unnecessary.

**Evaluation-time inputs.** An editor receives only the source image and ordered replacement pairs, not source or target boxes, a reference edit, category labels, or the length policy. GlyphPlan variants additionally receive the rendered condition in Appendix E. This ensures ReflowText measures autonomous localization and layout adaptation rather than execution of a supplied layout.

### B.4 OCR Metric Implementation Details

TextPecker-8B-Qwen3VL (Zhu et al., 2026) produces full-image OCR transcripts for the source and edited images. Before comparison, annotations and OCR outputs undergo the same deterministic case, punctuation, and whitespace normalization.

For each normalized target $t_i$, FG-NED selects the contiguous span $W_i$ of the edited-image OCR transcript with the minimum Levenshtein distance (Lcvenshtcin, 1966) and computes

$$\text{FG-NED}_i = \frac{d_{\text{lev}}(\text{norm}(t_i), \text{norm}(W_i))}{|\,\text{norm}(t_i)|}. \tag{2}$$

The benchmark reports the mean over replacement regions. FG-Recall uses a stricter criterion: a region passes only when its complete normalized target appears as a contiguous substring of the edited-image OCR transcript,

$$\text{FG-Recall} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\text{norm}(t_i) \text{ is a substring of } \text{norm}(T^e)]. \tag{3}$$

For BG-Text F1, we remove the requested source strings from the source OCR and the matched target strings from the edited-image OCR. The remaining text is represented as a token multiset so that changes in full-image reading order do not affect the score. If $C$ is the character mass of the matched multiset intersection and $|O|$ and $|G|$ are the remaining source and edited-text character counts, respectively, then

$$P_{\text{bg}} = \frac{C}{|G|}, \qquad R_{\text{bg}} = \frac{C}{|O|}, \qquad \text{BG-TextF1} = \frac{2P_{\text{bg}}R_{\text{bg}}}{P_{\text{bg}} + R_{\text{bg}}}. \tag{4}$$

This order-invariant comparison preserves sensitivity to modified menu items, numbers, URLs, logo text, and other text outside the requested replacement regions.

## C Training and Inference Details

### C.1 ReflowJudge

**Supervision construction.** ReflowJudge uses Qwen3.6-VL-27B (Qwen Team, 2026a) as its locally deployable base model. Its training collection contains 20,000 text-replacement outputs that are disjoint from the 2,002 ReflowText test images. Each record contains the source image, edited image, ordered source and target strings, and source-region metadata. GPT-5.5 (OpenAI, 2026a) serves as the teacher and produces evidence-based labels for four tasks: edited-region matching, Style consistency, Layout & Reflow, and Background Preservation. Decomposing every image into these tasks yields approximately 80,000 dimension-level training examples. Teacher responses are parsed into a standardized JSON representation; malformed outputs, missing block identifiers, and responses that violate the required schema are removed before training.

**Split design.** The combined baseline asks one model to return all judgments in a single long response. In contrast, split assigns matching, Style, Layout, and Background to separate task-specific adapters. The matching adapter first associates every requested block with the most plausible rendered region and records whether the source remained untouched, the target is missing, or a candidate edited region is present. The three visual adapters then receive this shared correspondence information and apply only their dimension-specific criteria. This decomposition shortens each reasoning context and prevents evidence that is decisive for one dimension—for example, a spelling error—from automatically determining an unrelated dimension such as background preservation.

Each adapter is trained with LoRA rank 64 on the attention projection layers. We use bf16, a peak learning rate of $1 \times 10^{-5}$, cosine decay, and three epochs on eight 80GB GPUs. Training one adapter takes approximately nine hours. Model selection is performed on a held-out validation split using task accuracy and valid-JSON rate; the selected adapters are merged with the base checkpoint for evaluation. The ReflowText test set is never used for teacher-label generation, checkpoint selection, or prompt tuning.

### C.2 Layout Planner and Image Editors

**Line-level planner supervision.** For each paired training example, Qwen3-VL-235B-A22B-Instruct (Team, 2025) parses the target image into visual text lines. Each line annotation contains the exact rendered substring and a normalized bounding box, and is linked to one requested replacement block. We retain an example only when concatenating the annotated lines, after whitespace normalization, recovers the complete target string. Unlike heuristic wrapping inside a source box, these annotations describe the line breaks and spatial occupancy actually present in the target image, including changes in line count, width, position, and approximate scale. The planner is trained to predict this structured target layout from the source image and ordered replacement pairs alone.

The GlyphPlan planner is a full-parameter SFT of Qwen3.6-VL-27B (Qwen Team, 2026a). It is trained for two epochs with a maximum sequence length of 16K, a vision learning rate of $2 \times 10^{-6}$, a language learning rate of $8 \times 10^{-6}$, cosine decay, and 200 warm-up steps on eight 80GB GPUs. Training takes approximately 21 hours. Inference uses FP8 weights, greedy decoding with temperature zero, and a maximum output length of 4,096 tokens. Outputs that fail JSON parsing are repaired only at the serialization level; the system does not rewrite target strings or introduce new line breaks after planner decoding.

**Controlled editor training.** Qwen-Image-Edit-2511 (Wu et al., 2025) and SenseNova-U1.5-8B-MoT (Diao et al., 2026a) are trained on the same 91,321 replacement examples. Within each backbone, No-Glyph, Naive-Glyph, and GlyphPlan-Glyph use identical source–target pairs, optimization budgets, image preprocessing, and text instructions. No-Glyph receives only the source image and replacement instruction. Naive-Glyph renders every target string into its original source box using fixed wrapping and scale-to-fit rules. GlyphPlan-Glyph instead renders the target-image line annotations during training and planner-predicted lines during inference. Therefore, the controlled comparison changes the spatial quality of the condition rather than the underlying replacement data.

We use LoRA rank 32, a learning rate of $1 \times 10^{-5}$, bf16, DeepSpeed ZeRO-2, approximately two epochs, and a maximum input area of $1024^2$ pixels. Each configuration is trained for 22,832 update steps. The Qwen adapter targets attention, MLP, and modulation modules in the DiT; the SenseNova adapter targets the generation branch of its mixture-of-transformers backbone. Training examples retain their original aspect ratio and are resized subject to the maximum-area constraint. The source image and glyph image undergo the same geometric resizing so that their coordinates remain aligned.

Table 8: Summary of the trainable components.

| Component | Base model | Update | Learning rate | Epochs |
|---|---|---|---|---|
| ReflowJudge | Qwen3.6-VL-27B | LoRA, $r = 64$ | $1 \times 10^{-5}$ | 3 |
| Layout planner | Qwen3.6-VL-27B | Full SFT | $2/8 \times 10^{-6}$ (V/L) | 2 |
| Qwen editor | Qwen-Image-Edit-2511 | LoRA, $r = 32$ | $1 \times 10^{-5}$ | $\sim$2 |
| SenseNova editor | SenseNova-U1.5-8B-MoT | LoRA, $r = 32$ | $1 \times 10^{-5}$ | $\sim$2 |

### C.3 Inference Configuration

All variants of a given backbone use identical inference settings. Table 9 lists the controlled configurations. Qwen-Image-Edit-2511 uses 40 denoising steps, `true_cfg_scale`= 4.0, `guidance_scale`= 1.0, maximum side length 1328, and seed 0. SenseNova-U1.5-8B-MoT uses 50 steps, `cfg_scale`= 4.0, `img_cfg_scale`= 1.0, `timestep_shift`= 3.0, a maximum input area of 4,194,304 pixels, and seed 42. Zero-shot and SFT glyph comparisons change only the presence and construction of the glyph condition.

Table 9: Inference settings for the controlled GlyphPlan experiments.

| Backbone | Steps | Text CFG | Image CFG | Shift | Seed | Resolution constraint |
|---|---|---|---|---|---|---|
| Qwen-Image-Edit-2511 | 40 | 4.0 | 1.0 | — | 0 | maximum side 1328 |
| SenseNova-U1.5-8B-MoT | 50 | 4.0 | 1.0 | 3.0 | 42 | maximum area 4,194,304 |

Other open-weight systems follow the configurations recommended by their official releases, including their native resolution and sampler settings. Proprietary systems are queried through their official interfaces with one output per input and without manual result selection. The same ordered instruction template is used whenever an interface permits direct text instructions. Model-version identifiers, access dates, and interface-specific parameters are recorded in the release configuration files so that hosted systems can be distinguished even if their endpoints later change.

## D Prompt Templates

### D.1 Replacement and Glyph Guidance

**Standard replacement prompt.** The following template is used for models that accept a source image and a free-form editing instruction. Replacement pairs are inserted in dataset order.

```
Replace the text "<SOURCE_1>" with the text "<TARGET_1>";
replace the text "<SOURCE_2>" with the text "<TARGET_2>";
...

All new text should preserve the original font style, color,
and background. Keep everything else in the image unchanged
except the specified text regions.
```

When a model requires a conversational wrapper, this text is placed in the user turn after the source image. We do not mention source boxes, target boxes, or the SHORTER/SIMILAR/LONGER label.

**Glyph-guided replacement prompt.** Glyph-conditioned variants receive two images in a fixed order: Image 1 is the source and Image 2 is the black-background glyph condition. The following text is appended to the standard replacement instruction:

```
Image 1 is the source image to edit. Image 2 is a glyph
reference containing white text on a black background.
```

```
For every replaced text block, match Image 2 character by
character and follow its layout, including position, scale,
line breaks, and orientation. Use Image 1 to preserve the
original typography, color, texture, perspective, lighting,
and surrounding content. Do not copy the black background
from Image 2 into the edited image.
```

The instruction format is identical for zero-shot, Naive-Glyph SFT, and GlyphPlan-Glyph SFT; only the glyph construction differs.

### D.2 Layout Planner

The planner receives the source image followed by the complete prompt below. The replacement statements are generated from the ordered region list.

```
The image above is the original image to be edited. Based on
the following source-to-target replacement pairs, predict the
rendering layout of every target text block in the edited image.

Replacement pairs:
1. Replace "<SOURCE_1>" with "<TARGET_1>".
2. Replace "<SOURCE_2>" with "<TARGET_2>".
...

For each target block:
- preserve every target character exactly;
- split the target according to its expected visual line breaks;
- return one bounding box for every rendered line;
- place the lines where they form a coherent layout with the
  original image and neighboring visual elements;
- allow the occupied region to expand or contract when the new
  content cannot be arranged naturally inside the source region;
- do not output boxes for text that is not listed above.

Use integer coordinates in [0,999], normalized by the full image
width and height. Output JSON only, with no Markdown fences and
no explanation outside the JSON.
```

The required response schema is:

```
[
  {
    "block_id": 1,
    "source_text": "<SOURCE_1>",
    "target_text": "<TARGET_1>",
    "lines": [
      {
        "text": "<TEXT_ON_VISUAL_LINE_1>",
        "bbox_2d": [x1, y1, x2, y2]
      }
    ]
  }
]
```

The concatenation of the line strings must recover the target text modulo line-breaking whitespace. Coordinates are integers in $[0, 999]$ relative to the full image width and height. A block may contain one or multiple lines, and the planner may change line count and occupied extent relative to the source.

### D.3 REFLOWJUDGE

Each ReflowJudge call receives Image 1 (source), Image 2 (edited output), and the ordered block metadata. The matching stage supplies candidate correspondences to the visual evaluators in the following common prefix:

```
You are evaluating a text-replacement result.

Image 1: the original source image.
Image 2: the edited image to evaluate.

Required text blocks, in order:
<TEXT_BLOCK_JSON>

Candidate region matches from the matching evaluator:
<MATCH_JSON>

Judge only the requested dimension. Base the decision on visible
evidence from Image 1 and Image 2. Do not reward an image merely
for looking plausible if the requested replacement is missing.
Return valid JSON only, without Markdown fences.
```

**Style evaluator.** The common prefix is followed by the complete dimension instruction below.

```
Dimension: STYLE CONSISTENCY.

For each requested block, compare the appearance of the rendered
target text in Image 2 with the source text in Image 1. Consider:
1. overall font family or visual typeface category;
2. weight, width, capitalization pattern, and italic/slant;
3. fill color and visible color treatment;
4. outline or stroke presence, thickness, and color;
5. shadow, glow, bevel, texture, or other visible decoration;
6. glyph integrity: the rendered characters must not be broken,
   melted, smeared, or visually malformed.

Do not judge exact spelling in this dimension except when the text
is missing or illegible enough that its style cannot be assessed.
If the original source text remains untouched, or if the required
target is absent, mark the block as failed because the requested
editing task was not completed.

For every block, first state concrete visual evidence.
Then give a conclusion consistent with the boolean value.

Output schema:
{
  "text_blocks": [
    {
      "block_id": 1,
      "source_text": "...",
      "target_text": "...",
      "recognized_text": "...",
      "style_consistency": {
        "analysis": "specific evidence and conclusion",
        "pass": true
      }
    }
  ]
```

```
}
```

**Layout & Reflow evaluator.**

```
Dimension: LAYOUT AND REFLOW.

For each requested block, compare its spatial organization in
Image 2 with the role of the source block in Image 1.
Consider:
1. whether the target appears in the correct semantic region;
2. alignment, orientation, reading order, and hierarchy;
3. line breaks, spacing, scale, and occupied extent;
4. containment within a sign, panel, button, label, card,
   or another text-carrying region;
5. overlap or occlusion with neighboring text and icons,
   graphics, faces, or important scene content;
6. whether nearby spacing or containers form a coherent reflow
   when the target is shorter or longer than the source.

Do not require reuse of the source box. Expansion, contraction,
new lines, and movement can pass when necessary and coherent.
Fail severe overflow, abnormally tiny text, or implausible
empty space, wrong-region binding, cross-block interference,
or destructive overlap. If the source remains untouched or the
target is missing, mark the block as failed.

Output schema:
{
  "text_blocks": [
    {
      "block_id": 1,
      "source_text": "...",
      "target_text": "...",
      "recognized_text": "...",
      "layout_consistency": {
        "analysis": "specific evidence and conclusion",
        "pass": true
      }
    }
  ]
}
```

**Background Preservation evaluator.**

```
Dimension: BACKGROUND PRESERVATION.

Evaluate the image once. Consider all content outside the listed
replacement blocks. Compare Image 2 against Image 1 for:
1. deletion or omission of objects, text, or structure;
2. addition or replacement of content not requested by the user;
3. movement, scaling, or shape changes of preserved elements;
4. distortion or garbling of preserved text, including logos,
   trademarks, formulas, URLs, numbers, and foreign languages;
5. local artifacts around edited regions, including remnants of
   source text, smearing, broken boundaries, repeated content, or
   inconsistent inpainting.

Ignore harmless pixel-level differences caused by regeneration,
compression, or resizing when content and geometry are unchanged;
```

```
legibility, or local structure. Do not penalize coherent changes
inside requested text regions that are necessary for reflow.

Output schema:
{
  "background_preservation": {
    "analysis": "specific evidence and conclusion",
    "pass": true
  }
}
```

The combined baseline uses the same definitions but requests all outputs in one response. Split applies the common prefix and one dimension block per call. Analyses are retained for auditing, while the benchmark aggregates only the binary `pass` values. The released supplementary material will include the machine-readable prompts without LaTeX line wrapping.

# E GlyphPlan Renderer Details

The deterministic renderer converts the planner's line-level JSON into an image-sized glyph condition. Let a source image have width $W$ and height $H$, and let a predicted normalized line box be $b_j = (x_1, y_1, x_2, y_2)$ in $[0, 999]^2$. We map it to pixels as

$$b_j^{\text{px}} = \left( \frac{Wx_1}{999}, \frac{Hy_1}{999}, \frac{Wx_2}{999}, \frac{Hy_2}{999} \right), \tag{5}$$

with coordinates rounded and clipped to the image boundary. Rendering occurs on a black canvas with the same $W \times H$ resolution as the source image.

## E.1 Line Rendering and Font Fitting

For every predicted line string $\ell_j$, the renderer searches a discrete set of font sizes and chooses

$$s_j^* = \max \left\{ s : w(\ell_j, s) \le w(b_j^{\text{px}}),\ h(\ell_j, s) \le h(b_j^{\text{px}}) \right\}, \tag{6}$$

where $w(\ell, s)$ and $h(\ell, s)$ are the rasterized text extents under the renderer's default font. The line is placed inside its predicted box using the planner-prescribed spatial extent. The renderer never performs additional word wrapping, hyphenation, paraphrasing, or truncation: the number of lines and all break positions are fixed by the planner output.

The glyph condition uses uniform white text and contains no estimate of the target font identity, color, stroke, texture, material, perspective, or lighting. It therefore encodes only the information intended for spatial conditioning: character content, line decomposition, position, and approximate scale. Appearance recovery remains the responsibility of the image editor, conditioned on the source image.

## E.2 Collision Repair

Planner predictions occasionally contain small overlaps between adjacent line boxes. Before rasterization, a lightweight repair stage processes lines in reading order and detects intersecting boxes. A conflicting box is translated by the minimum displacement needed to separate it from previously accepted boxes, preferring the axis consistent with the local reading direction. Coordinates are clipped after translation. Repair never modifies the line string, creates or removes a line, changes relative block order, or independently rescales a box. If no non-overlapping in-bounds translation exists, the original plan is retained rather than silently changing its semantic structure.

## E.3 Training–Inference Consistency

The same coordinate mapping, collision repair, font fitting, and rasterization code is used during training and inference. Training glyphs are

$$G^* = R(\mathcal{L}^*), \tag{7}$$

where $\mathcal{L}^*$ is extracted from the target image. Inference glyphs are

$$\hat{G} = R(\hat{\mathcal{L}}), \tag{8}$$

where $\hat{\mathcal{L}}$ is predicted from the source image and replacement pairs. Consequently, the glyph representation itself is identical across training and inference; the remaining gap is planner prediction error. This shared renderer prevents conditioning-format mismatch and allows the controlled ablation to isolate whether spatial alignment, rather than the mere presence of target characters, explains the gain.

# F Additional ReflowJudge Analysis

## F.1 Human Validation Protocol

The human study samples 50 benchmark images and collects outputs from 14 editing systems, covering proprietary, open-weight, and fine-tuned variants. For every source–output pair, annotators receive the ordered replacement list and the same dimension definitions used by ReflowJudge. Style and Layout are labeled for the requested edits, whereas Background Preservation is labeled once for the full output. Two annotators independently assign binary labels, and disagreements are resolved through adjudication after both annotators state the visual evidence supporting their decisions. This procedure yields $50 \times 14 \times 3 = 2{,}100$ binary judgment units for the reported dimension-level comparison.

The positive class is *pass*: the edited output satisfies the corresponding dimension. Precision therefore measures how often a judger's accepted cases are also accepted by humans, while recall measures how much of the human-accepted set the judger retains. Accuracy alone can conceal a permissive evaluator that labels most cases as passing, so we report precision, recall, F1, and Cohen's $\kappa$. The evaluated judgers are GPT-5.5, the task-specific combined and split Qwen3.6-VL-27B variants, the unadapted Qwen base model, Gemini 3.1 Pro, and Gemini 3.1 Flash Lite.

## F.2 Overall Agreement

Table 10 reports the full human-agreement comparison. GPT-5.5 obtains the highest aggregate agreement, while split trails it by 2.7 points and achieves the strongest Layout agreement. Relative to the unadapted Qwen base model, task-specific training improves overall agreement by 13.5 points and Cohen's $\kappa$ by 0.202.

Table 10: Agreement with human annotations (%).

| Judger | Style | Layout | Background | Overall | $\kappa$ |
|---|---|---|---|---|---|
| GPT-5.5 teacher | **86.4** | 76.6 | **82.4** | **81.8** | **0.619** |
| Qwen3.6-27B split | 80.8 | **77.7** | 78.9 | **79.1** | **0.558** |
| Qwen3.6-27B combined | 80.9 | 74.8 | 78.9 | 78.2 | 0.551 |
| Gemini 3.1 Pro | 74.0 | 75.9 | 80.5 | 76.9 | 0.543 |
| Gemini 3.1 Flash Lite | 61.5 | 70.8 | 65.9 | 66.0 | 0.355 |
| Qwen3.6-27B base | 64.7 | 69.8 | 62.4 | 65.6 | 0.356 |

## F.3 Dimension-wise Behavior

Table 11 reports the complete dimension-wise precision/recall/F1 results. Task-specific training substantially reduces the permissive behavior of the base model. For Style, the base model and both Gemini variants attain high recall but low precision, meaning that they accept many edits that humans reject for inconsistent typography or malformed glyphs. Split raises Style precision from 49 to 68 relative to the base model while retaining 81 recall.

Table 11: Dimension-wise precision/recall/F1 (%). Each entry is P/R/F1, with *pass* as the positive class.

| Judger | Style | Layout | Background |
|---|---|---|---|
| GPT-5.5 teacher | 77/85/81 | 77/81/79 | 71/66/68 |
| Qwen3.6-27B split | 68/81/74 | 80/78/79 | 68/51/58 |
| Qwen3.6-27B combined | 68/82/74 | 77/76/76 | 61/77/68 |
| Gemini 3.1 Pro | 58/88/70 | 72/90/80 | 61/87/72 |
| Gemini 3.1 Flash Lite | 46/88/61 | 70/80/75 | 46/95/62 |
| Qwen3.6-27B base | 49/88/63 | 67/89/76 | 43/95/59 |

Layout shows the clearest benefit from dimension specialization. Split increases precision from 77 to 80 relative to combined while maintaining similar recall, and its 77.7% agreement is the highest among all evaluated judgers. This dimension requires joint reasoning over text blocks, carriers, and neighboring elements; isolating it from the longer combined prompt appears to reduce false acceptance of collisions, wrong-region bindings, and incoherent reflow.

Background Preservation exhibits a different tradeoff. The base model and Gemini variants are highly permissive, reaching 95 recall but low precision. Split is more conservative: its precision rises to 68, but recall falls to 51. Manual inspection indicates that many of its false negatives arise from subtle regeneration differences near edited regions that the evaluator interprets as structural damage. Thus, the remaining gap to GPT-5.5 is concentrated more in calibration of background strictness than in a uniform loss across all visual dimensions.

### F.4 Pairwise Agreement and Practical Tradeoff

Table 12: Pairwise agreement between representative judgers.

| Judger pair | Agreement |
|---|---|
| GPT-5.5 × Qwen combined | 77.4% |
| Qwen combined × Qwen split | 75.0% |
| GPT-5.5 × Qwen split | 71.8% |
| GPT-5.5 × Gemini 3.1 Pro | 70.8% |
| Qwen base × Gemini Flash Lite | 72.7% |
| Qwen split × Qwen base | 63.0% |

The controlled progression is

$$\text{Base: } 65.6\%,\ \kappa = 0.356 \rightarrow \text{Combined: } 78.2\%,\ \kappa = 0.551 \rightarrow \text{Split: } 79.1\%,\ \kappa = 0.558. \tag{9}$$

Most of the gain comes from task-specific distillation; dimension specialization adds a smaller overall improvement but a substantial $+2.9$ points on Layout. Combined remains behaviorally closest to its GPT-5.5 teacher, whereas split agrees with the unadapted Qwen base model on only 63.0% of decisions, confirming that adaptation changes the evaluator rather than merely reformatting its outputs.

GPT-5.5 is the strongest judge in aggregate, but using a proprietary teacher for every benchmark output is expensive and complicates reproducibility. Split is locally deployable, is only 2.7 agreement points behind the teacher overall, and is stronger on Layout. We therefore use it as the default visual evaluator and report all dimensions separately so that its conservative Background behavior remains visible rather than being hidden inside the aggregate score.