INFOEDIT: Probing Global Layout Reasoning in Infographic Editing

Cheng Yang[1,*] **Chufan Shi**[2,*] **Huijuan Wang**[2,*] **Bo Shui**[3]
Yaokang Wu[4] **Muzi Tao**[2] **Yibo Yan**[2] **Xuezhe Ma**[2] **Taylor Berg-Kirkpatrick**[1]
[1]University of California San Diego [2]University of Southern California
[3]University of Illinois Urbana-Champaign [4]Carnegie Mellon University
chy085@ucsd.edu {chufansh,huijuanw}@usc.edu

Abstract

Multimodal foundation models edit natural photographs at production quality, yet the same models struggle with structured visual content such as infographics. Unlike photographs, infographics encode information through logical relations; editing one element often requires surrounding elements to be adapted. We refer to this global layout reasoning capability as *reflow*. Existing image-editing benchmarks neither provide a dedicated setting for structured visual content nor evaluate the reflow capability. We introduce INFOEDIT, a novel benchmark of 1,000 infographics across eight logical-relation families, paired with 4,000 editing instructions across four editing tasks, and a reflow-aware evaluation protocol. Across eight frontier editors, only GPT-Image-2 clears 60% average success rate; most models fall below 7%, and no editor exceeds 36% on the SWAP-BLOCK task even with perfect target localization. We further show that code-level editing can match the strongest pixel-level editor, revealing complementary strengths across tasks. INFOEDIT identifies reflow as a central challenge in structured visual content editing and provides a diagnostic benchmark to facilitate future progress.

1 Introduction

Multimodal foundation models such as GPT-Image-2 (OpenAI, 2026) and Nanobanana-Pro (Google DeepMind, 2025a) have pushed natural-image editing towards production quality. They can faithfully execute instructions such as *"remove the tree"*, producing edits that are visually coherent and semantically aligned with user intent (Huang et al., 2025a). Such instructions are often local in effect: they modify a specified target while keeping the surrounding scene consistent with the source.

Meanwhile, these same models are increasingly capable of generating structured visual content, such as infographics (Nallatamby et al., 2025; Google Labs, 2026; Zhu et al., 2026a,b). Editing such content, however, remains underexplored. Fig. 1 sketches a typical workflow: the user asks the model to insert a missing step into an infographic. The instruction names one element, but correct execution touches many: downstream nodes shift and connectors reroute. We call this *reflow*, a form of global layout reasoning that requires the model to automatically adapt the mentioned element and other implicitly affected elements (e.g., re-position and re-size), so the resulting image remains balanced, legible and logically valid. Existing image-editing benchmarks (Brooks et al., 2023; Ye et al., 2025; Pan et al., 2025; Yu et al., 2025) focus on natural-image editing; they leave infographic editing underexplored and do not evaluate the reflow capability it requires.

Therefore, we present INFOEDIT (Sec. 2), a novel benchmark for editing infographics. Such infographics cover eight logical-relation families: List, Process, Cycle, Hierarchy, Relationship, Matrix, Pyramid and Picture. Each family imposes its own reflow contract, such as re-routing arrows in a Process, or reparenting in a Hierarchy. INFOEDIT contains 1,000 human-curated infographics spanning these families on diverse real-world topics, each paired with editable source files in HTML (800) or PowerPoint (200). For each infographic, we annotate four editing tasks at increasing perturbation scope, from text-level changes to canvas-level restructurings: EXPAND-TEXT, INSERT-ELEMENT, SWAP-BLOCK, and RESHAPE-CANVAS, yielding 4,000 instruction–infographic pairs in total. Together, these infographics and instructions make genuine reflow a prerequisite.

Furthermore, we develop a reflow-aware evaluation protocol (Sec. 3) that grades each task separately, decomposing every verdict into two binary checks: edit compliance, the named change is executed on its target; and content preservation, every

[*] Equal contribution. Project Page: https://infoedit.github.io

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 57130–57168
October 24-29, 2026 ©2026 Association for Computational Linguistics

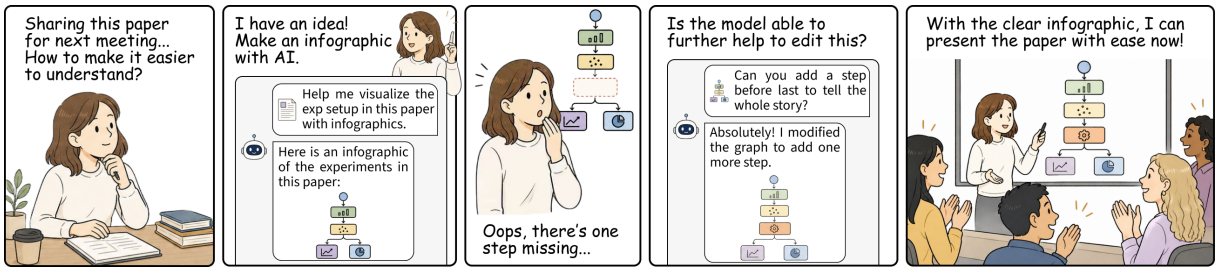


Figure 1: A typical "vibe-design" workflow with structured visual content. A user asks an image editor to add a missing step to an infographic. The instruction names one element, but the surrounding nodes and connectors must all reflow for the result to be usable. INFOEDIT measures whether current editors can perform this reflow reliably.

other element survives intact. We report success only when both hold. The decomposition matters because failures on such structured visual content include not only wrong target edits but also correct edits that silently damage unmentioned elements.

We benchmark eight frontier proprietary and open-source models on INFOEDIT (Sec. 4). Current models remain far from solving infographic editing that requires reflow. Only GPT-Image-2 clears 60% average success rate; the next two models reach 38–42%, and the remaining five, including all three open-weight baselines, fall below 7%. Our evaluation shows that reflow failures are task-specific, e.g., SWAP-BLOCK is bottlenecked by edit compliance, capping at 36.5% across all eight. Open-weight models lag especially far behind, with 0% success rate on INSERT-ELEMENT, SWAP-BLOCK, and RESHAPE-CANVAS.

Further diagnostics reveal where the bottlenecks arise (Sec. 5). Code-level editing with editable source and visual grounding reaches 61.6% average success rate, just matching GPT-Image-2 at 62.0%, but the two pathways show complementary strengths across tasks. Interactive selection hints improve SWAP-BLOCK success by only 3.2–7.4 points, with no model exceeding 36% even under perfect target localization, indicating that the bottleneck is the reflow operation itself rather than merely target localization. Difficulty and error analyses show that reflow is multi-faceted, spanning correct placement, neighbor-aware adjustment, and global style and structure preservation.

To sum up, we envision INFOEDIT as a novel comprehensive benchmark suite for guiding research on infographic editing. By providing editable-source infographics, multi-scale editing tasks, and a reflow-aware evaluation framework, INFOEDIT not only evaluates how well foundation models execute the requested edit while preserving non-target content via reflow, but also offers diagnostic insights into their capabilities and failure modes when editing structured visual artifacts.

2 INFOEDIT Benchmark

2.1 Task Definition

INFOEDIT assesses an image editor's ability to perform reflow editing on structured visual content. Given an original infographic X and a natural-language editing instruction I that names a structural change, the model f must produce

$$X' = f(X, I) \quad \text{s.t.} \quad \mathcal{C}(X', X, I) \wedge \mathcal{P}(X', X, I).$$

$\mathcal{C}$ is the *compliance* predicate: the change named by I is faithfully executed in X'. $\mathcal{P}$ is the *preservation* predicate: every element of X not mentioned by I remains present in X', free to reflow to accommodate the change. We provide representative data examples of INFOEDIT in Appx. A.3.

2.2 Infographics Construction

Design Principles. Each infographic in INFOEDIT is designed around three principles that together support reliable evaluation. They are (i) topically realistic, drawn from real-world infographic domains, so the benchmark reflects practical use cases. They are (ii) aesthetically credible, with coherent palettes, balanced layouts, and typography on par with production work. Crucially, they are (iii) logically structured: each infographic conveys information through visual logical-relation components.

Logical-Relation Taxonomy. Structured visual content conveys information through logical relations, which INFOEDIT organizes into eight families (Fig. 2, left): List, Process, Cycle, Hierarchy, Relationship, Matrix, Pyramid, and Picture. These eight families cover common infographic structures used in everyday design tools (Microsoft); each encodes a distinct semantic relation and imposes reflow contracts that any valid edit must satisfy, such as reparenting in a Hierarchy.

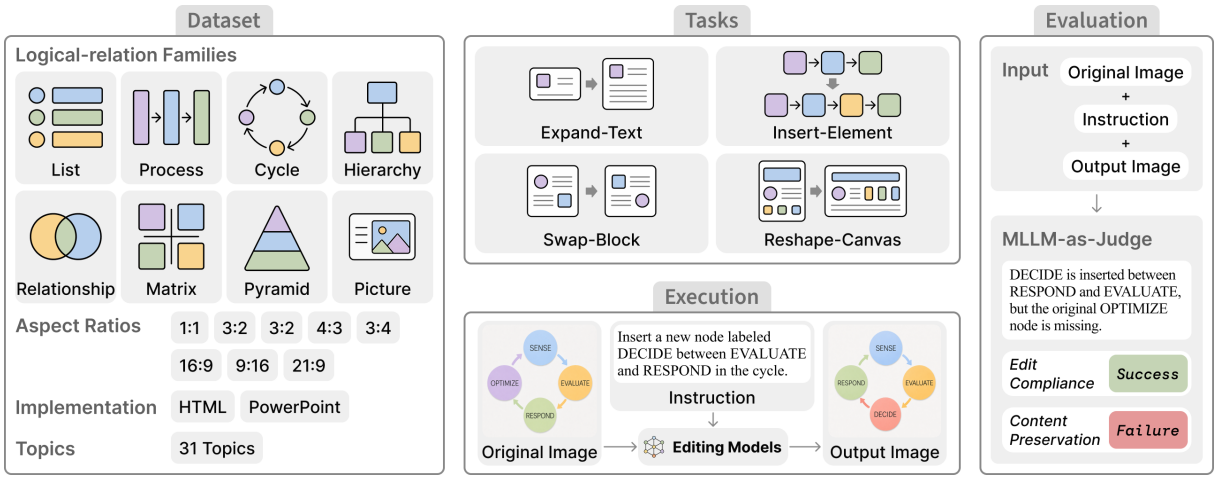


Figure 2: Overview of INFOEDIT. We curate 1,000 infographics covering eight logical-relation families (left), annotate four editing tasks for each infographic (middle), and evaluate model outputs on two binary aspects, edit compliance and content preservation (right). Data examples are provided in Appx. A.3.

Annotation Pipeline. The annotation pipeline runs in five stages and yields 1,000 infographics, each implemented with editable, structurally addressable source files (800 HTML and 200 PowerPoint). We summarize the pipeline below; full implementation details are provided in Appx. A.1.

- *Stage 1: Topic sourcing.* We define 31 topic categories where infographics are commonly produced (e.g., companies and athletes), and prompt Claude-Opus-4.7, Gemini-3.1-Pro, and GPT-5.4 to iteratively enumerate 200 concrete topics per category, yielding 6,200 candidate topics.
- *Stage 2: Visual reference collection.* For each candidate topic, we generate a reference image using Nanobanana-Pro and GPT-Image-1.5. These references are used only as color palette, not as content to reproduce. Expert annotators then select 1,000 references with aesthetically coherent and topic-faithful content.
- *Stage 3: Logical template construction.* We build a library of 79 templates instantiating the eight logical relation families. Each template has two implementations: an HTML version exposing nodes, edges, and containers as addressable DOM elements, and a PowerPoint version using named vector shapes and connectors.
- *Stage 4: Infographic composition.* Each infographic is composed by transferring the color palette of a reference image onto logical-relation templates. The HTML track is model-assisted: an annotator selects several templates and assembles them together. Claude-Opus-4.7 then performs two passes conditioned on the reference image and selected templates: a styling pass that adapts the CSS to match the reference's color palette, and a content pass that fills textual placeholders with topic-relevant text. The annotator then post-edits the result. In contrast, the PowerPoint track is authored fully manually by annotators, who select and arrange templates and adjust the color palette to match the reference image.
- *Stage 5: Quality control.* Every infographic is reviewed carefully by a second annotator against three criteria: (i) structural integrity, no overflow or broken connectors, and complete logical-relation instances; (ii) stylistic faithfulness, palette matching the reference; and (iii) content completeness, all textual fields and labels are topic-relevant and role-compatible.

Diversity and Coverage. Fig. 3 shows that INFOEDIT spans the diversity of real-world infographics along three axes: *structural complexity*, *family coverage*, and *canvas shape*.

- *Structural complexity* (Fig. 3a). Each infographic contains four blocks on average, drawn from one or more logical-relation families. Only 19% realize a single family; the remaining 81% compose two or more on the canvas, with three families the modal case and instances reaching as many as six. The benchmark therefore exercises both within-family and across-family reflow.
- *Family coverage* (Fig. 3b). All eight families appear, with a non-uniform distribution: List, Matrix, and Process are more frequent, while Hierarchy appears less often. These family annotations are non-mutually exclusive: for example, the 88% reported for List means that 88% of infographics contain a List structure, not that 88% of the dataset belongs exclusively to a List class.

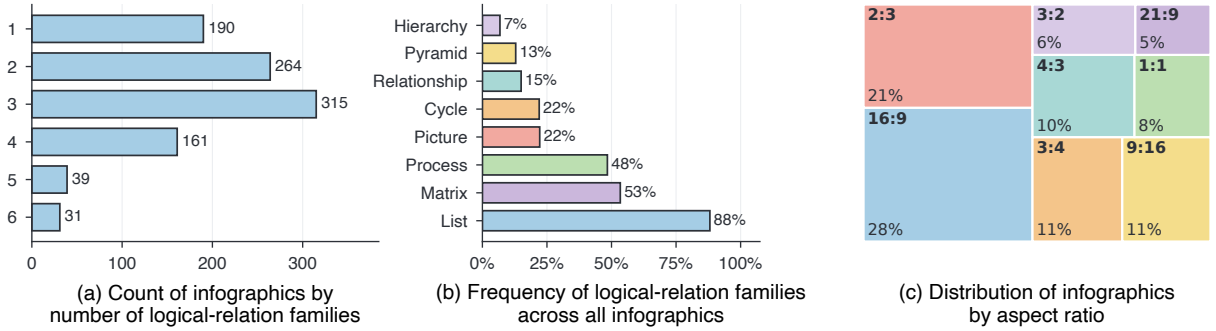


Figure 3: Distributional summary of INFOEDIT. (a) Distinct logical-relation families per infographic; 19% contain one, 81% contain two to six. (b) Fraction of infographics containing each family; an infographic can contain multiple families. (c) Source aspect-ratio distribution over eight ratios.

List-based organization is a fundamental structure that commonly co-occurs with other relations in infographics. We keep this natural skew rather than enforcing uniform coverage while still including all relation families; per-family results in Appx. F show that the main model comparison holds across all eight families.

- *Canvas shape* (Fig. 3c). Source aspect ratios span landscape (21:9, 16:9, 4:3, 3:2), portrait (9:16, 3:4, 2:3), and square (1:1) formats, allowing evaluation across diverse practical canvas shapes.

2.3 Editing Tasks

As illustrated in Fig. 2 (middle), INFOEDIT defines four editing tasks, ordered by the scope of structural perturbation they demand, from text-level changes to canvas-level restructurings:

- EXPAND-TEXT lengthens the text of one target element, forcing the host to grow while its neighbors absorb the displacement through reflow.
- INSERT-ELEMENT adds one or more new elements into an existing logical-relation block (e.g., a step in a Process), so the host structure must reflow, such as re-routing connectors, and redistributing space among existing siblings.
- SWAP-BLOCK exchanges the canvas positions of two logical-relation blocks as whole units, requiring the rest of the canvas to reflow by resizing and re-aligning around the new arrangement.
- RESHAPE-CANVAS re-renders the infographic at a different aspect ratio, demanding a canvas-level reflow in which every block is re-positioned, re-scaled, and re-aligned to fit the new shape.

Instruction Annotation. For each of the 1,000 infographics, annotators write one instruction for each editing task, yielding 4,000 (infographic, instruction) pairs. We require every instruction to be feasible and reflow-inducing, as verified by three checks. (i) *Spatial coupling*: targets are placed in regions where the edit affects nearby nodes, cards, text fields, or connectors, forcing displacement, resizing, re-wrapping, re-alignment, or rerouting. (ii) *Shape-sensitive anchoring*: targets are drawn from shaped or connected relation instances, such as pyramid tiers or hierarchy branches, to test diverse geometry, family-specific spacing, and connector structure. (iii) *Coverage*: the four instructions for each infographic cover different regions and different relation families.

Difficulty Levels. INFOEDIT defines three difficulty levels for tasks with explicit magnitude axes: EXPAND-TEXT, INSERT-ELEMENT, and SWAP-BLOCK. Easy, medium, and hard correspond to 5–10, 11–20, and 21–30 appended words for EXPAND-TEXT; 1, 2–3, and 4–5 inserted elements for INSERT-ELEMENT; and 1, 2, and 3 swapped block pairs for SWAP-BLOCK. RESHAPE-CANVAS has no discrete magnitude axis. Full difficulty-level statistics are provided in Appx. A.2.1.

2.4 Comparison with Existing Benchmarks

Tab. 1 contrasts INFOEDIT with related benchmarks. Existing image-editing benchmarks (Ye et al., 2025; Pan et al., 2025; Jiang et al., 2026) focus on natural images, often assume pixel-level preservation of unmentioned regions, and do not emphasize logical-relation taxonomies, leaving reflow outside their task design and evaluation. Infographic generation benchmarks (Peng et al., 2025; Li et al., 2026b; Tang et al., 2026) share our domain, but study text-to-image generation, and do not systematically account for logical-relation diversity. In contrast, INFOEDIT first targets infographic editing, organizes samples by logical-relation families, and evaluates reflow-aware edits.

Benchmark	Task	Domain	#Samples	Logical Relation	Reflow-Aware	Evaluation Metric
Image Editing						
ImgEdit (Ye et al., 2025)	Image Editing	Natural	811	✗	✗	MLLM-judge
WiseEdit (Pan et al., 2025)	Image Editing	Natural	1,220	✗	✗	MLLM-judge
GEditBench-v2 (Jiang et al., 2026)	Image Editing	Natural	1,200	✗	✗	Pairwise MLLM
Infographics Generation						
BizGen (Peng et al., 2025)	Text-to-Image	Infographic	1,000	✗	–	MLLM + OCR
BizGenEval (Li et al., 2026b)	Text-to-Image	Infographic	400	✗	–	MLLM-judge
IGenBench (Tang et al., 2026)	Text-to-Image	Infographic	600	✗	–	MLLM-judge
INFOEDIT (Ours)	Image Editing	Infographic	4,000	✓	✓	MLLM-judge

Table 1: Comparison of INFOEDIT with related benchmarks. "–" indicates not applicable to text-to-image generation. The 4,000 samples come from 1,000 infographics with 4 edit tasks each.

3 Evaluation Protocol

We operationalize the compliance and preservation predicates of Sec. 2.1 as two binary aspects: *Edit Compliance* and *Content Preservation*. For each (X, I, X') triple, we prompt an MLLM with the original image, edited image, and instruction, using a task-specific rubric that reflects the edit's reflow contract. The judge returns a JSON response with binary verdicts for the two aspects and an overall verdict equal to their conjunction. The four tasks share this schema but use task-specific criteria; full prompts are provided in Appx. L.

- *Edit Compliance (EC).* EXPAND-TEXT requires the named target to carry a moderately longer, on-topic body text fully rendered inside the canvas. INSERT-ELEMENT requires the new element to appear at the location named in the instruction with its content fully rendered. SWAP-BLOCK requires the two named blocks to exchange canvas locations as whole units, each carrying its content unchanged. RESHAPE-CANVAS requires the canvas to match the target aspect ratio.
- *Content Preservation (CP).* Since INFOEDIT instructions are reflow-inducing, CP does not require pixel-level immobility. Instead, it checks whether unedited source content remains intact after necessary reflow, such as repositioning, resizing, re-alignment, and connector rerouting.

We report *Success Rate (SR)* as the percentage of edits for which both *Edit Compliance* and *Content Preservation* are judged successful.

4 Experiments

4.1 Setup

We benchmark eight frontier image-editing models spanning proprietary and open-source families. The proprietary models include GPT-Image-2 (OpenAI, 2026), Nanobanana-Pro (Google DeepMind, 2025a), Nanobanana-2 (Google DeepMind, 2026b), Nanobanana (Google DeepMind, 2025b), and Seedream-5.0-Lite (ByteDance Seed Team, 2026). The open-source models include Qwen-Image-Edit (Wu et al., 2025a), HunyuanImage-3.0 (Cao et al., 2025), and FLUX.2-klein-base-9B (Black Forest Labs, 2026). All models are queried with our 4,000 (infographic, instruction) pairs under each model's recommended inference settings, and every triple is judged by Gemini-3.1-Pro (Google DeepMind, 2026a), validated against expert human annotators in Sec. 5.4.

4.2 Main Results

A clear capability gap separates the frontier proprietary models from the rest. As shown in Tab. 2, GPT-Image-2 leads at 62.0% average SR, followed by Nanobanana-Pro and Nanobanana-2 at 41.6% and 38.0%. The remaining five models, including Nanobanana, Seedream-5.0-Lite, and all open-weight models, fall at or below 7%. The gap reflects more than a gradual quality difference: many models fail to produce usable infographics at all, underscoring the need for INFOEDIT.

Failure modes are task-specific. The EC/CP decomposition shows that each edit task stresses a different part of reflow. On EXPAND-TEXT, the top three proprietary models largely execute the target edit, with EC between 85.1% and 99.0%, but SR still trails EC because CP remains lower. RESHAPE-CANVAS exposes a stronger preservation bottleneck: the same models achieve high EC (87.2–96.3%), yet CP drops to 36.4–65.1%, indicating that canvas-level reflow often damages source content or visual structure. INSERT-ELEMENT is difficult on both axes, yielding 72.0% SR for GPT-Image-2 but only around 39–40% for

Model	EXPAND-TEXT			INSERT-ELEMENT			SWAP-BLOCK			RESHAPE-CANVAS			Avg.
	EC	CP	SR	EC	CP	SR	EC	CP	SR	EC	CP	SR	
Proprietary													
GPT-Image-2	99.0	86.7	85.9	85.4	80.7	72.0	29.0	79.2	25.5	96.3	65.1	64.4	62.0
Nanobanana-Pro	87.1	64.0	57.8	58.8	57.8	40.2	36.5	79.8	32.0	87.2	36.9	36.4	41.6
Nanobanana-2	85.1	58.6	52.6	65.3	49.4	39.0	30.0	72.1	24.0	92.0	36.4	36.3	38.0
Nanobanana	23.2	23.8	9.2	3.1	14.6	0.4	1.6	41.8	0.9	25.4	8.4	3.5	3.5
Seedream-5.0-Lite	22.1	28.0	9.1	12.3	11.0	1.3	9.3	37.0	5.2	64.1	4.6	4.3	5.0
Open-Weight													
HunyuanImage-3.0	21.3	34.8	12.6	2.5	4.3	0.0	0.4	16.4	0.0	1.4	5.2	0.0	3.2
Qwen-Image-Edit	9.2	3.3	1.5	1.2	2.7	0.0	0.0	6.3	0.0	16.5	0.0	0.0	0.4
FLUX.2-klein-base-9B	6.1	7.2	0.0	0.0	21.2	0.0	0.0	77.8	0.0	20.3	0.0	0.0	0.0

Table 2: Editing performance of eight image-editing models on INFOEDIT across four editing tasks. EC = edit compliance, CP = content preservation, SR = success rate, the conjunction of both. All values are percentages. Avg. is the mean SR across tasks.

the two Nanobanana variants. SWAP-BLOCK is bottlenecked by edit compliance: no model exceeds 36.5% EC or 32.0% SR, while CP can remain comparatively high, showing that compliance and preservation decouple on block-level swaps.

Open-weight models lag behind by a wide margin. HunyuanImage-3.0, Qwen-Image-Edit, and FLUX.2-klein-base-9B reach 3.2%, 0.4%, and 0.0% average SR, all far below the strongest proprietary model GPT-Image-2 (62.0%). The low averages are not driven by a single failure task: all three open-weight models score 0% SR on INSERT-ELEMENT, SWAP-BLOCK, and RESHAPE-CANVAS. High CP alone does not imply success: FLUX.2-klein-base-9B reaches 77.8% CP on SWAP-BLOCK but 0.0% EC, so the requested structural edit is not completed. These results reveal a substantial gap between open-weight and proprietary editors, and INFOEDIT identifies reflow as a concrete target for improving open-weight models.

5 Discussion

5.1 Pixel-Level versus Code-Level Editing

Structured visual content generation and editing remain open challenges for multimodal systems, and can be approached through two pathways (detailed in Appx. B): pixel-level pathway with image generation models, or code-level pathway over textual source. Since every infographic in INFOEDIT is implemented with HTML or PowerPoint, we can directly compare code-level editing with pixel-level image editing under the same evaluation protocol. We test two input modalities: *Code* feeds the model the source code alone, while *Code + Image* additionally provides the rendered original as visual grounding. Gemini-3.1-Pro, Gemini-3.5-Flash and Gemini-3.1-Flash serve as code generators, and we render their edited outputs into images.

As shown in Tab. 3, Gemini-3.1-Pro with *Code + Image* reaches 61.6% average SR, just matching GPT-Image-2's 62.0% average SR. However, the task-level pattern is not uniform. GPT-Image-2 remains stronger on EXPAND-TEXT and RESHAPE-CANVAS, while Gemini-3.1-Pro is comparable on INSERT-ELEMENT and much stronger on SWAP-BLOCK, improving EC from 29.0% to 69.0% and SR from 25.5% to 60.2%. Source code makes named blocks directly addressable, turning block swapping into an explicit structural operation. These findings reveal complementary strengths between the two pathways and point toward future improvements, or even hybrid systems, that combine their respective advantages.

5.2 Interactive Selection Hint

SWAP-BLOCK failures could stem from two distinct causes: the model fails to execute the structural verb (*swap*), or it may just fail to resolve which two blocks the instruction names. To separate these, we annotate the input image with interactive selection hints: each swap pair is marked with bounding boxes and pair indices (detailed in Appx. C), and the model is instructed to treat the marks as localization guidance. If the failure were primarily a target-localization problem, hints should produce a large gain; if the failure is genuinely about reflow, the gain should be modest.

Tab. 4 reports the result. The gain is consistently modest: +3.2 to +7.4 SR across the three strongest editors, with GPT-Image-2 benefiting

Model	EXPAND-TEXT			INSERT-ELEMENT			SWAP-BLOCK			RESHAPE-CANVAS			Avg.
	EC	CP	SR	EC	CP	SR	EC	CP	SR	EC	CP	SR	
Code													
Gemini-3.1-Pro	85.7	72.3	67.4	85.6	70.7	68.0	68.8	77.6	59.6	69.4	45.2	45.1	60.0
Gemini-3.5-Flash	83.3	68.1	64.1	84.1	64.5	62.0	60.1	79.2	54.8	70.4	42.4	42.2	55.8
Gemini-3.1-Flash	79.6	65.5	60.6	63.4	49.1	40.2	56.1	72.7	47.1	38.0	5.3	5.2	38.3
Code + Image													
Gemini-3.1-Pro	88.2	72.0	68.5	85.3	72.2	71.0	69.0	78.4	60.2	71.8	47.2	46.8	61.6
Gemini-3.5-Flash	85.1	69.5	65.8	85.4	67.2	63.2	60.9	79.8	56.8	71.2	46.6	43.1	57.2
Gemini-3.1-Flash	80.2	64.5	59.6	64.3	51.4	43.0	56.4	74.0	50.2	38.1	8.7	8.6	40.4

Table 3: Code-level editing on INFOEDIT. *Code* feeds the source code alone; *code + image* adds the rendered image. EC = edit compliance, CP = content preservation, SR = the conjunction of both. All values are percentages. Avg. is the mean SR across tasks.

Model	Hint	EC	CP	SR	ΔSR
GPT-Image-2	w/o	29.0	79.2	25.5	–
	w/	34.2	92.7	32.9	+7.4
Nanobanana-Pro	w/o	36.5	79.8	32.0	–
	w/	39.4	83.6	35.2	+3.2
Nanobanana-2	w/o	30.0	72.1	24.0	–
	w/	34.2	79.1	27.5	+3.5

Table 4: Effect of interactive selection hints on SWAP-BLOCK performance.

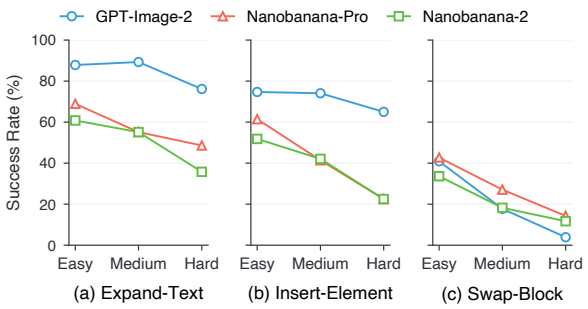


Figure 4: Success rates across difficulty levels for edit tasks: EXPAND-TEXT, INSERT-ELEMENT, and SWAP-BLOCK.

most (25.5 → 32.9) and the two NanoBanana variants gaining only 3–3.5 points. Even with perfect localization, no model exceeds 36% SR on SWAP-BLOCK. The bottleneck INFOEDIT surfaces is therefore not simply the inability to localize the target blocks; it is the reflow operation itself, which current editors cannot reliably perform even when the targets are unambiguously marked.

5.3 Different Difficulty Levels

Fig. 4 breaks down performance by difficulty level for the three tasks with explicit magnitude axes: EXPAND-TEXT, INSERT-ELEMENT, and SWAP-BLOCK. Across all three tasks, success rates generally decrease as requested edits become harder, confirming that INFOEDIT captures graded difficulty. The degradation is mildest for EXPAND-TEXT, where top models can often absorb longer text by shrinking fonts or locally re-wrapping neighboring elements. In contrast, INSERT-ELEMENT shows a sharper drop, especially for weaker models, because adding multiple elements requires redistributing existing space and maintaining logical-relation consistency. SWAP-BLOCK is the most difficult across all settings: even at the easy level, models achieve substantially lower success rates than on the other two tasks, and performance falls quickly as more block pairs are exchanged.

5.4 Human Agreement and Judge Robustness

To validate our MLLM-as-a-judge protocol, three expert annotators independently evaluate 800 outputs from each of GPT-Image-2, Nanobanana-2, and HunyuanImage-3.0, balanced across the four editing tasks. Their majority verdict serves as the human reference. Gemini-3.1-Pro achieves agreement of 0.920 to 0.981 across the three models. Agreement also remains high when stratified by task, difficulty, and predicted outcome, and is stable under a cross-vendor judge and repeated evaluation. Full results are reported in Appx. E.

5.5 Error Analysis

We decompose each failure case from GPT-Image-2, the strongest editor we evaluate, into the diagnostic sub-fields of Sec. 3 and report the frequency of each failure type in Fig. 5. Three patterns emerge.

First, MISPLACEMENT dominates the structurally demanding tasks, accounting for 79% of SWAP-BLOCK failures and 42% of INSERT-ELEMENT failures. The model can identify that

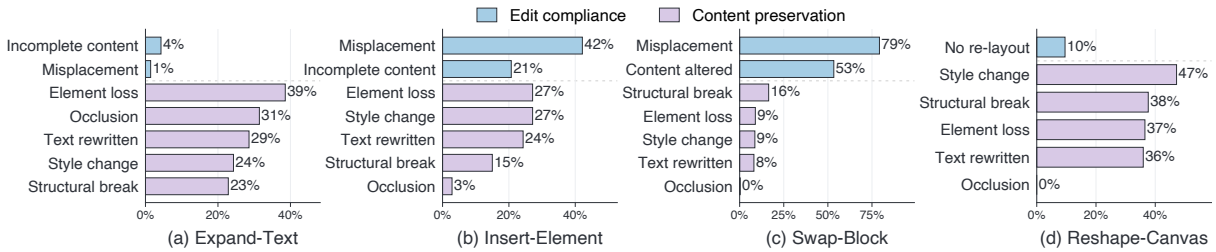


Figure 5: Distribution of failure types per task. We categorize unsuccessful cases by the diagnostic sub-field they violate, separated into Edit Compliance (blue) and Content Preservation (purple). Detailed error cases in Appx. K.

a swap or insertion is required, but cannot place the named block or element correctly through reflow; this matches the finding in Sec. 5.2 that interactive selection hints only partially relieve.

Second, EXPAND-TEXT exhibits a different pattern: edit compliance rarely fails ($\leq 4\%$), while most errors arise from content preservation failures. The target text is often expanded correctly, but nearby nodes fail to reflow, causing element loss, occlusion or structural break. This silent-damage pattern highlights the need to evaluate content preservation separately from edit compliance.

Third, RESHAPE-CANVAS concentrates its failures on global stylistic and structural breakdown (47% style change, 38% structural break), indicating that re-rendering at a new aspect ratio rarely preserves the original visual organization. Even when the canvas reaches the target ratio, the layout inside is re-painted, losing the source palette and structural relation. Occlusion is the most common preservation failure on EXPAND-TEXT at 31%, but drops to 0% on SWAP-BLOCK and RESHAPE-CANVAS, where elements redistribute globally rather than overwhelm a neighbor.

Overall, the failures show that reflow is multi-faceted rather than a single operation: INSERT-ELEMENT and SWAP-BLOCK stress correct placement, EXPAND-TEXT stresses neighbor-aware adjustment, and RESHAPE-CANVAS stresses global style and structure preservation.

6 Related Work

Image Editing Benchmarks. Existing image-editing benchmarks largely follow a pixel-identity assumption. ImgEdit-Bench (Ye et al., 2025) evaluates natural-image edits on instruction adherence, editing quality, and detail preservation, while WiseEdit (Pan et al., 2025) introduces more cognition- and creativity-informed edits. Other benchmarks and editing datasets (Zhang et al., 2023; Brooks et al., 2023; Hui et al., 2024; Ge et al., 2024; Zhao et al., 2024; Ma et al., 2024; Yu et al., 2025; Jia et al., 2025; Gao et al., 2026; Zhao et al., 2025) differ in scale, operation taxonomy, and annotation cost, but similarly treat unmentioned regions as content to preserve unchanged. This pixel-identity assumption is reasonable for natural photographs, but breaks down on structured content, where edits often require layout changes. These benchmarks, therefore, do not target structured visual content editing or evaluate the required reflow capability.

Structured Visual Content Generation. Current multimodal foundation models have demonstrated strong capabilities in generating structured visual content such as infographics (Nallatamby et al., 2025; Yang et al., 2025; Google Labs, 2026; Zhu et al., 2026a,b), and recent benchmarks evaluate this regime: BizGen (Peng et al., 2025) on visual text rendering for infographics and slides, IGenBench (Tang et al., 2026) on text-to-infographic reliability through atomic yes/no questions, and BizGenEval (Li et al., 2026b) on diverse commercial document generation. These works focus on creating structured visuals from text, whereas INFOEDIT studies editing existing infographics with diverse logical relations while preserving their structure through reflow.

7 Conclusion

We present INFOEDIT, a novel benchmark for infographic editing that probes *reflow*, a capability for which current image-editing models are not systematically evaluated. INFOEDIT covers $1{,}000$ infographics across eight logical-relation families and $4{,}000$ instruction-paired editing tasks, together with a reflow-aware evaluation protocol that decomposes each verdict into edit compliance and content preservation. Across eight frontier editors, only GPT-Image-2 clears 60% average success rate,

and no model exceeds 36% on SWAP-BLOCK even when given perfect target localization, identifying reflow as the dominant bottleneck. We further examine code-level editing and conduct error analyses to provide additional diagnostic insights. We anticipate that INFOEDIT will motivate training and evaluation frameworks that treat reflow as a central capability of structured visual content editing.

Limitations

INFOEDIT evaluates single-turn editing instructions to cleanly attribute each outcome to a specific editing task; real-world design workflows may involve multi-turn refinements, where users iteratively adjust content, layout, and style. Extending reflow evaluation to such interactive settings is another promising direction.

Acknowledgements

This research was supported in part by the NSF under grants 2146151 and 2422214.

References

Black Forest Labs. 2026. FLUX.2 [klein]: Towards interactive visual intelligence. https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence. Accessed: 2026-04-30.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402.

ByteDance Seed Team. 2026. Seedream 5.0 Lite. https://seed.bytedance.com/en/seedream5_0_lite. Accessed: 2026-04-30.

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, and 1 others. 2025. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, and 1 others. 2025. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*.

Xin Gao, Cheng Yang, Chufan Shi, and Taylor Berg-Kirkpatrick. 2026. Do text edits generalize to visual generation? benchmarking cross-modal knowledge editing in umms. *arXiv preprint arXiv:2606.00477*.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and 1 others. 2025. Autopresent: Designing structured visuals from scratch. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2902–2911.

Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. 2024. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*.

Google DeepMind. 2025a. Introducing Nano Banana Pro. https://blog.google/innovation-and-ai/products/nano-banana-pro/. Accessed: 2026-04-30.

Google DeepMind. 2025b. Nano Banana image generation. https://ai.google.dev/gemini-api/docs/image-generation. Accessed: 2026-04-30.

Google DeepMind. 2026a. Gemini 3.1 Pro: A smarter model for your most complex tasks. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/. Accessed: 2026-04-30.

Google DeepMind. 2026b. Nano Banana 2: Combining pro capabilities with lightning-fast speed. https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/. Accessed: 2026-04-30.

Google Labs. 2026. Gemini Notebook. https://notebooklm.google.com/. Accessed: Aug 1, 2026.

Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Liangliang Cao, and Shifeng Chen. 2025a. Diffusion model-based image editing: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zheng Huang, Xukai Liu, Tianyu Hu, Kai Zhang, and Ye Liu. 2025b. Pptbench: Towards holistic evaluation of large language models for powerpoint layout and design understanding. *arXiv preprint arXiv:2512.02624*.

Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. 2024. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*.

Vijay Jaisankar, Sambaran Bandyopadhyay, Kalp Vyas, Varre Chaitanya, and Shwetha Somasundaram. 2024. Postdoc: Generating poster from a long multimodal document using deep submodular optimization. *arXiv preprint arXiv:2405.20213*.

Bohan Jia, Wenxuan Huang, Yuntian Tang, Junbo Qiao, Jincheng Liao, Shaosheng Cao, Fei Zhao, Zhaopeng Feng, Zhouhong Gu, Zhenfei Yin, and 1 others. 2025. Compbench: Benchmarking complex instruction-guided image editing. *arXiv preprint arXiv:2505.12200*.

Zhangqi Jiang, Zheng Sun, Xianfang Zeng, Yufeng Yang, Xuanyang Zhang, Yongliang Wu, Wei Cheng, Gang Yu, Xu Yang, and Bihan Wen. 2026. Gedit-bench v2: A human-aligned benchmark for general image editing. *arXiv preprint arXiv:2603.28547*.

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

Shuo Li, Jiajun Sun, Zhekai Wang, Xiaoran Fan, Hui Li, Dingwen Yang, Zhiheng Xi, Yijun Wang, Zifei Shan, Tao Gui, and 1 others. 2026a. ChartE^3: A comprehensive benchmark for end-to-end chart editing. *arXiv preprint arXiv:2601.21694*.

Yan Li, Zezi Zeng, Ziwei Zhou, Xin Gao, Muzhao Tian, Yifan Yang, Mingxi Cheng, Qi Dai, Yuqing Yang, Lili Qiu, and 1 others. 2026b. Bizgeneval: A systematic benchmark for commercial visual content generation. *arXiv preprint arXiv:2603.25732*.

Zhen Li, Duan Li, Yukai Guo, Xinyuan Guo, Bowen Li, Lanxi Xiao, Shenyu Qiao, Jiashu Chen, Zijian Wu, Hui Zhang, and 1 others. 2025. Chartgalaxy: A dataset for infographic chart understanding and generation. *arXiv preprint arXiv:2505.18668*.

Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, and 1 others. 2025. Uniworld-v1: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, and 1 others. 2025. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*.

Yiwei Ma, Jiayi Ji, Ke Ye, Weihuang Lin, Zhibin Wang, Yonghan Zheng, Qiang Zhou, Xiaoshuai Sun, and Rongrong Ji. 2024. I2ebench: A comprehensive benchmark for instruction-based image editing. *Advances in Neural Information Processing Systems*, 37:41494–41516.

Himanshu Maheshwari, Sambaran Bandyopadhyay, Aparna Garimella, and Anandhavelu Natarajan. 2024. Presentations are not always linear! gnn meets llm for text document-to-presentation transformation with attribution. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15948–15962.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706.

Microsoft. All SmartArt graphics, described. https://support.microsoft.com/en-us/office/all-smartart-graphics-described-cf1a453b-de4a-4217-8da0-1aff97bb32cd. Accessed: 2026-05-01.

Ishani Mondal, Zongxia Li, Yufang Hou, Anandhavelu Natarajan, Aparna Garimella, and Jordan Lee Boyd-Graber. 2024. Scidoc2diagrammer-maf: Towards generation of scientific diagrams from documents guided by multi-aspect feedback refinement. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13342–13375.

Vincent Nallatamby, Arnaud Benard, and Sam El-Husseini. 2025. From idea to app: Introducing Stitch, a new way to design UIs. https://developers.googleblog.com/en/stitch-a-new-way-to-design-uis/. Announced at Google I/O 2025. Accessed: 2026-05-01.

OpenAI. 2026. Introducing ChatGPT Images 2.0. https://openai.com/index/introducing-chatgpt-images-2-0/. Accessed: 2026-04-30.

Kaihang Pan, Weile Chen, Haiyi Qiu, Qifan Yu, Wendong Bu, Zehan Wang, Yun Zhu, Juncheng Li, and Siliang Tang. 2025. Wiseedit: Benchmarking cognition-and creativity-informed image editing. *arXiv preprint arXiv:2512.00387*.

Yuyang Peng, Shishi Xiao, Keming Wu, Qisheng Liao, Bohan Chen, Kevin Lin, Danqing Huang, Ji Li, and Yuhui Yuan. 2025. Bizgen: Advancing article-level visual text rendering for infographics generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23615–23624.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. 2024. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8871–8879.

Chufan Shi, Cheng Yang, Yaokang Wu, Linghao Jin, Bo Shui, Taylor Berg-Kirkpatrick, and Xuezhe Ma. 2026. Are vlms seeing or just saying? uncovering the illusion of visual re-examination. *arXiv preprint arXiv:2605.15864*.

Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. 2024. A thorough examination of decoding methods in the era of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8601–8629. Association for Computational Linguistics.

Xiaoyan Su, Peijie Dong, Zhenheng Tang, Song Tang, Yuyao Zhai, Kaitao Lin, Liang Chen, Gai Yuhang, Yuyu Luo, Qiang Wang, and 1 others. 2026. Vcg-bench: Towards a unified visual-centric benchmark for structured generation and editing. *arXiv preprint arXiv:2605.15677*.

Yinghao Tang, Xueding Liu, Boyuan Zhang, Tingfeng Lan, Yupeng Xie, Jiale Lao, Yiyao Wang, Haoxuan Li, Tingting Gao, Bo Pan, and 1 others. 2026. Igenbench: Benchmarking the reliability of text-to-infographic generation. *arXiv preprint arXiv:2601.04498*.

Muzi Tao, Chufan Shi, Huijuan Wang, Shengbang Tong, and Xuezhe Ma. 2026. Asymmetric idiosyncrasies in multimodal models. *arXiv preprint arXiv:2602.22734*.

Hongyang Wei, Baixin Xu, Hongbo Liu, Size Wu, Jie Liu, Yi Peng, Peiyu Wang, Zexiang Liu, Jingwen He, Yidan Xietian, and 1 others. 2025. Skywork unipic 2.0: Building kontext model with online rl for unified multimodal model. *arXiv preprint arXiv:2509.04548*.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, and 1 others. 2025a. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, and 1 others. 2025b. Omnigen2: Towards instruction-aligned multimodal generation. *arXiv preprint arXiv:2506.18871*.

Tianchi Xie, Minzhi Lin, Mengchen Liu, Yilin Ye, Changjian Chen, and Shixia Liu. 2025. Infochartqa: A benchmark for multimodal question answering on infographic charts. In *Advances in Neural Information Processing Systems*, volume 38.

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, and 1 others. 2025. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation. In *International Conference on Learning Representations*.

Cheng Yang, Chufan Shi, Bo Shui, Yaokang Wu, Muzi Tao, Huijuan Wang, Ivan Yee Lee, Yong Liu, Xuezhe Ma, and Taylor Berg-Kirkpatrick. 2026. From reasoning to pixels: Benchmarking the alignment gap in unified multimodal models. *Preprint*, arXiv:2602.08336.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. 2025. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. 2025. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26125–26135.

Ziyun Zeng, David Junhao Zhang, Wei Li, and Mike Zheng Shou. 2025. Draw-in-mind: Rebalancing designer-painter roles in unified multimodal models benefits image editing. *arXiv preprint arXiv:2509.01986*.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. 2023. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449.

Shuoshuo Zhang, Zijian Li, Yizhen Zhang, Jingjing Fu, Lei Song, Jiang Bian, Jun Zhang, Yujiu Yang, and Rui Wang. 2026. Pixelcraft: A multi-agent system for high-fidelity visual reasoning on structured images. In *International Conference on Learning Representations*.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. 2024. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093.

Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, and 1 others. 2025. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. *Advances in Neural Information Processing Systems*, 38.

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2025. Pptagent: Generating and evaluating presentations beyond text-to-slides. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14402–14418.

Dawei Zhu, Rui Meng, Yale Song, Xiyu Wei, Sujian Li, Tomas Pfister, and Jinsung Yoon. 2026a. Paperbanana: Automating academic illustration for ai scientists. *arXiv preprint arXiv:2601.23265*.

Jiangning Zhu, Yuxing Zhou, Zheng Wang, Juntao Yao, Yima Gu, Yuhui Yuan, and Shixia Liu. 2025. Infodet: A dataset for infographic element detection. *arXiv preprint arXiv:2505.17473*.

Minjun Zhu, Zhen Lin, Yixuan Weng, Panzhong Lu, Qiujie Xie, Yifan Wei, Sifan Liu, Qiyao Sun, and Yue Zhang. 2026b. Autofigure: Generating and refining publication-ready scientific illustrations. *arXiv preprint arXiv:2602.03828*.

Le Zhuo, Songhao Han, Yuandong Pu, Boxiang Qiu, Sayak Paul, Yue Liao, Yihao Liu, Jie Shao, Xi Chen, Si Liu, and 1 others. 2025. Factuality matters: When image generation and editing meet structured visuals. *arXiv preprint arXiv:2510.05091*.

Appendix

A Details on Benchmark Construction **13**
- A.1 Infographics Construction 13
- A.2 Editing Tasks 14
- A.3 Data Examples 18

B Additional Related Work **18**

C Details on Interactive Selection Hint **19**

D Details on Code Generation Baselines **20**

E Evaluation Reliability **20**
- E.1 Human Agreement across Output Models 20
- E.2 Stratified Agreement 22
- E.3 Alternative Judge 23
- E.4 Repeated Evaluation 23

F Fine-Grained Results **23**
- F.1 Logical-Relation Families 23
- F.2 HTML versus PowerPoint Sources 23
- F.3 Canvas Transformation Types . . . 24

G Additional Robustness Analyses **24**
- G.1 Color-Palette Sensitivity 24
- G.2 Multilingual Pilot 24

H Ethical Statement **24**

I Reproducibility **24**

J Models **24**

K Error Analysis **24**

L Prompts **26**
- L.1 MLLM-as-a-Judge Evaluation Prompts 26

A Details on Benchmark Construction

A.1 Infographics Construction

Existing infographic resources (Mathew et al., 2022; Li et al., 2025; Zhu et al., 2025; Xie et al., 2025; Tang et al., 2026; Zhang et al., 2026) do not satisfy the requirements of infographic editing evaluation. First, many datasets provide only rendered images, sometimes with limited visual quality, and lack editable source files. This prevents code-level editing and limits diagnostic analysis, since target elements and surrounding structures are not explicitly addressable. Consequently, addressable tasks (detailed in Appx. A.2.2) would have to be defined by visual intuition over pixels rather than by concrete operations on editable HTML or PowerPoint objects. Second, many existing infographic datasets are designed for VQA, OCR, or chart-centric understanding and generation, where the main challenge is reading text, extracting values, or interpreting charts. They therefore offer limited structural diversity and do not systematically cover logical relations. We thus build INFOEDIT from logical-relation templates and editable sources, so that each sample supports feasible, non-trivial, and reflow-inducing edits.

A.1.1 Stage 1: Topic Sourcing

We define 31 topic categories spanning domains in which infographics are commonly produced (e.g., companies, products, athletes). To reduce the topical and stylistic bias that may arise from relying on a single language model, we use a multi-model topic sourcing procedure. For each category, Claude-Opus-4.7 first enumerates 70 concrete topics. We then provide the existing list to Gemini-3.1-Pro and ask it to supplement the category with additional non-duplicate topics. Finally, we provide the accumulated list to GPT-5.4 and ask it to further complete the category until the list reaches 200 topics. The resulting topics are deduplicated within each category and shuffled before entering the next stage. Across 31 categories, this process yields 6,200 candidate topics.

A.1.2 Stage 2: Visual Reference Collection

For each candidate topic, we generate a richly designed reference infographic that serves as a reference for color palette. To avoid tying the color palette of the benchmark to a single image generation model, we split the 6,200 candidate topics evenly across two generators, NanoBanana-Pro and GPT-Image-1.5, with each model generating approximately half of the reference images. Five expert annotators independently filter these candidates down to 1,000 references, keeping only those with coherent palettes, and topic-faithful content. They serve as reference that supply concrete guidance on color palette for the rest of the pipeline, sparing annotators from designing color from scratch.

Family	#Templates	Family	#Templates
List	16	Matrix	6
Process	16	Pyramid	6
Relationship	15	Hierarchy	4
Cycle	7	Picture	9
		Total	**79**

Table 5: Templates per logical-relation family in the logical template library. Each template is implemented in both HTML and PowerPoint.

A.1.3 Stage 3: Logical Template Construction

We build a logical template library that instantiates the 8 logical-relation families, comprising 79 templates in total, distributed across families as in Table 5. These families abstract common organization patterns in structured visual content.

- LIST templates present grouped or non-sequential items with comparable importance;
- PROCESS templates encode ordered steps, timelines, or directional workflows;
- CYCLE templates represent recurring stages or closed-loop dependencies;
- HIERARCHY templates express parent–child, reporting, or multi-level tree relations;
- RELATIONSHIP templates capture comparisons, overlaps, containment, convergence, opposition, or central–peripheral associations;
- MATRIX templates organize concepts by quadrants or two-dimensional axes;
- PYRAMID templates convey layered, proportional, or bottom-up/top-down structures;
- PICTURE templates use image slots as first-class carriers of information, often paired with captions or short textual annotations.

Each template is implemented twice: an HTML version that exposes every node, edge, and container as an addressable DOM element, and a PowerPoint version that realizes the same structure as named vector shapes and connectors. Examples from the template library are shown in Figs. 6–13,

covering the eight logical-relation families from LIST to PICTURE. Importantly, these templates specify only the underlying logical structure; during infographic composition, they are further specialized in visual style, color palette, spacing, and content to match the target infographic design.

A.1.4 Stage 4: Infographic Composition

With references and templates in hand, an infographic is composed by transferring the color palette of a reference image onto logical-relation templates. The HTML track (800 infographics) is model-assisted: the annotator selects templates from the library that fit the topic and logical structure and assembles them; Claude-Opus-4.7 is then invoked in two sequential passes, a *styling pass* that adapts the templates' CSS to match the reference image's palette, and a *content pass* that populates textual placeholders with topic-relevant text from public source, such as Wikipedia. The reference is used as a color palette cue, not as content or a layout to reproduce. The resulting draft is then post-edited by the annotator to fix layout artifacts, adjust visual details, and verify the topic-specific content. The PowerPoint track (200 infographics) is fully manual: the annotator picks templates, arranges them, adjusts colors and typography, and writes the topic text. We use these two authoring tracks because HTML and PowerPoint are both common and accessible ways of creating infographics in practice, while also exposing structured layout objects that can be edited and evaluated programmatically. Some templates may include chart-like components to improve visual diversity and better reflect real-world infographic design. However, our editing instructions focus on infographic-level reflow rather than specific chart-internal facts such as numerical values, axes, or plotted trends, whose fine-grained factuality has been explored by Zhuo et al. (2025) and Li et al. (2026a).

We design the construction pipeline to avoid making the benchmark dependent on the color palette of any single upstream model. Reference images are generated by two different models and are used only to provide high-level cues for color palette. The final infographics are not copied from these references; instead, annotators rebuild them from logical-relation templates and manually post-edit colors, spacing, typography, and layout details. In addition, INFOEDIT is an editing benchmark rather than a text-to-image generation benchmark: models are evaluated on modifying an existing infographic, not on generating a new image from the reference. This design makes the benchmark less sensitive to the color preferences of the reference generators.

A.1.5 Stage 5: Quality Control

Every infographic is reviewed by a second annotator who did not participate in its creation, against three criteria: (i) *structural integrity*, every logical-relation instance is complete and renders without overflow or broken connectors; (ii) *stylistic faithfulness*, the rendered output matches the reference image in palette; (iii) *content completeness*, all textual fields and labels are coherent, topic-relevant, fully readable, and compatible with their logical roles, with no placeholders, truncation, duplicated boilerplate, or garbled text.

A.1.6 Topic Distribution

The final set of 1,000 infographics covers a broad range of real-world topics. As shown in Fig. 14, the distribution spans 31 topic categories, including companies, management, technology, finance, open-source projects, cities, products, landmarks, universities, entertainment, sports, and public figures. This diversity is intended to reduce topic-specific bias and ensure that INFOEDIT evaluates structured-content editing across varied infographic use cases rather than a narrow domain.

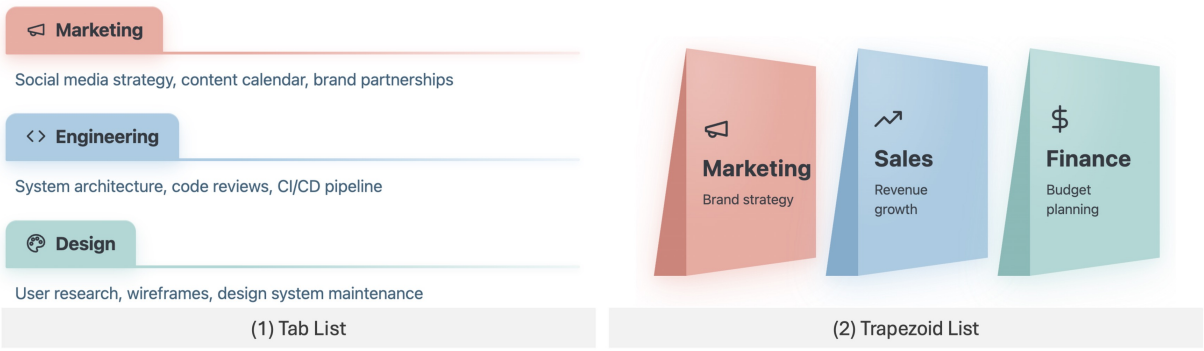
A.2 Editing Tasks

A.2.1 Difficulty-Level Statistics

For the three editing tasks with an explicit magnitude axis, we assign each instruction to a difficulty level according to the size of the requested structural perturbation. For EXPAND-TEXT, difficulty is determined by the number of additional words appended to the target element. For INSERT-ELEMENT, difficulty is determined by the number of new elements inserted into the existing structure. For SWAP-BLOCK, difficulty is determined by the number of block pairs to be exchanged. RESHAPE-CANVAS does not use a discrete magnitude axis; its difficulty depends instead on the source–target aspect-ratio transformation, since different transformations impose different degrees of global relayout. Table 6 reports the number of samples at each difficulty level.

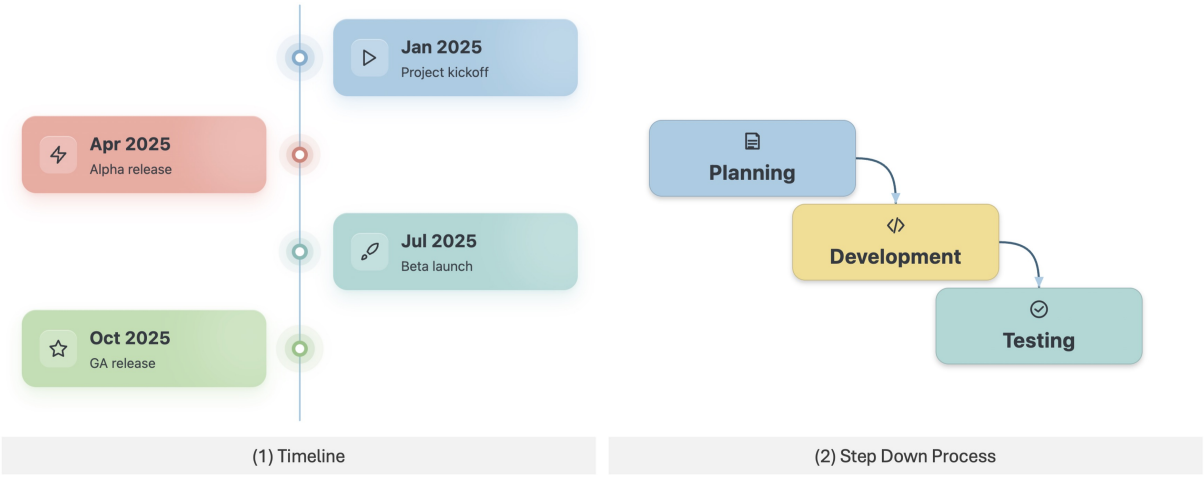
A.2.2 Addressable Tasks

All editing instructions in INFOEDIT are manually annotated to be *addressable*: each requested edit



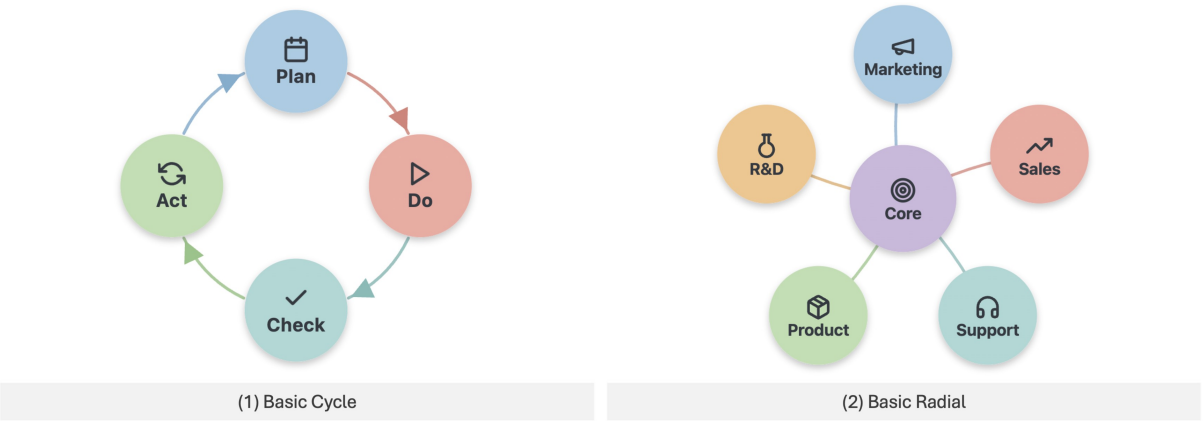
(1) Tab List (2) Trapezoid List

Figure 6: LIST: Tab List and Trapezoid List.



(1) Timeline (2) Step Down Process

Figure 7: PROCESS: Timeline and Step Down Process.



(1) Basic Cycle (2) Basic Radial

Figure 8: CYCLE: Basic Cycle and Basic Radial.

must correspond to a concrete and feasible structural operation on the original infographic. Since every infographic is paired with an editable source file, annotators can verify addressability against the underlying HTML or PowerPoint objects rather than relying only on visual intuition from the rendered image. Annotators verify that the target element or block can be unambiguously identified and that the requested change can be completed without removing content or breaking the underlying logical relation.

Specifically, EXPAND-TEXT instructions target text fields that can reasonably support longer content through wrapping, resizing, or local reflow. INSERT-ELEMENT instructions are applied only to relation instances that can accommodate the requested new elements while preserving their semantics. SWAP-BLOCK instructions require both

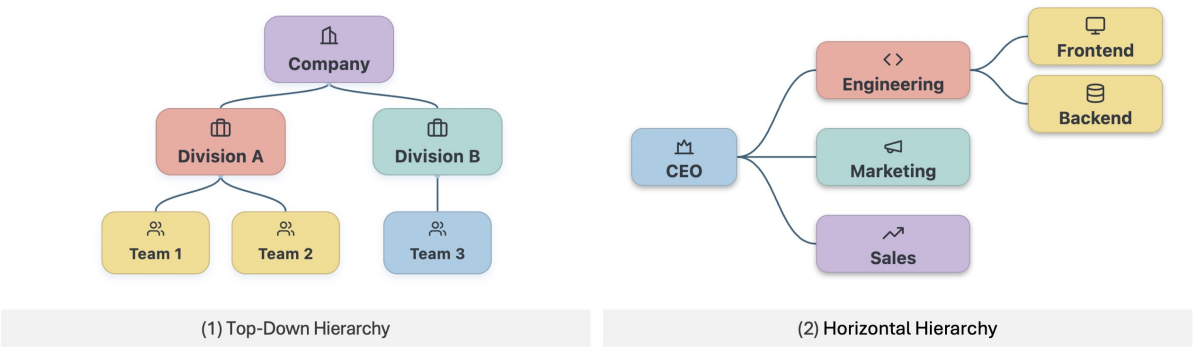


Figure 9: HIERARCHY: Top-Down Hierarchy and Horizontal Hierarchy.

Figure 10: RELATIONSHIP: Basic Venn and Nested Target.

Figure 11: MATRIX: Basic Matrix and Titled Matrix.

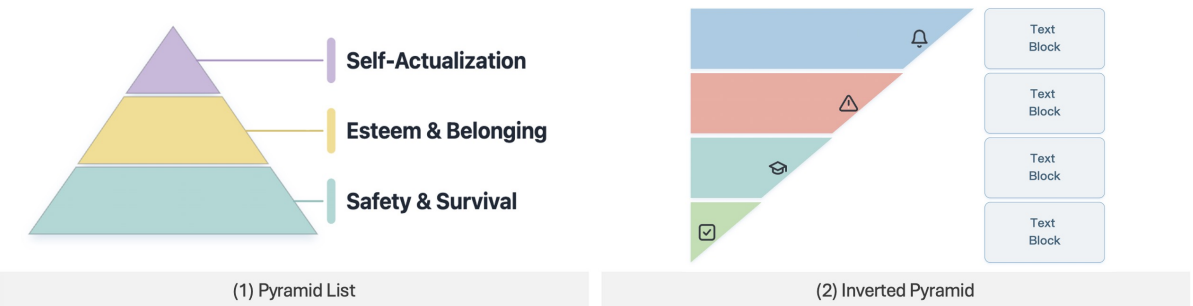


Figure 12: PYRAMID: Pyramid List and Inverted Pyramid.

targets to be coherent layout modules that can move as whole units. RESHAPE-CANVAS instructions are kept only when the original content can fit the target aspect ratio through global re-layout rather than content removal.

We further require that the task is non-trivial: the edit should not be solvable by simply pasting content into empty whitespace, but should require

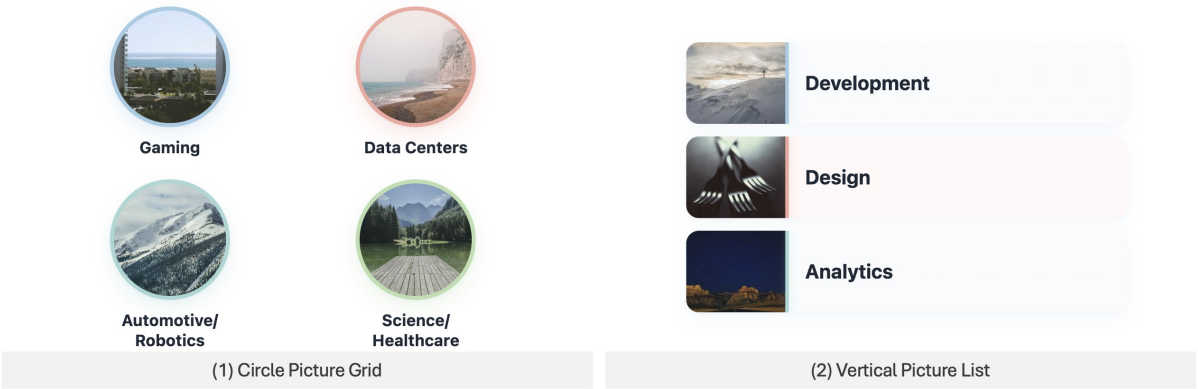


Figure 13: PICTURE: Circle Picture Grid and Vertical Picture List.

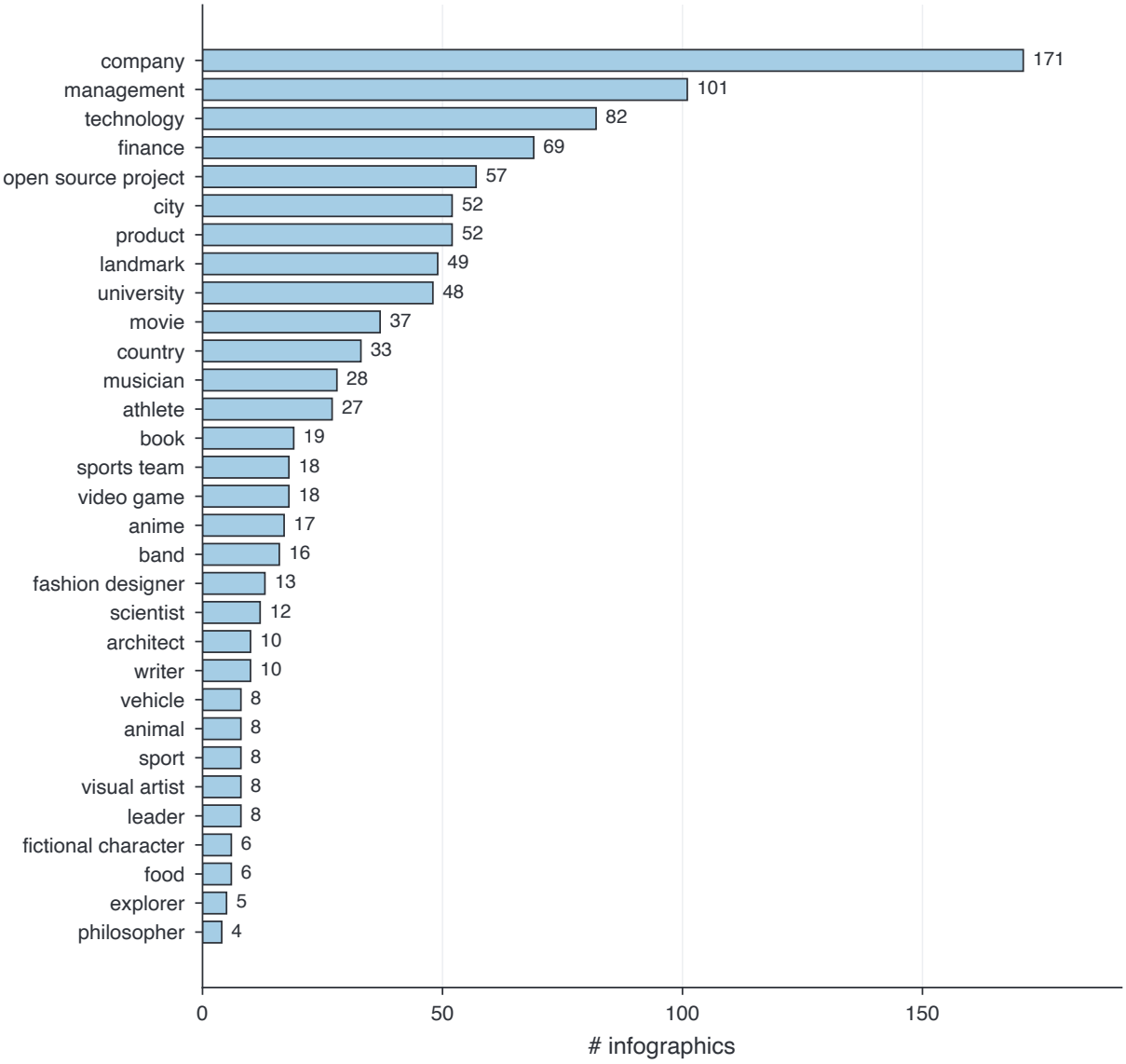


Figure 14: Distribution of infographic topics across 31 categories in INFOEDIT, demonstrating broad coverage of real-world domains.

some degree of reflow. A second annotator reviews each instruction for target clarity, feasibility, and non-triviality; invalid or underspecified instructions are revised or discarded.

Task	Magnitude Bucket	#Samples	Ratio	Difficulty
EXPAND-TEXT	5–10 words	285	28.50%	Easy
	11–15 words	302	30.20%	Medium
	16–20 words	205	20.50%	Medium
	21–25 words	99	9.90%	Hard
	26–30 words	109	10.90%	Hard
INSERT-ELEMENT	1 element	170	17.00%	Easy
	2 elements	323	32.30%	Medium
	3 elements	257	25.70%	Medium
	4 elements	174	17.40%	Hard
	5 elements	76	7.60%	Hard
SWAP-BLOCK	1 pair	432	43.20%	Easy
	2 pairs	420	42.00%	Medium
	3 pairs	148	14.80%	Hard

Table 6: Difficulty-level statistics for editing tasks with explicit magnitude axes. Counts are reported over 1,000 instructions per task.

A.3 Data Examples

We provide representative examples for the four editing tasks in Figs. 15–18. Each example contains the original infographic, the natural-language editing instruction, an expected edited image, and an unexpected edited image. The expected edited image illustrates a valid edit that satisfies both edit compliance and content preservation, while the unexpected edited image highlights typical failures where the requested change is incomplete or the surrounding structured content is not properly reflowed. Together, these examples illustrate how INFOEDIT evaluates not only whether the target edit is executed, but also whether the rest of the infographic remains readable, coherent, and logically valid after reflow.

B Additional Related Work

Image Editing Models. Instruction-based image editing has progressed from diffusion-specialized pipelines (Brooks et al., 2023; Sheynin et al., 2024; Hui et al., 2024) to unified multimodal models with a shared backbone for image understanding and generation (Huang et al., 2025a; Deng et al., 2025; Wu et al., 2025b; Lin et al., 2025). The current generation, both closed-source (OpenAI, 2026; Google DeepMind, 2025a; ByteDance Seed Team, 2026) and open-source (Liu et al., 2025; Wei et al., 2025; Zeng et al., 2025; Labs et al., 2025; Wu et al., 2025a), performs multi-turn, identity-preserving edits at near-production quality. Yet across this lineage, models are trained and ranked under one implicit assumption: the instruction names a region, the model edits that region, and the rest of the canvas is held pixel-identical to the source.

Code-Level vs. Pixel-Level Structured Visual Generation and Editing. Structured visual content can be generated and edited through two broad pathways. The code-level pathway represents visual artifacts as editable source, such as HTML, PowerPoint, or diagram code, and then renders the source into the final visual output (Jaisankar et al., 2024; Mondal et al., 2024; Maheshwari et al., 2024; Ge et al., 2025; Zheng et al., 2025; Yang et al., 2025; Su et al., 2026; Huang et al., 2025b). This representation makes structural elements addressable and enables deterministic rendering, but it also requires access to editable source and robust program-level generation or modification. Historically, this code-level pathway has dominated structured visual generation, since earlier image-generation models struggled to render complete and legible text, let alone produce coherent infographics. Recent pixel-level models, however, such as Nano Banana and GPT-Image-2, can render readable text and generate visually appealing infographics, motivating renewed interest in direct pixel-level generation and editing. In contrast, the pixel-level pathway directly produces or edits rendered images, as in recent structured-content generation and editing systems (OpenAI, 2026; Google DeepMind, 2025a; ByteDance Seed Team, 2026; Wu et al., 2025a; Nallatamby et al., 2025; Google Labs, 2026; Zhu et al., 2026a; Shi et al., 2026; Zhu et al., 2026b). This pathway is more general when source files are unavailable, but it must infer or maintain the latent structure of the visual artifact from pixels alone. Faithfully realizing textual spec-

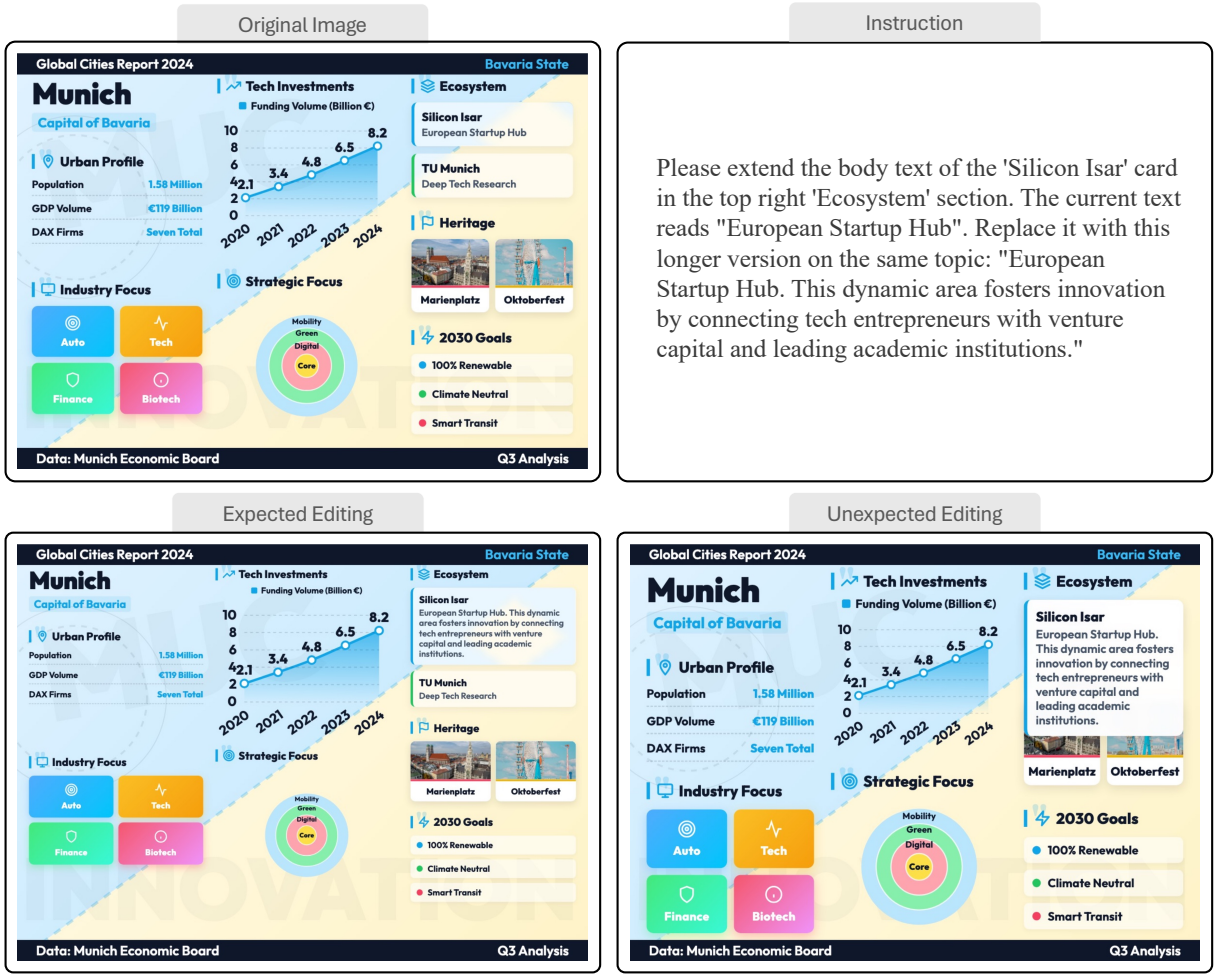


Figure 15: Example of the EXPAND-TEXT task in INFOEDIT. The editor must lengthen the body text of a specified card while preserving its content. The expected edit reflows nearby elements to maintain readability and layout consistency, whereas the unexpected edit inserts the longer text without properly adjusting the local layout.

ifications in generated images remains challenging across multimodal generation settings (Yang et al., 2026; Tao et al., 2026).

Which pathway is better suited for structured visual content remains an open question. In Sec. 5, we take a first step toward examining this question in the editing setting by comparing pixel-level image editors with code-level editing over the editable sources provided in INFOEDIT. This comparison is not intended as a like-for-like competition, since the code-level pathway has privileged access to editable source objects, while pixel-level editors operate only on rendered images. Instead, our goal is to characterize the trade-off between two practical deployment regimes: source-available editing and source-free image editing. Our results show that the two pathways achieve comparable overall performance but exhibit complementary strengths across tasks. In particular, editable sources provide strong structural addressability for operations such as SWAP-BLOCK, while pixel-level editors remain more general when source files are unavailable. These findings suggest that future systems may benefit from combining pixel-level visual flexibility with code-level structural addressability.

C Details on Interactive Selection Hint

For the hinted version of SWAP-BLOCK, we add visual localization marks to the input image. Each target block is annotated with a tight bounding box, and blocks assigned the same numeric label form a swap pair. For example, two blocks labeled "(1)" are swapped. The annotation is made at the block level, where the box covers the whole logical module, including its title, text, icons, charts, connectors, and local container when applicable.

The marked image is used only as an auxiliary input to help the model identify the swap targets. The editing instruction is unchanged except for an additional note that the boxes and labels are

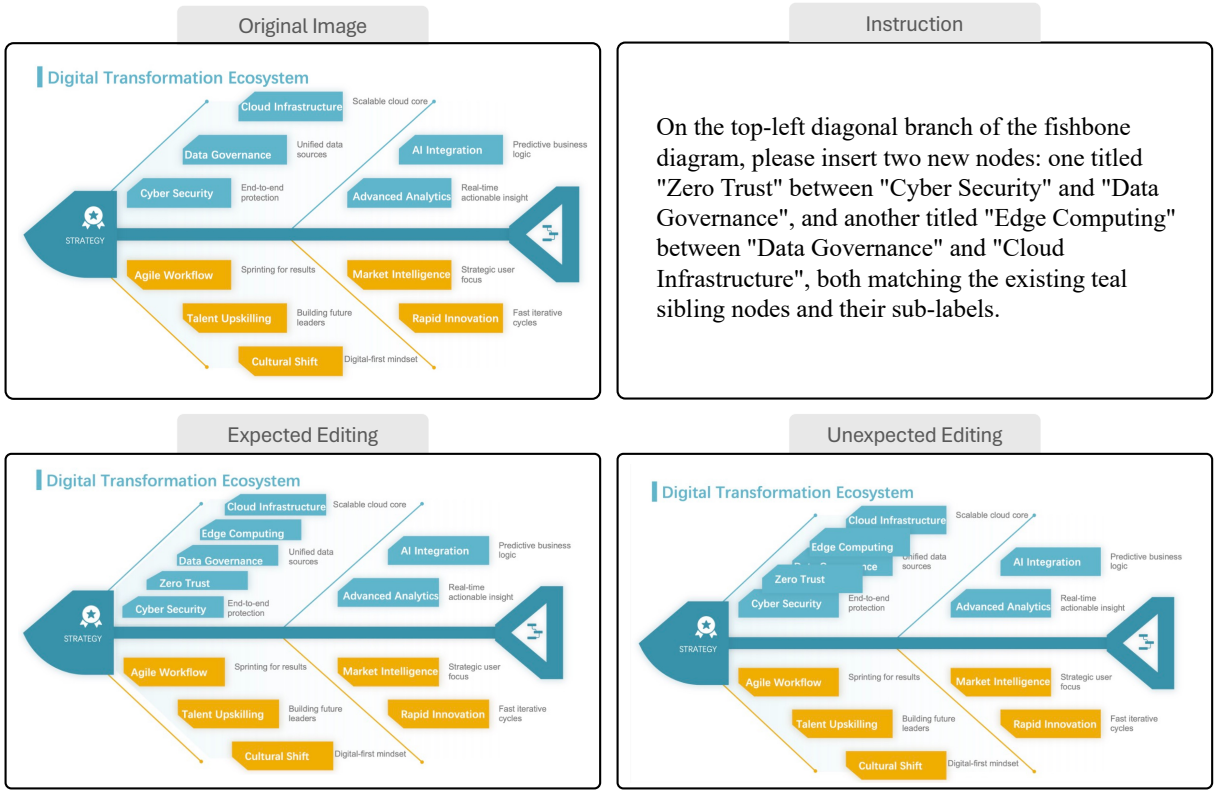


Figure 16: Example of the INSERT-ELEMENT task in INFOEDIT. The editor must insert new nodes into an existing logical structure. The expected edit places them at the specified positions and adjusts the surrounding structure, whereas the unexpected edit adds the nodes without sufficient reflow, resulting in an invalid local layout.

selection hints and must not appear in the final output. Fig. 19 shows an example. The two modules highlighted with the same label are the blocks to be exchanged.

D Details on Code Generation Baselines

We evaluate code generation as an alternative editing paradigm for INFOEDIT. Instead of directly modifying the rendered image, a model is given the underlying source representation of the infographic and asked to apply the same natural-language editing instruction at the code level. The modified source is then rendered back to an image, and the resulting PNG is evaluated with the same MLLM-as-judge protocol used for image-generation editors.

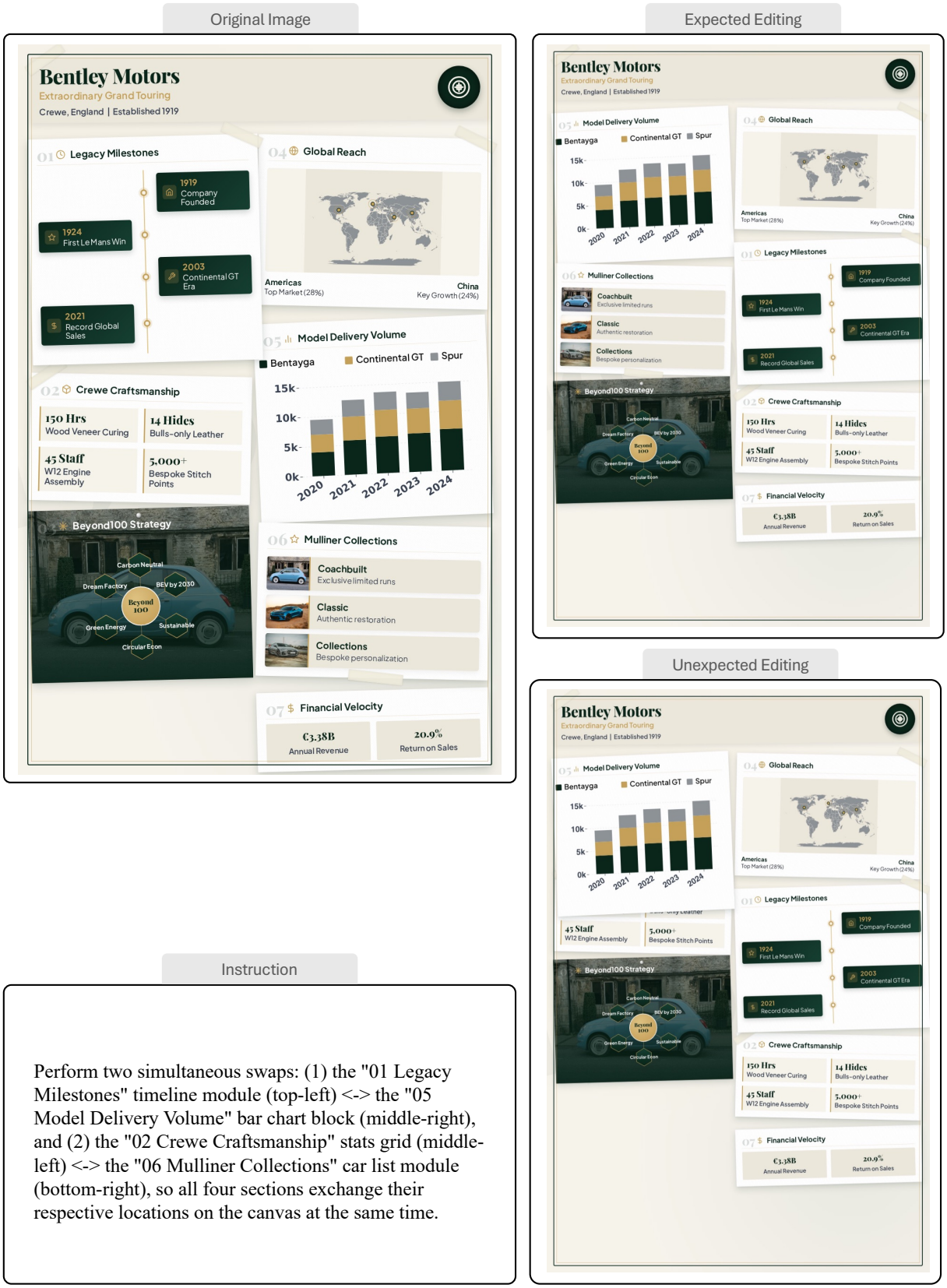
We consider two input settings. In the *code* setting, the model receives only the editable source and the instruction. In the *code + image* setting, the model additionally receives the original rendered infographic as visual grounding, so that it can better infer the visual layout, styling, and spatial relationships while still producing code as output.

The implementation differs slightly across the two source formats. For HTML infographics, the model rewrites the standalone HTML document, which is subsequently rendered to PNG using a browser. For PowerPoint infographics, we first expose a structured textual dump of the slide, including shapes, text, positions, and sizes; the model then produces a Python editing function using `python-pptx`, and the edited slide is rendered to PNG.

E Evaluation Reliability

E.1 Human Agreement across Output Models

We validate the binary overall verdict against expert human judgments across three output models spanning different capability levels. For each model, we use 800 outputs under the same protocol, with exactly 200 examples from each editing task. Three expert annotators independently assign a binary success or failure label, and their majority label is treated as the human reference. Table 7 reports agreement between Gemini-3.1-Pro and the human majority. The high agreement on HunyuanImage-3.0 should be interpreted together with its output distribution: most outputs are clear failures, which are straightforward for both the human annotators and the judge to identify.



Instruction

Perform two simultaneous swaps: (1) the "01 Legacy Milestones" timeline module (top-left) <-> the "05 Model Delivery Volume" bar chart block (middle-right), and (2) the "02 Crewe Craftsmanship" stats grid (middle-left) <-> the "06 Mulliner Collections" car list module (bottom-right), so all four sections exchange their respective locations on the canvas at the same time.

Figure 17: Example of the SWAP-BLOCK task in INFOEDIT. The instruction asks the editor to exchange multiple logical blocks on the canvas. The expected edit swaps the specified blocks as whole units while preserving their internal content and adapting the surrounding layout; the unexpected edit only partially satisfies the requested swaps or damages the original layout structure.

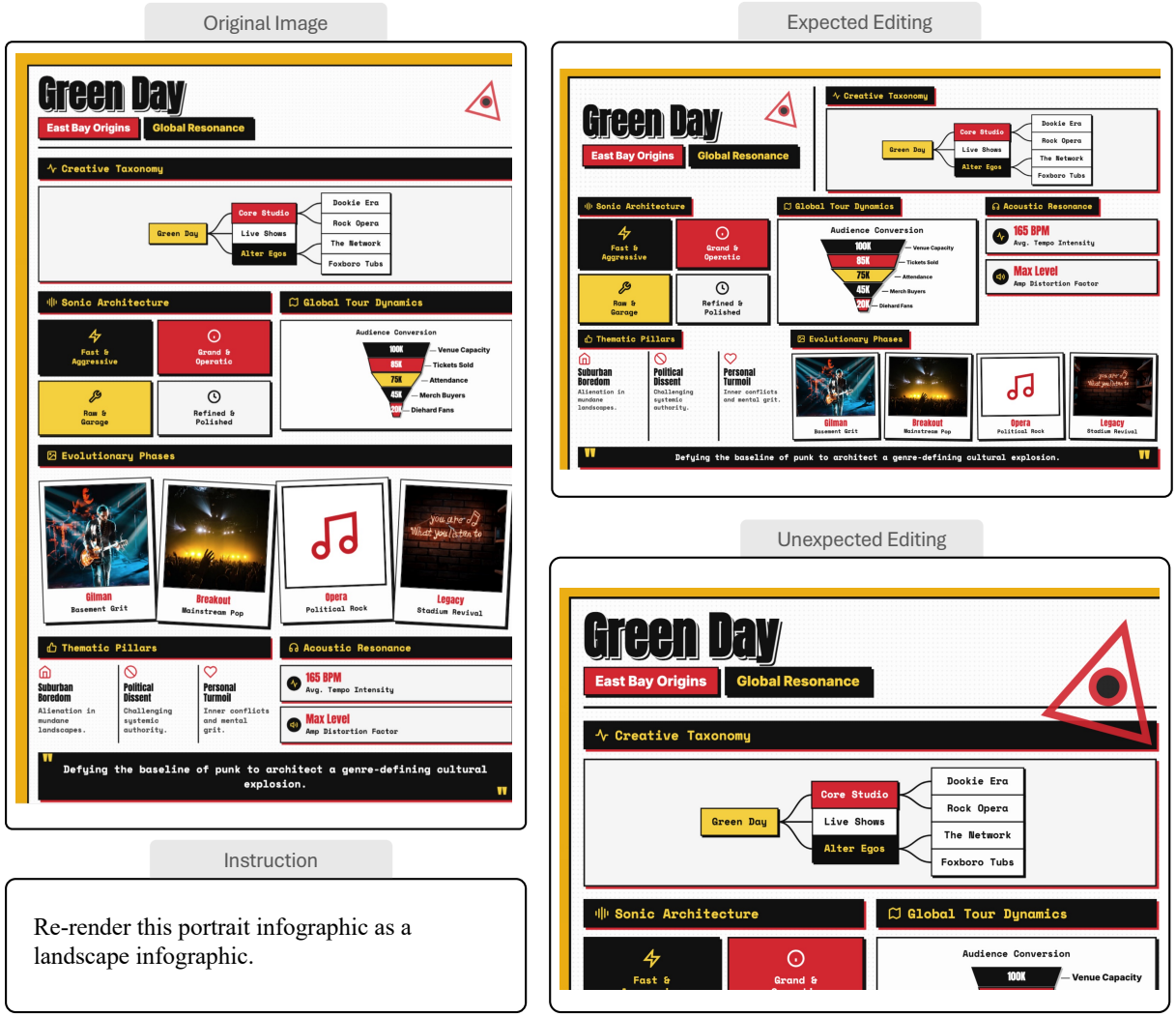


Figure 18: Example of the RESHAPE-CANVAS task in INFOEDIT. The instruction asks the editor to re-render a portrait infographic as a landscape infographic. The expected edit globally reflows the content to fit the target aspect ratio while preserving the original information and visual style; the unexpected edit changes the canvas shape but fails to perform a complete global re-layout.

Output model	Samples	Agreement
GPT-Image-2	800	0.920
Nanobanana-2	800	0.934
HunyuanImage-3.0	800	0.981

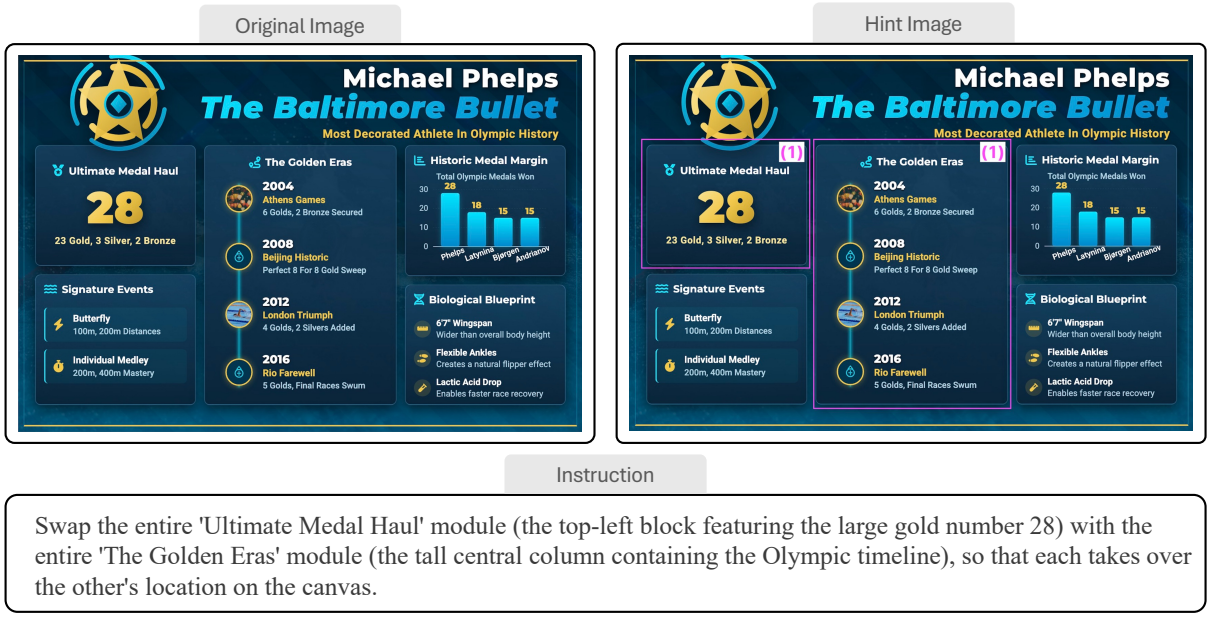
Table 7: Agreement between Gemini-3.1-Pro and the human-majority verdict across output models.

E.2 Stratified Agreement

For GPT-Image-2, inter-annotator agreement is 0.940. We additionally stratify judge–human agreement by task, requested-edit difficulty, and the judge's predicted outcome. As shown in Table 8, agreement remains high throughout. The difficulty analysis covers the three tasks with explicit magnitude axes; RESHAPE-CANVAS is excluded because it has no easy/medium/hard magnitude label. The outcome analysis is a post-hoc split by the judge's binary verdict.

Axis	Subset	Agreement
Task	EXPAND-TEXT	0.920
	INSERT-ELEMENT	0.905
	SWAP-BLOCK	0.925
	RESHAPE-CANVAS	0.930
Difficulty	Easy	0.943
	Medium	0.912
	Hard	0.890
Judge verdict	Success	0.913
	Failure	0.931

Table 8: Gemini-3.1-Pro agreement with the human majority on GPT-Image-2 outputs, stratified by task, difficulty, and predicted outcome.



Swap the entire 'Ultimate Medal Haul' module (the top-left block featuring the large gold number 28) with the entire 'The Golden Eras' module (the tall central column containing the Olympic timeline), so that each takes over the other's location on the canvas.

Figure 19: Interactive selection hint for SWAP-BLOCK. Matching numeric labels indicate the two blocks to be exchanged. The boxes are used only for target localization and should not appear in the edited output.

E.3 Alternative Judge

We test whether evaluation reliability depends on the choice of MLLM judge by re-evaluating the same model outputs with Claude-Opus-4.7 under the same rubric and prompt. Table 9 compares both judges against the human-majority labels. The two cross-vendor judges obtain similar agreement on outputs from both models.

Judge	GPT-Image-2	Nanobanana-2
Gemini-3.1-Pro	0.920	0.934
Claude-Opus-4.7	0.914	0.928

Table 9: Agreement with the human-majority verdict for two MLLM judges.

E.4 Repeated Evaluation

We also run Gemini-3.1-Pro five times on the same 800 outputs per model. Table 10 reports the mean agreement and standard deviation across runs. The small variation indicates that evaluation randomness does not affect the reported comparison.

Output model	Agreement over five runs
GPT-Image-2	0.924 ± 0.007
Nanobanana-2	0.936 ± 0.008

Table 10: Repeated Gemini-3.1-Pro evaluation against human-majority labels.

F Fine-Grained Results

F.1 Logical-Relation Families

We report per-family results on INSERT-ELEMENT, because every instruction in this task targets a specific logical-relation block and can be cleanly attributed to its host family. Table 11 shows that GPT-Image-2 leads across all families. Cycle is the hardest family for all three editors, consistent with the need to reroute and reclose a loop of connectors after insertion.

Family	GPT-Image-2	Nanobanana-Pro	Nanobanana-2
List	78.9	45.3	40.5
Matrix	75.2	37.0	34.5
Process	77.1	47.3	45.0
Picture	75.5	36.5	35.2
Cycle	49.3	20.6	31.6
Pyramid	72.7	49.5	46.5
Relationship	67.1	38.4	39.7
Hierarchy	78.7	63.8	48.9

Table 11: INSERT-ELEMENT SR by host logical-relation family. All values are percentages.

F.2 HTML versus PowerPoint Sources

Table 12 reports pixel-level editing results by the source format used to author the input infographic. Because pixel-level editors receive only the rendered image, their results are similar between HTML and PowerPoint, and their ranking is unchanged.

Table 13 gives the corresponding breakdown for code-level editing with both source code and ren-

Model	Source	EXPAND-TEXT			INSERT-ELEMENT			SWAP-BLOCK			RESHAPE-CANVAS			Avg.
		EC	CP	SR	EC	CP	SR	EC	CP	SR	EC	CP	SR	SR
GPT-Image-2	Overall	99.0	86.7	85.9	85.4	80.7	72.0	29.0	79.2	25.5	96.3	65.1	64.4	62.0
	HTML	99.1	86.3	85.5	85.8	80.3	71.9	29.5	78.9	25.9	96.6	64.6	64.0	61.8
	PPT	98.5	88.5	87.5	84.0	82.5	72.5	27.0	80.5	24.0	95.0	67.0	66.0	62.5
Nanobanana-2	Overall	85.1	58.6	52.6	65.3	49.4	39.0	30.0	72.1	24.0	92.0	36.4	36.3	38.0
	HTML	84.9	58.1	52.1	65.9	48.9	38.9	30.5	71.8	24.4	92.3	35.9	35.9	37.8
	PPT	86.0	60.5	54.5	63.0	51.5	39.5	28.0	73.5	22.5	91.0	38.5	38.0	38.6

Table 12: Pixel-level editing results by source format.

dered image as input. In this setting, both models perform substantially better on HTML than PowerPoint.

F.3 Canvas Transformation Types

We categorize RESHAPE-CANVAS instructions by their source-to-target aspect-ratio direction. Table 14 shows that all three models are weakest on landscape-to-portrait transformations.

G Additional Robustness Analyses

G.1 Color-Palette Sensitivity

The generated reference images in our construction pipeline are used only as color-palette cues. To test whether this choice systematically affects editor performance, we replace each infographic's palette while keeping its layout, structure, and content unchanged. The replacement palettes are suggested and applied by GLM-5.2, and annotators verify that only the color scheme changes. As shown in Table 15, the two evaluated models move in opposite directions by small amounts in average SR, and their ranking remains unchanged.

G.2 Multilingual Pilot

To examine whether the editable-source construction can support multilingual evaluation, we construct a preliminary Chinese version of the samples. We use GLM-5.2 to translate the infographic text directly in the editable source, re-render the sources, and translate the corresponding editing instructions. Annotators verify that the layout, structure, and logical relations remain unchanged. Table 16 shows lower average SR for both models, while their ranking remains unchanged. This rapid pilot is intended to demonstrate extensibility rather than serve as a complete multilingual evaluation.

H Ethical Statement

INFOEDIT is constructed for research purposes to evaluate infographic editing and reflow capabilities. The benchmark does not directly republish existing copyrighted infographic datasets as editable assets. Some samples mention real-world entities such as companies, products, universities or cities. We use only publicly available information, such as Wikipedia or official public pages, and manually review the content to avoid private, sensitive, defamatory, or misleading claims.

I Reproducibility

Upon publication, we will release INFOEDIT for research use. For reproducibility, Tab. 17 reports the exact model identifiers used in our experiments. We will release the code to reproduce the results. Following prior analyses of LLM decoding sensitivity (Shi et al., 2024), we set the temperature to 0.1 for all automated MLLM and LLM evaluators.

J Models

Tab. 17 summarizes the models used in our experiments and the corresponding inference interfaces. As discussed in Appx. B, many earlier image-editing models cannot reliably render complete and legible text, making them unsuitable for infographic editing where text is a core component. We therefore exclude such models and focus on recent baselines that meet this basic requirement. For proprietary models, we report the exact API model identifiers used in our calls; for open-weight models, we report the Hugging Face repositories from which the checkpoints are obtained.

K Error Analysis

We conducted an error mode analysis on GPT-Image-2, the best-performing model in our evaluation. As shown in Fig. 5, we present the distribu-

Model	Source	EXPAND-TEXT			INSERT-ELEMENT			SWAP-BLOCK			RESHAPE-CANVAS			Avg.
		EC	CP	SR	EC	CP	SR	EC	CP	SR	EC	CP	SR	SR
Gemini-3.1-Pro	Overall	88.2	72.0	68.5	85.3	72.2	71.0	69.0	78.4	60.2	71.8	47.2	46.8	61.6
	HTML	96.5	79.6	77.6	92.6	80.9	79.5	75.6	85.4	68.0	75.9	53.4	52.9	69.5
	PPT	55.0	41.5	32.0	56.0	37.5	37.0	42.5	50.5	29.0	55.5	22.5	22.5	30.1
Gemini-3.5-Flash	Overall	85.1	69.5	65.8	85.4	67.2	63.2	60.9	79.8	56.8	71.2	46.6	43.1	57.2
	HTML	93.1	76.9	74.5	92.8	75.6	70.8	66.8	86.9	64.1	75.1	52.8	48.8	64.5
	PPT	53.0	40.0	31.0	56.0	33.5	33.0	37.5	51.5	27.5	55.5	22.0	20.5	28.0

Table 13: Code-level editing with code and image input, broken down by source format.

Model	P→L	L→P	S→L	S→P
GPT-Image-2	69.1	59.4	72.5	67.5
Nanobanana-Pro	36.7	31.8	65.0	60.0
Nanobanana-2	41.2	32.0	35.0	37.5

Table 14: RESHAPE-CANVAS SR by transformation direction. P, L, and S denote portrait, landscape, and square.

Model	Setting	EXPAND-TEXT			INSERT-ELEMENT			SWAP-BLOCK			RESHAPE-CANVAS			Avg.
		EC	CP	SR	EC	CP	SR	EC	CP	SR	EC	CP	SR	SR
GPT-Image-2	Original	99.0	86.7	85.9	85.4	80.7	72.0	29.0	79.2	25.5	96.3	65.1	64.4	62.0
	Color-Change	98.2	86.2	84.8	84.2	81.3	74.2	31.0	81.2	27.2	98.2	67.1	66.8	63.3
Nanobanana-2	Original	85.1	58.6	52.6	65.3	49.4	39.0	30.0	72.1	24.0	92.0	36.4	36.3	38.0
	Color-Change	86.2	57.2	51.7	63.2	49.0	37.2	32.1	73.2	24.8	92.3	35.2	34.9	37.2

Table 15: Color-palette control. All values are percentages.

Model	Setting	EXPAND-TEXT			INSERT-ELEMENT			SWAP-BLOCK			RESHAPE-CANVAS			Avg.
		EC	CP	SR	EC	CP	SR	EC	CP	SR	EC	CP	SR	SR
GPT-Image-2	Original	99.0	86.7	85.9	85.4	80.7	72.0	29.0	79.2	25.5	96.3	65.1	64.4	62.0
	Chinese	93.0	84.5	82.1	80.4	74.2	67.2	25.2	75.2	24.1	93.2	61.2	57.2	57.7
Nanobanana-2	Original	85.1	58.6	52.6	65.3	49.4	39.0	30.0	72.1	24.0	92.0	36.4	36.3	38.0
	Chinese	86.2	62.1	56.4	42.5	46.2	25.3	22.6	67.2	15.2	86.2	28.2	26.3	30.8

Table 16: Preliminary Chinese translation pilot. All values are percentages.

tion of different failure modes per task, including Incomplete Content, Misplacement, Element Loss, Occlusion, Text Rewritten, Style Change, Structural Break, Content Altered, and No Re-layout. We define each error mode as follows:

- **Incomplete Content**: The target element of an add or text-expand operation is not fully rendered. Its content, such as icons, labels, numbers, or expanded body text, is truncated, clipped, or overflows the container.
- **Misplacement**: The target element is misplaced. For Insert-Element, the new element does not reference the specified relative anchor. For Swap-Block, the two blocks are not correctly exchanged.
- **Element Loss**: One or more original elements no longer fully exist on the canvas. They are either deleted outright, pushed off the visible area, or cropped at the edges during reflow.
- **Occlusion**: An original element is partially or fully covered by another element due to incorrect ordering or layout collisions.
- **Text Rewritten**: The text of non-target elements are altered or rewritten.
- **Style Change**: The visual styling of original elements, such as icons, logos, or color palette,

Model Name	Type	Inference Interface
Proprietary models		
GPT-Image-2	API	gpt-image-2
GPT-Image-1.5	API	gpt-image-1.5
Nanobanana-Pro	API	gemini-3-pro-image-preview
Nanobanana-2	API	gemini-3.1-flash-image-preview
Nanobanana	API	gemini-2.5-flash-image
Seedream-5.0-Lite	API	doubao-seedream-5-0-260128
Gemini-3.1-Pro	API	gemini-3.1-pro-preview
Gemini-3.5-Flash	API	gemini-3.5-flash
Gemini-3.1-Flash	API	gemini-3.1-flash-lite-preview
Claude-Opus-4.7	API	claude-opus-4-7
GPT-5.4	API	gpt-5.4-2026-03-05
Open-weight models		
HunyuanImage-3.0-Instruct	Hugging Face	https://huggingface.co/tencent/HunyuanImage-3.0-Instruct
Qwen-Image-Edit	Hugging Face	https://huggingface.co/Qwen/Qwen-Image-Edit-2511
FLUX.2-klein-base-9B	Hugging Face	https://huggingface.co/black-forest-labs/FLUX.2-klein-base-9B

Table 17: Models used in our experiments and their inference interfaces. For proprietary models, we report the API model identifier used in our calls; for open-weight models, we report the corresponding Hugging Face repository.

is redesigned rather than preserved.

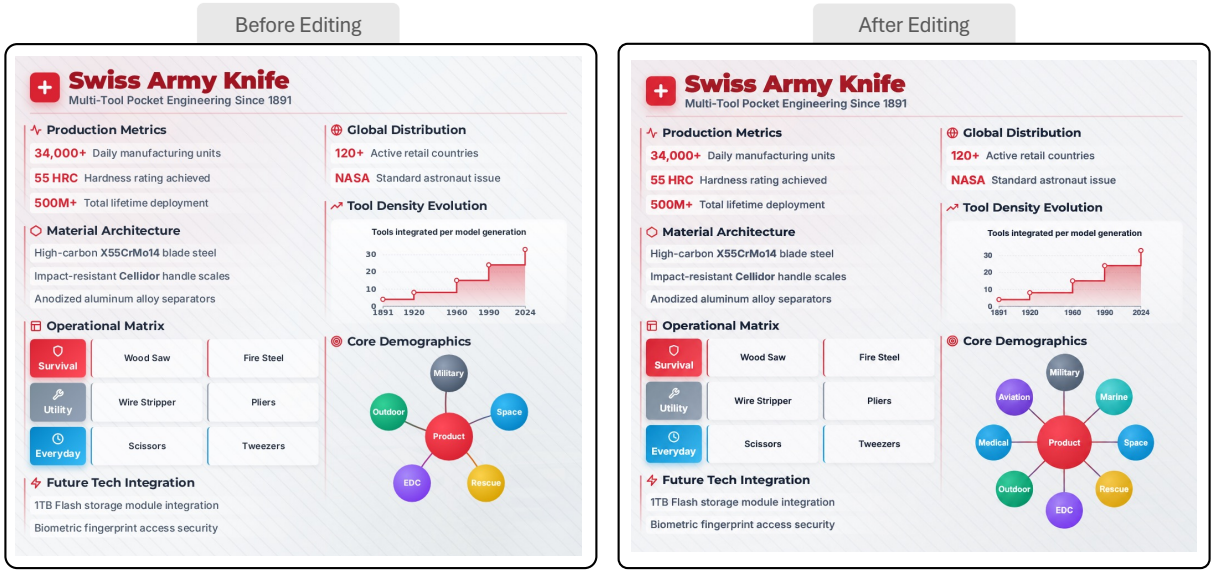
- **Structural Break**: Structural relationships among elements in the canvas are broken, manifesting as redirected connectors in process flows, broken closure in cyclic diagrams, or skipped indices in numbered sequences.
- **Content Altered**: The internal content (titles, icons, text, numbers, background, or visual style) of the swapped blocks is modified during editing.
- **No Re-layout**: The model fails to genuinely re-arrange the infographic for the new canvas shape, instead treating it as a flat image.

We further visualize some representative qualitative samples to illustrate how these failure modes manifest in practice, including those for Incomplete Content in Figure 22, Misplacement in Figure 20, Element Loss in Figures 23, 25, Style Change in Figures 21, 24, 26, 29, Structural Break in Figures 27, 28, and Text Rewritten in Figure 30.

L Prompts

L.1 MLLM-as-a-Judge Evaluation Prompts

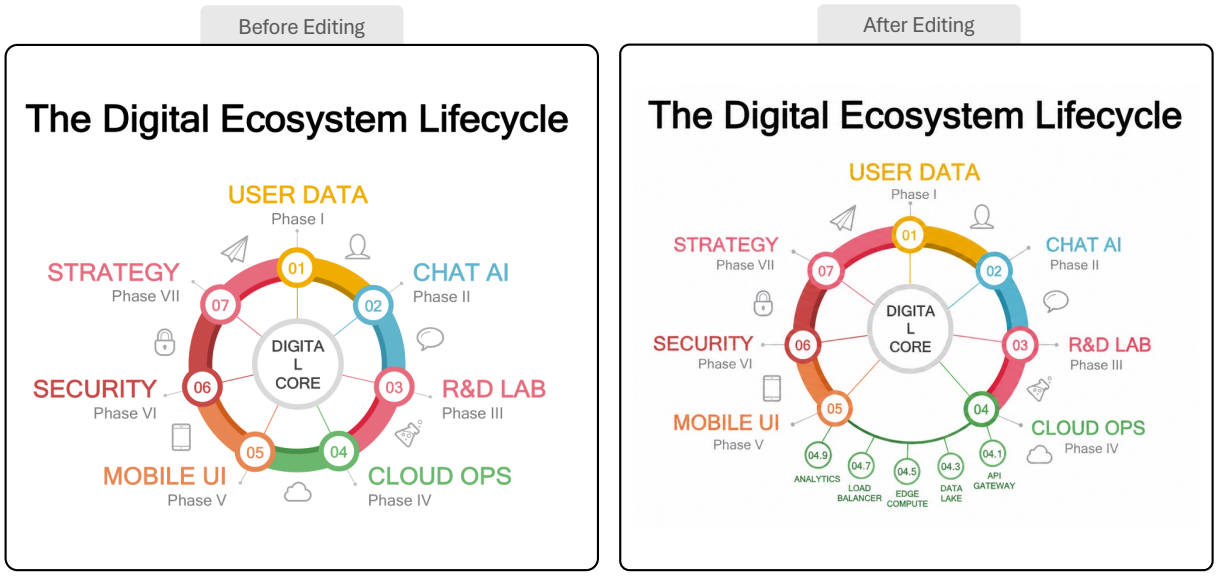
The prompts we use for MLLM-as-a-Judge evaluation are shown in Fig. 31, 32, 33 and 34.



Instruction

Please insert three new outer circles "Aviation", "Medical", and "Marine" between "Military" and "Outdoor" in the Core Demographics radial diagram, all matching the existing outer circles.

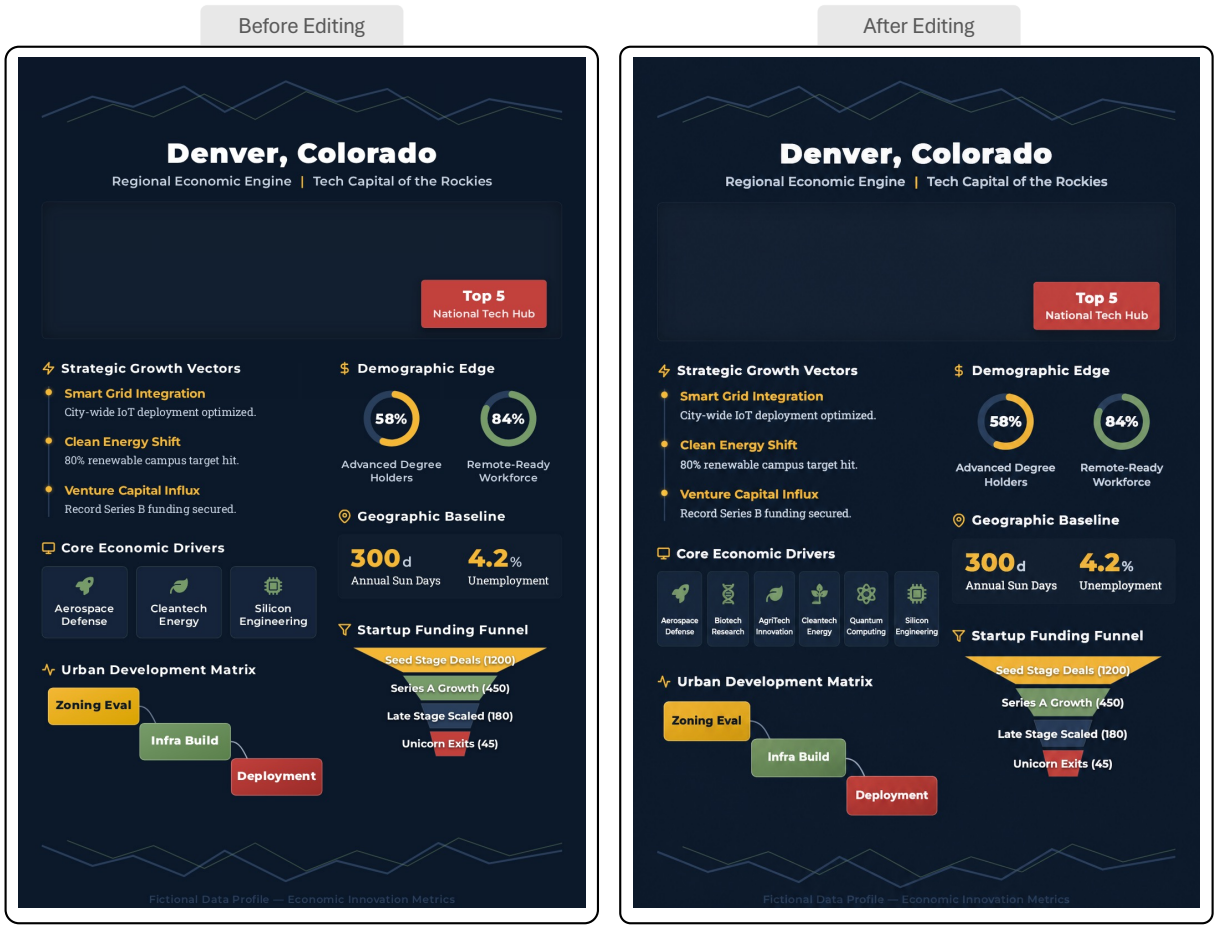
Figure 20: Misplacement after Insert-Element. The three new circles were inserted into the Core Demographics radial diagram, but “Marine” is placed in the wrong position. It is incorrectly positioned on the opposite side of the diagram between “Military” and “Space”.



Instruction

Please insert five new segments in the segmented cycle between "CLOUD OPS" (04) and "MOBILE UI" (05): 04.1 "API GATEWAY", 04.3 "DATA LAKE", 04.5 "EDGE COMPUTE", 04.7 "LOAD BALANCER", and 04.9 "ANALYTICS", all matching the existing segments.

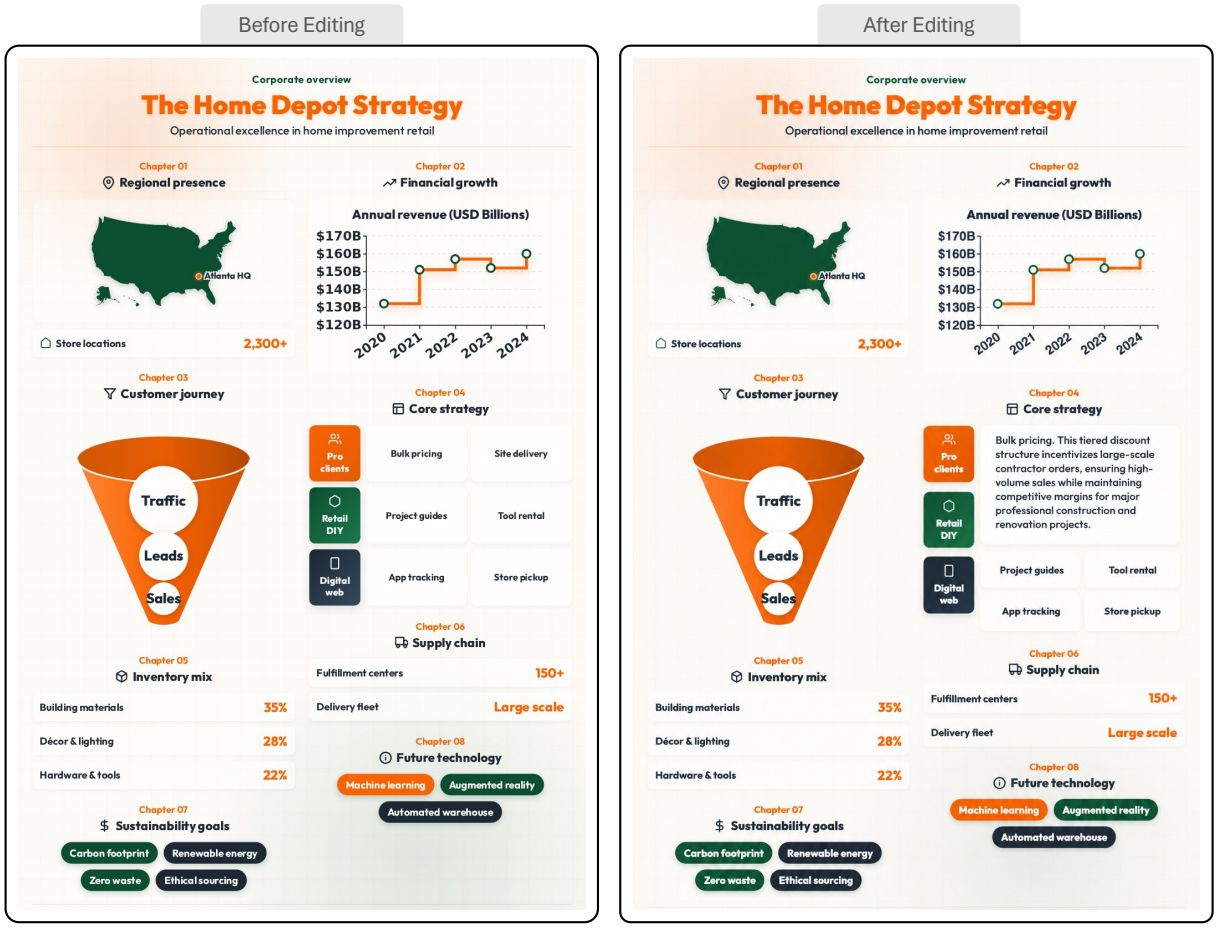
Figure 21: Style change after Insert-Element. The five new segments were inserted between “Cloud Ops” and “Mobile UI”, but they do not match the style of the existing segments. Instead of being integrated as arc segments within the main ring, they are rendered as smaller circular nodes branching outward below the cycle.



Instruction

Please insert four new cards "Biotech Research" and "AgriTech Innovation" between "Aerospace Defense" and "Cleantech Energy", and "Quantum Computing" and "Advanced Manufacturing" between "Cleantech Energy" and "Silicon Engineering" in the Core Economic Drivers horizontal layout, all matching the existing cards.

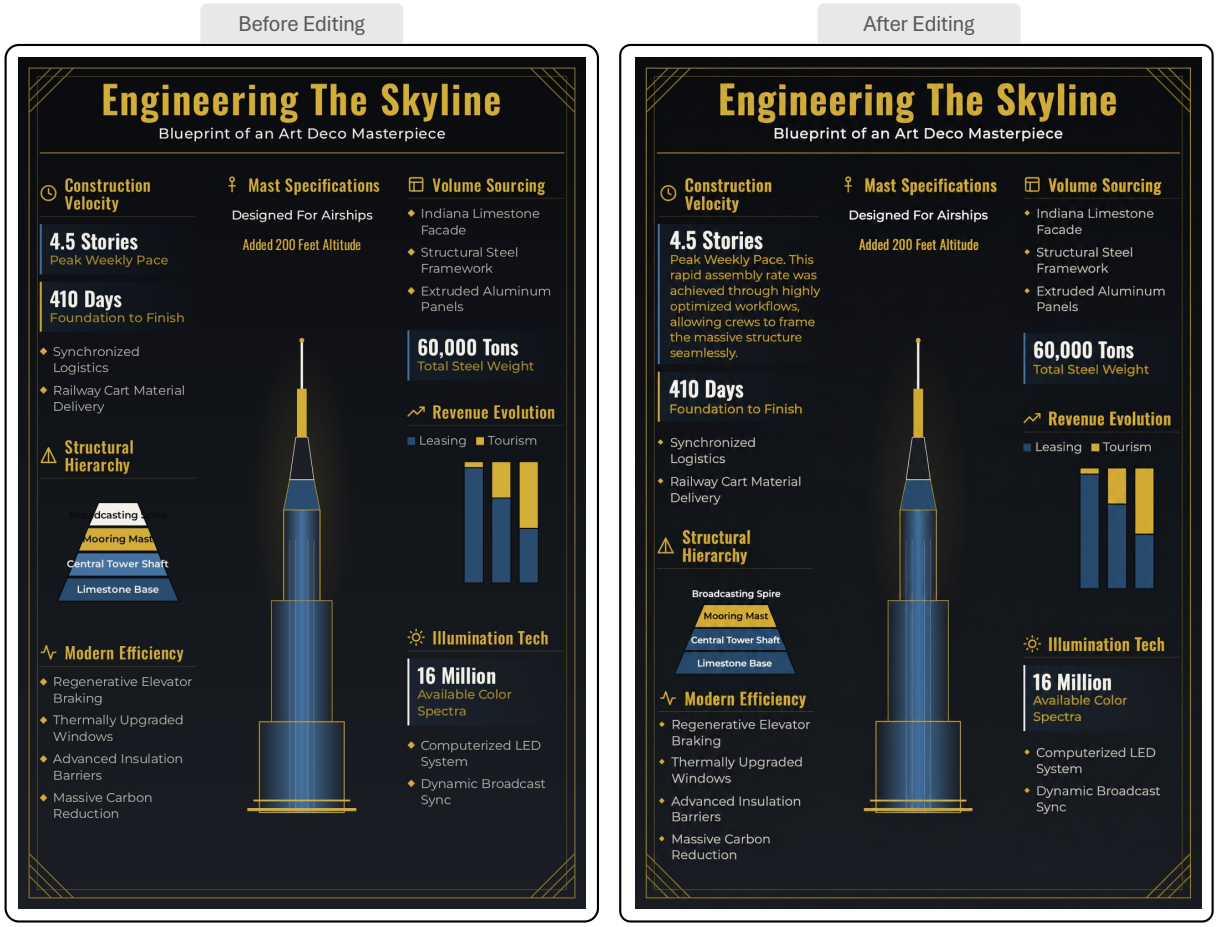
Figure 22: Incomplete content after Insert-Element. The Core Economic Drivers row is incomplete. There are only three of the four requested cards were inserted, while “Advanced Manufacturing” is missing between “Cleantech Energy” and “Silicon Engineering”.



Instruction

Please extend the text inside the top-left white rectangular card in the Chapter 04 "Core strategy" grid. The current text reads "Bulk pricing". Replace it with this longer version on the same topic: "Bulk pricing. This tiered discount structure incentivizes large-scale contractor orders, ensuring high-volume sales while maintaining competitive margins for major professional construction and renovation projects."

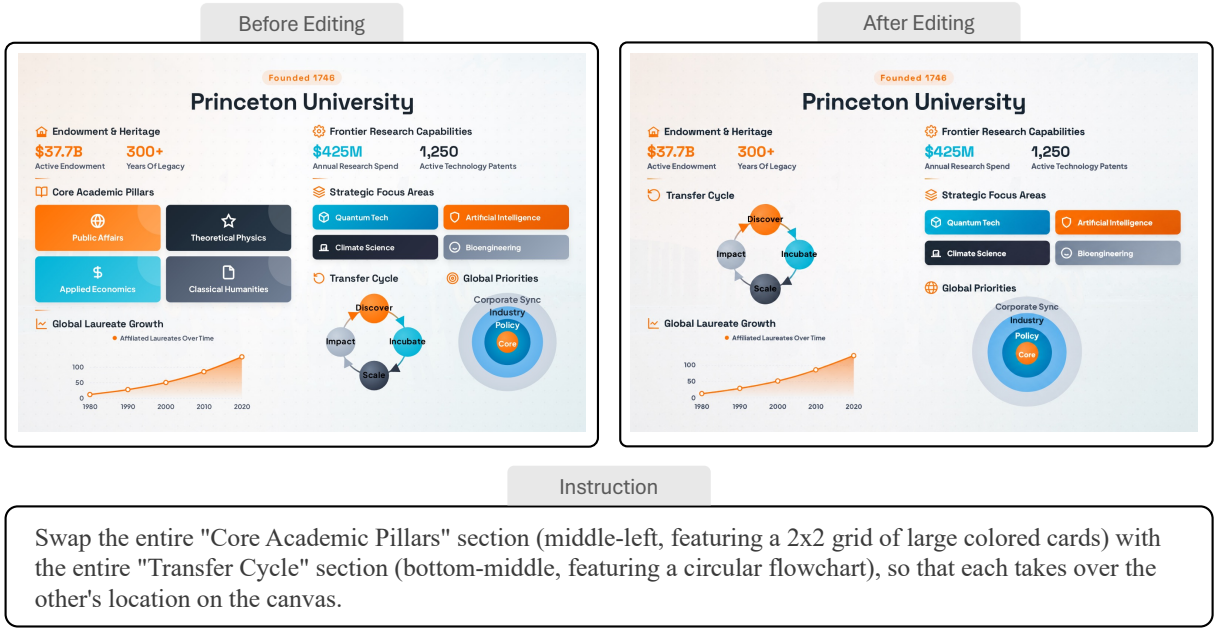
Figure 23: Element loss after Expand-Text. The “Bulk pricing” was successfully expanded with the longer description, but the adjacent “Site delivery” card in the top-right of the Chapter 04 “Core strategy” grid has been removed. It makes the grid incomplete and disrupts the original 3×2 layout.



Instruction

Please extend the caption under '4.5 Stories' in the 'Construction Velocity' section on the left. The current body reads 'Peak Weekly Pace'. Replace it with this longer version on the same topic (keep the numeric title unchanged): 'Peak Weekly Pace. This rapid assembly rate was achieved through highly optimized workflows, allowing crews to frame the massive structure seamlessly.'

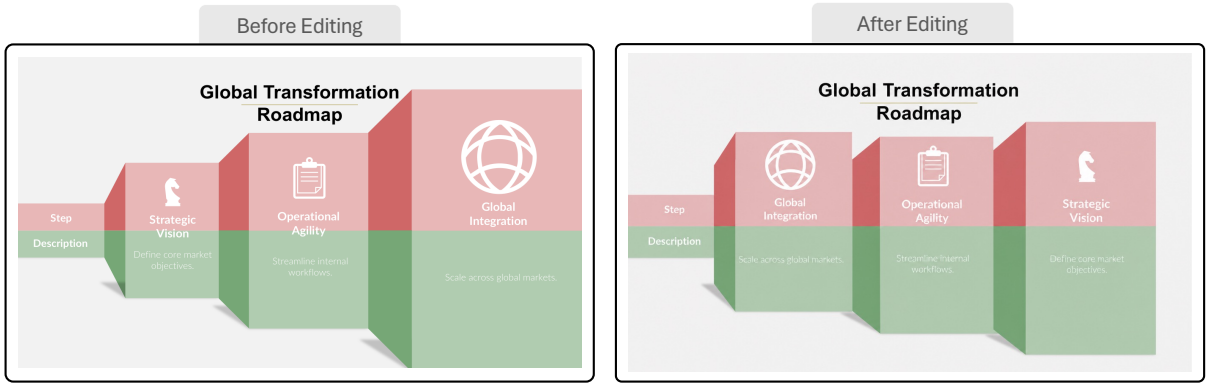
Figure 24: Style change after Expand-Text. The “4.5 Stories” caption was correctly expanded, but the “Broadcasting Spire” label at the top of “Structural Hierarchy” pyramid has changed from black text on a light background to white text. This breaks visual consistency with the original styling.



Instruction

Swap the entire "Core Academic Pillars" section (middle-left, featuring a 2x2 grid of large colored cards) with the entire "Transfer Cycle" section (bottom-middle, featuring a circular flowchart), so that each takes over the other's location on the canvas.

Figure 25: Element loss after Swap-Block. The swap was only partially executed. The “Transfer Cycle” was moved to the middle-left position previously occupied by “Core Academic Pillars” but the “Core Academic Pillars” is missing entirely instead of being relocated to the bottom-middle.



Instruction

Swap the entire 'Strategic Vision' vertical module (the leftmost main roadmap step, featuring the knight icon) with the entire 'Global Integration' vertical module (the rightmost, largest roadmap step, featuring the globe icon).

Figure 26: Style change after Swap-Block. The “Strategic” and “Global Integration” modules were swapped in position, but their original styling was not preserved. The ascending size progression of the roadmap steps is lost.

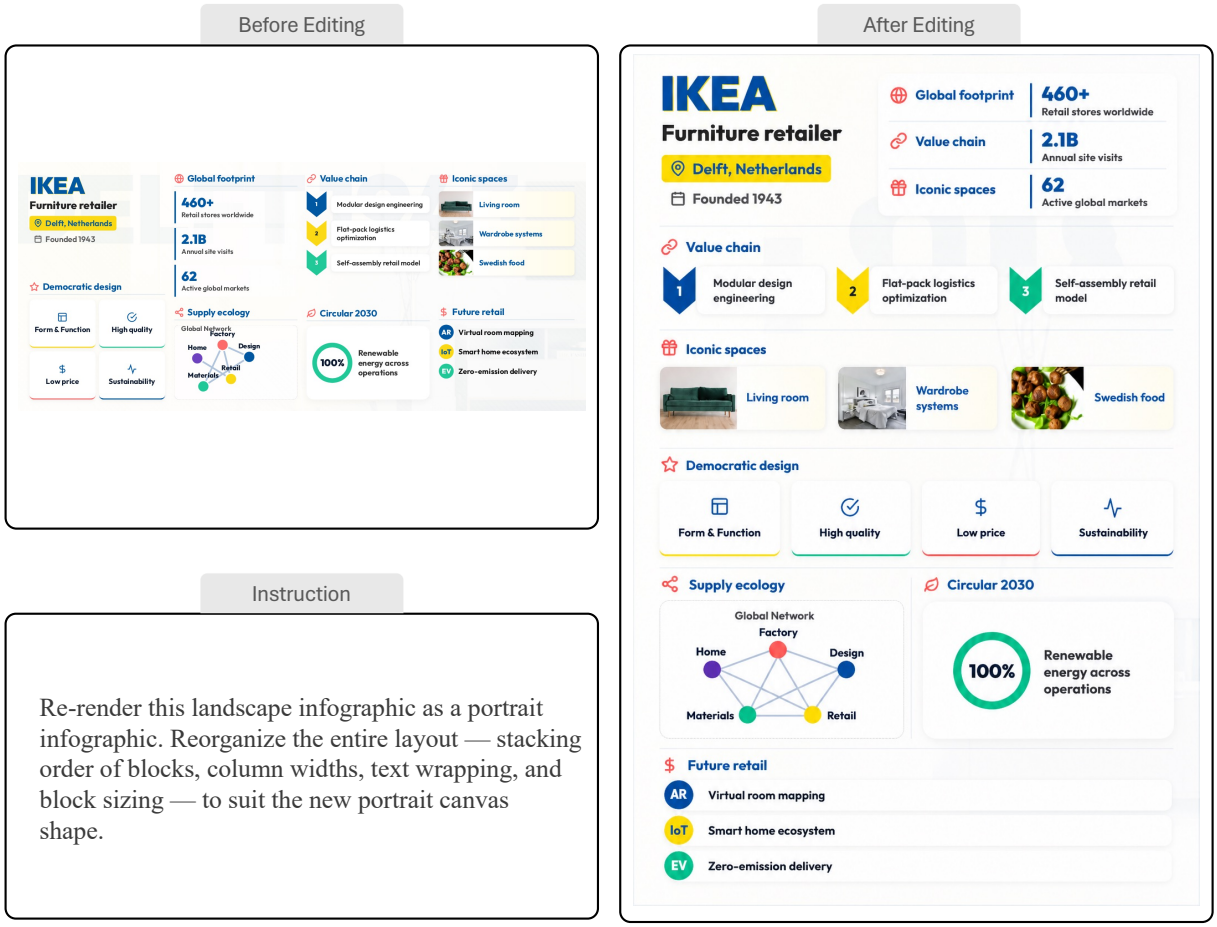


Figure 27: Structural break after Reshape-Canvas. The “Supply ecology” network diagram has been structurally altered. The connection topology between the nodes no longer matches the original.

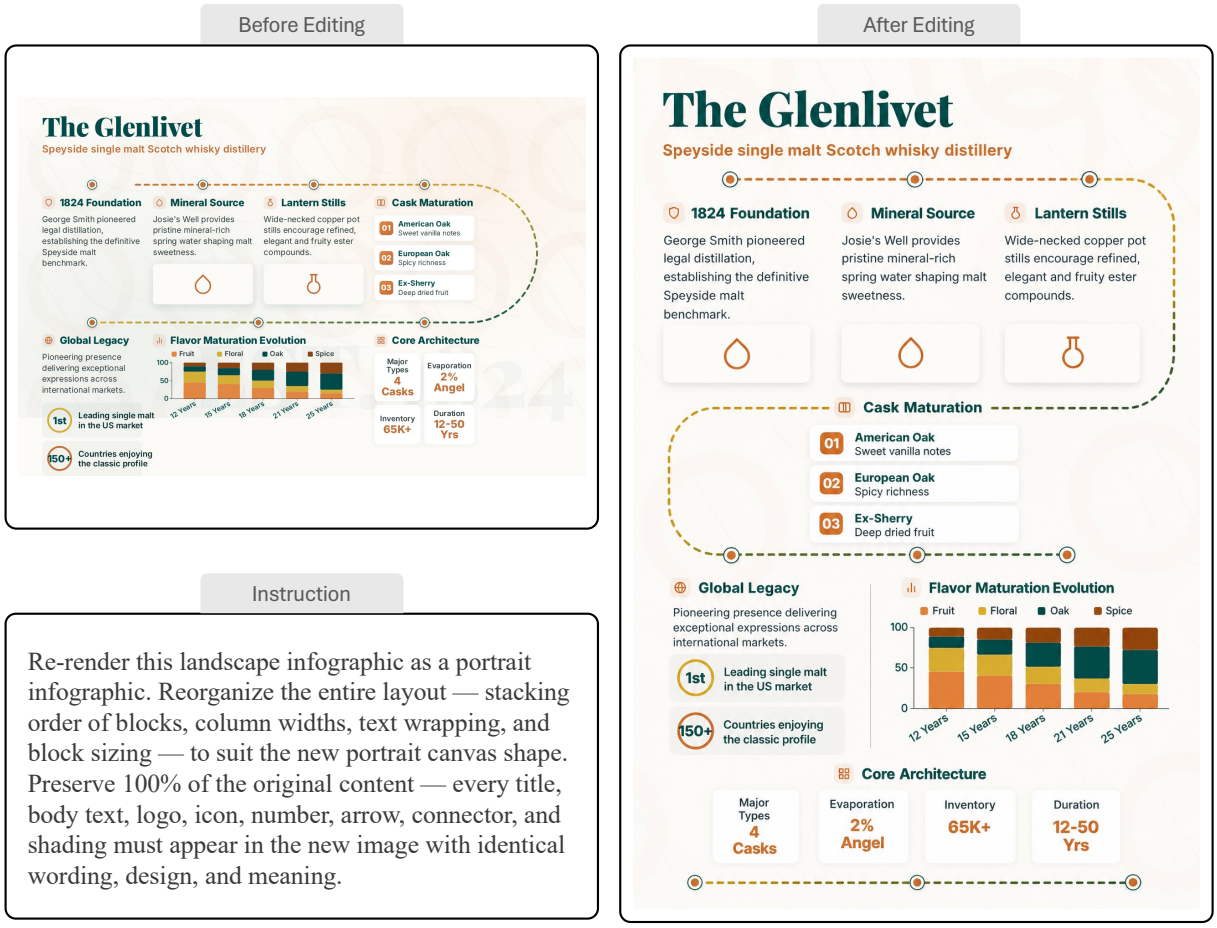


Figure 28: Structural break after Reshape-Canvas. All original content was preserved when rendered as a portrait infographic, but the block ordering does not follow a coherent reading flow.

Figure 29: Style change after Reshape-Canvas. The infographic was re-rendered into a landscape canvas with all original content preserved, but the card styling has been altered.

Instruction

Perform two simultaneous swaps: (1) the "01 Legacy Milestones" timeline module (top-left) <-> the "05 Model Delivery Volume" bar chart block (middle-right), and (2) the "02 Crewe Craftsmanship" stats grid (middle-left) <-> the "06 Mulliner Collections" car list module (bottom-right), so all four sections exchange their respective locations on the canvas at the same time.

Figure 30: Text rewritten after Swap-Block using Qwen-Image-Edit. The model partially changes the layout, but many original texts, labels, and chart annotations are rewritten or corrupted instead of being preserved. This shows that the model fails to maintain non-target content.

Prompt for MLLM-as-a-Judge Evaluation (Expand-Text)

Role: You are a strict image editing quality evaluator.

Objective: Given an original image, an editing instruction that asks to EXTEND the body text of ONE specific element by adding one or more additional sentences on the same topic (moderate extension, same topic, just enough to overflow the original slot), and the edited image, determine whether the extension has been successfully implemented. Your judgement must be binary: either success or failure, with no middle ground.

[Evaluation Criteria (BOTH must be met for success)]:

1. **Edit Compliance**: The element whose body text gets rewritten in the edited image must be exactly the one named in the instruction — same title, same anchor location, same numbering — and the target's title, label, and number must NOT change. AND the rewritten body text must be noticeably longer than the original — adding several additional short sentences added on the same topic — but NOT tripled, quadrupled, or turned into a dense paragraph. The extended text must be fully and legibly rendered inside the canvas — no truncation, no overflow beyond the canvas, no overlap with neighboring elements, no cropped lines, no text rendered outside its container. Targeting the wrong element, changing multiple elements' texts, trivial length changes, dense paragraph-length rewrites, off-topic rewrites, or visible overflow/clipping all count as failure.

2. **Content Preservation**: Every original element other than the target whose body text is being extended must remain intact and content-unchanged. Repositioning, resizing, proportional rescaling, and font-size adjustments (e.g. shrinking text to fit a tightened cell, and re-wrapping of non-target elements) are ALLOWED — these are normal reflow behaviors, not failures. What counts as failure: deletion of an original element, alteration of an element's text / icon / number / shading, occlusion that hides content, cropping that pushes content off-canvas, or shrinking so aggressive that the text becomes illegible. Scrutinize neighbors immediately adjacent to the target most aggressively — they are the most common victims of occlusion, displacement, or being pushed off-canvas when text is extended.

[Editing Instruction]:

{instruction}

Please compare the original image (first image) with the edited image (second image) carefully. Pay special attention to whether the named element's body text has been moderately extended on-topic without truncation or overflow, and whether all other elements remain intact and fully visible — especially the neighbors of the target.

Output in the following JSON format strictly:

```
{
  "reasoning": "<step-by-step analysis: 1) verify the target element identity and
that its rewritten body text is moderately longer (several short sentences added,
on the same topic, not a dense paragraph) and fully rendered without truncation,
overflow, or overlap; 2) verify all other content is preserved, with neighbors of
the target especially checked for occlusion, displacement, or being pushed off-
canvas>",
  "edit_compliance": "<success/failure>",
  "content_preservation": "<success/failure>",
  "overall_judgement": "<success/failure>"
}
```

Figure 31: Prompt for MLLM-as-a-Judge evaluation on EXPAND-TEXT editing tasks.

Prompt for MLLM-as-a-Judge Evaluation (Insert-Element)

Role: You are a strict image editing quality evaluator.

Objective: Given an original image, an editing instruction that asks to INSERT one or more new elements at a specified location, and the edited image, determine whether the insertion has been successfully implemented. Your judgement must be binary: either success or failure, with no middle ground.

[Evaluation Criteria (BOTH must be met for success)]:

1. **Edit Compliance**: The new element(s) must be inserted at the exact position specified in the instruction — for example, if the instruction says "insert between A and B", the new element must appear precisely between A and B, not above A, not below B, not in any other location. AND the specified element(s) must be fully and correctly added as described. Any deviation in placement, or missing / incomplete / incorrect additions, count as failure.
2. **Content Preservation**: Every original element must remain intact and content-unchanged. Repositioning, resizing, proportional rescaling, and font-size adjustments (e.g. shrinking text to fit a smaller cell after reflow) are ALLOWED — these are normal reflow behaviors, not failures. What counts as failure: deletion of an original element, alteration of an element's text / icon / number / shading (subject to the Visual-Ambiguity Caveat declared at the top of this prompt), occlusion that hides content, cropping that pushes content off-canvas, or shrinking so aggressive that the text becomes illegible.

[Editing Instruction]:

{instruction}

Please compare the original image (first image) with the edited image (second image) carefully. Pay special attention to whether the new element is placed at the exact position specified in the instruction with the specified content.

Output in the following JSON format strictly:

```
{
  "reasoning": "<step-by-step analysis: 1) identify what the instruction asks to
add and where exactly, then verify the new element exists in the edited image at
the specified position with the specified content; 2) verify all original content
is preserved>",
  "edit_compliance": "<success/failure>",
  "content_preservation": "<success/failure>",
  "overall_judgement": "<success/failure>"
}
```

Figure 32: Prompt for MLLM-as-a-Judge evaluation on INSERT-ELEMENT editing tasks.

Prompt for MLLM-as-a-Judge Evaluation (Swap-Block)

Role: You are a strict image editing quality evaluator.

Objective: Given an original image, an editing instruction that asks to SWAP ENTIRE blocks/modules on the canvas, and the edited image, determine whether the swap has been successfully implemented. Your judgement must be binary: either success or failure, with no middle ground.

[Evaluation Criteria (BOTH must be met for success)]:

1. **Edit Compliance**: The named blocks/modules must have exchanged their locations on the canvas as whole units — Block A must now sit at Block B's original location, and vice versa. AND each block must carry ALL of its internal contents — titles, icons, text, numbers, shading — unchanged into its new location. Any partial move, single-block shift, location other than the swap, or any block that loses / gains / alters its internal content during the swap counts as failure.
2. **Content Preservation**: Every original element other than the named blocks (which exchange locations) must remain intact and content-unchanged. Repositioning, resizing, proportional rescaling, and font-size adjustments (e.g. shrinking text on surrounding blocks to fit after the swap) are ALLOWED — these are normal reflow behaviors, not failures. What counts as failure: deletion of an original element, alteration of an element's text / icon / number / shading (subject to the Visual-Ambiguity Caveat declared at the top of this prompt), occlusion that hides content, cropping that pushes content off-canvas, or shrinking so aggressive that the text becomes illegible.

[Editing Instruction]:

{instruction}

Please compare the original image (first image) with the edited image (second image) carefully. Pay special attention to whether exactly the named blocks have exchanged locations while carrying their internal contents intact.

Output in the following JSON format strictly:

```
{
  "reasoning": "<step-by-step analysis: 1) identify the blocks/modules named in
the instruction and verify each block is now at the other's original location
with its internal content unchanged; 2) verify untouched blocks are preserved>",
  "edit_compliance": "<success/failure>",
  "content_preservation": "<success/failure>",
  "overall_judgement": "<success/failure>"
}
```

Figure 33: Prompt for MLLM-as-a-Judge evaluation on SWAP-BLOCK editing tasks.

Prompt for MLLM-as-a-Judge Evaluation (Reshape-Canvas)

Role: You are a strict image editing quality evaluator.

Objective: Given an original image, an editing instruction that asks to RE-RENDER the infographic in a new canvas aspect ratio (e.g. landscape ↔ portrait), and the edited image, determine whether the re-layout has been successfully implemented. Your judgement must be binary: either success or failure, with no middle ground.

[Source-Ratio-Dependent Track Selection]:

Before applying the criteria, estimate the ORIGINAL image's aspect ratio and pick the evaluation track:

- **LENIENT track** — original is approximately 1:1 (square): since forcing a square composition into a landscape / portrait frame inherently requires some geometric accommodation, mechanical uniform scaling, stretching, letterbox padding, or centered cropping are ACCEPTABLE ways to reach the target ratio. A full block-level re-layout is NOT required. Content Preservation (criterion 2) is still judged strictly.
- **STRICT track** — original is NOT 1:1 (already landscape or portrait): both criteria below must be met as written. Mechanical stretching, squeezing, cropping, or padding in place of a genuine re-layout is a failure.

[Evaluation Criteria (BOTH must be met for success)]:

1. **Edit Compliance**: The edited image's canvas must clearly match the TARGET aspect ratio specified in the instruction (e.g. 9:16 portrait, 16:9 landscape, 1:1 square) — a noticeable deviation from that shape, or leaving the canvas at the original aspect ratio, is a failure. AND every original element must remain within the edited canvas and fully readable — no text pushed off-canvas, no block missing, no content clipped by the frame edge.
   - STRICT track: in addition, blocks must be genuinely re-arranged — stacked, re-wrapped, resized, or reordered — to suit the new canvas shape; simply cropping or padding the original image to the new shape (without re-laying-out the blocks), or mechanically squeezed / stretched arrangements that cause overlaps, cropped text, or empty voids, are a failure.
   - LENIENT track: uniform scaling, stretching, letterbox padding, or centered cropping of the original 1:1 composition is fine even if it distorts geometry (e.g., circles become ovals, photos look widened). Only score failure here if content is actually lost or clipped off the frame.
2. **Content Preservation**: Every original element must remain intact and content-unchanged. Repositioning, resizing, proportional rescaling, and font-size adjustments (e.g. blocks re-stacked, re-flowed, or re-wrapped to fit the new canvas shape) are ALLOWED — these are normal reflow behaviors, not failures. What counts as failure: deletion of an original element, alteration of an element's text / icon / number / shading, occlusion that hides content, cropping that pushes content off-canvas, or shrinking so aggressive that the text becomes illegible. This criterion is judged strictly regardless of which track (LENIENT/STRICT) was selected for criterion 1.

[Editing Instruction]:

{instruction}

Please compare the original image (first image) with the edited image (second image) carefully. First decide which track applies based on the original's aspect ratio, then apply criteria (a) Edit Compliance — canvas matches the target ratio AND every block fits / is readable in the new canvas under the selected track, (b) all original content preserved.

Output in the following JSON format strictly:

```
{
  "reasoning": "<step-by-step analysis: 1) estimate the ORIGINAL image's aspect
ratio and declare whether the LENIENT (source ≈ 1:1) or STRICT (source non-1:1)
track applies; 2) verify the edited image's canvas matches the target aspect
ratio AND that every original block fits and remains readable in the new canvas —
STRICT requires genuine re-arrangement, LENIENT accepts stretching / padding /
cropping; 3) verify every original block's content is present and unmodified
(strict regardless of track)>",
  "edit_compliance": "<success/failure>",
  "content_preservation": "<success/failure>",
  "overall_judgement": "<success/failure>"
}
```

Figure 34: Prompt for MLLM-as-a-Judge evaluation on RESHAPE-CANVAS editing tasks.