PAINTBENCH: DETERMINISTIC EVALUATION OF PRECISE VISUAL EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

While current multimodal models are proficient at open-ended visual editing, executing precise single-answer edits remains an important obstacle. To probe this challenge, we introduce PAINTBENCH, a dynamically scalable benchmark targeting 20 fundamental precise visual editing operations across four categories: geometric transformation, structural manipulation, color change, and symbolic reasoning. Procedural generation with configurable complexity enables an effectively infinite, contamination-resistant evaluation suite, and deterministic pixel-level evaluation eliminates reliance on bias-prone judge models. Across 11 image editing models, we find overall low performance, with the current highest-performing industry leader scoring only 17.1% (mIoU). Task decomposition reveals especially challenging operation types (geometric transformation, most structural manipulation, formula-based color change) and model-specific specializations. Fine-grained benchmark diagnostics further show performance degradations induced by scene variations in object count, background complexity, color scheme, and edit-region size. To test generalization of PAINTBENCH scores to applied task performance, we create a procedural, deterministic evaluation for data visualization editing (TINYGRAFIXBENCH) and find strong linear correlation with PAINTBENCH scores ($R^2 = 0.91, p < 0.001$). Altogether, PAINTBENCH provides a rigorous foundation for measuring and driving progress in precise multimodal visual editing.

1 INTRODUCTION

Generative image editing has advanced rapidly through the development of architectures such as instruction-tuned diffusion models (Brooks et al., 2023), flow-matching editors (Black Forest Labs et al., 2025), and native multimodal generators (OpenAI, 2026; Google, 2026). Yet evaluating whether these models can execute precise single-answer edits remains an open challenge. Consider a simple task: recolor the olive-colored triangle to cyan (#0FE1DF). A model may produce an output that looks correct at first glance, but the generated cyan may deviate substantially from the target color, and the background may be slightly altered. Reducing such errors is crucial for a myriad of applications that require exact precision.

Existing image editing benchmarks (Zhang et al., 2023; Basu et al., 2023; Ma et al., 2024) largely rely on human judgment or model-based evaluation via vision-language models and perceptual metrics (*e.g.*, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), FID (Heusel et al., 2017)). These are natural choices for many natural-image editing tasks without a single correct answer: "make the sky more blue" admits infinitely many valid outputs, so evaluation is closer to preference scoring than correctness verification. Recent benchmarks targeting complex editing (Han et al., 2025; Yang et al., 2026) continue this tradition of using powerful learned judge models.

However, many editing tasks *do* have unique correct answers, in which case evaluation can be performed deterministically without potential biases of model-based scoring or human judgment. Examples of such tasks include moving a shape 50 pixels to the right, flood-filling a region to a target color, or removing the smallest instance of a shape. Motivated by this observation, we create PAINTBENCH, a benchmark of such deterministic tasks. The name reflects the simplicity of its constituent operations: these tasks can be completed easily in a raster image editor, yet they are far from easy for current multimodal models.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

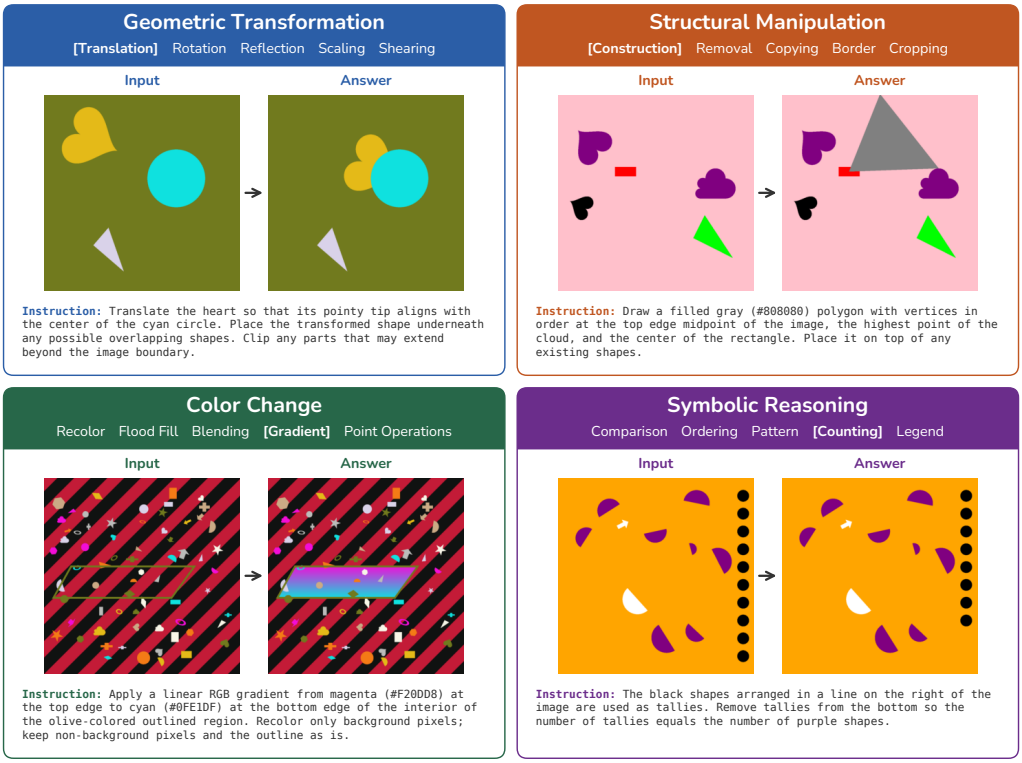


Figure 1: **PAINTBENCH spans 20 fundamental visual editing operations in four categories.** Each panel shows an example seed-generated problem from one operation (highlighted as [∗]), with an input image, instruction, and ground-truth answer. The deterministic, single-answer design enables automated, pixel-level evaluation without bias-prone judge models.

PAINTBENCH generates problems procedurally: a random seed and parameter configuration produces an input image, natural-language instruction, and unique correct answer (see Section 3 for design principles). In the same way that HumanEval (Chen et al., 2021) evaluates code generation by executing programs against test cases rather than judging surface plausibility, PAINTBENCH evaluates visual editing by direct pixel comparison against ground truth across a range of color precision thresholds, without relying on bias-prone judge models. Because problems are seed-generated with configurable scene parameters, PAINTBENCH also functions as a flexible diagnostic instrument: practitioners can generate task sets targeting specific operations or visual conditions of interest, going beyond a static snapshot of capabilities. In line with the broader vision-first push in multimodal research (Wang et al., 2024; Tong et al., 2024; 2026; Zhao et al., 2025a; Gabeur et al., 2026), we focus our evaluation on native pixel-space image editing models, rather than agentic or programmatic approaches that delegate the edit to a raster engine.

Across 11 models spanning native multimodal generators, diffusion editors, and flow-matching editors, we find that even the best-performing model reaches only 17.1% mIoU. Task difficulty tracks the underlying fundamental operation: geometric transformation, formula-based color change, and most structural manipulation tasks are consistently hard across all models (Section 5), while removal and single-color operations are more tractable, though still far from solved. Beyond overall rankings, models exhibit notable task-specific specializations on individual operations that diverge from their overall standing. By varying PAINTBENCH's procedural scene parameters, we find that striped backgrounds, high object counts, nonstandard color palettes, and small edit-regions all substantially degrade performance. To test whether PAINTBENCH scores predict performance on *applied tasks* built upon the same fundamental operations, we introduce TINYGRAFIXBENCH, applying our procedurally generated, deterministically evaluated philosophy to data visualization editing. Model scores between the two are strongly correlated ($R^2 = 0.91$, $p < 0.001$), suggesting that PAINTBENCH captures capabilities that generalize beyond its synthetic-shape scope.

Our contributions include:

1. PAINTBENCH: a seed-generated deterministic benchmark of 20 fundamental precise visual editing operations across four categories, comprising a 1,920-problem test set.
2. A pixel-level evaluation protocol that factors in both edit- and preservation quality using perceptual color similarity, with no reliance on bias-prone judge models.
3. A systematic evaluation of 11 models showing that even the best-performing model reaches only 17.1% mIoU, with consistent challenges across geometric transformation, formula-based color change, and structural manipulation tasks.
4. PAINTBENCH's configurable procedural generator enables controlled scene-variation analysis: striped backgrounds, high object counts, nonstandard color palettes, and small edit-regions all substantially degrade performance.
5. TINYGRAFIXBENCH: a companion benchmark applying the same procedural, deterministic principles to data visualization editing. Model scores between the two are strongly correlated, suggesting that PAINTBENCH captures capabilities that generalize to applied visual editing.

2 RELATED WORK

We situate PAINTBENCH against prior image editing benchmarks, broader generative model evaluation, synthetic benchmarking methodologies, and image editing model capabilities.

Image Editing Benchmarks. Several benchmarks evaluate instruction-following image editors using real-image edit triplets and subjective evaluation. MagicBrush (Zhang et al., 2023), EditVal (Basu et al., 2023), I2EBench (Ma et al., 2024), and ImgEdit (Ye et al., 2025) curate such triplets and rely on human judgment, vision-language model scoring, or perceptual metrics. Complex-Edit (Yang et al., 2026) probes how performance degrades with instruction complexity, and UniREditBench (Han et al., 2025) adds a reasoning-heavy dual-reference variant. Recent work proposes fine-grained MLLM judges (Liu et al., 2026) for editing evaluation, but the community has not converged on reliable methods, and judge models still yield highly uncertain absolute scores (Kumar et al., 2026). PAINTBENCH departs from these designs: by focusing on edits with deterministic ground truth, pixel-level comparison against a known answer eliminates the need for judge models entirely. Benchmarks for reasoning-heavy edits with inherently subjective outputs (*e.g.*, RISEBench (Zhao et al., 2025b), KRIS-Bench (Wu et al., 2025b)) address a complementary problem.

Evaluation of Generative Models. GenEval (Ghosh et al., 2023) establishes object-focused text-to-image evaluation using detection models as proxy judges. T2I-CompBench (Huang et al., 2023) evaluates compositional generation via BLIP-based scoring. GenEval 2 (Kamath et al., 2025) identifies *benchmark drift* (static benchmarks becoming misaligned as models improve) and proposes harder evaluation sets. This motivates PAINTBENCH's dynamic generation: fresh problems can be generated at will, preventing saturation and contamination. A Very Big Video Reasoning Suite (Wang et al., 2026) proposes rule-based, human-aligned scoring for video reasoning benchmarks as an alternative to model-based judging.

Synthetic Visual Reasoning Benchmarks. CLEVR (Johnson et al., 2017) established programmatic visual scene generation for diagnostic evaluation; ARC (Chollet, 2019), RAVEN (Zhang et al., 2019), and Bongard-Logo (Nie et al., 2020) extend this tradition to abstract visual reasoning with deterministic answers. In the language domain, HumanEval (Chen et al., 2021) measures functional correctness of code generation by executing programs against test cases, philosophically aligned with PAINTBENCH's pixel-level verification. We extend this tradition of synthetic, deterministic evaluation to generative image editing. Concurrent work Gen-ViRe (Liu et al., 2025) evaluates generative visual reasoning for video generation, but targets world simulators and relies on VLM-assisted evaluation rather than deterministic comparison.

Image Editing Models. Modern image editing models span architectures that include instruction-tuned diffusion editors (Brooks et al., 2023; Wu et al., 2025a; Meituan LongCat Team et al., 2025), flow-matching editors (Black Forest Labs et al., 2025; Black Forest Labs, 2025), and native multimodal generators that produce text and images from a single backbone (Wang et al., 2024; Deng

et al., 2025; Tencent Hunyuan Foundation Model Team, 2025; Google, 2025; 2026; OpenAI, 2026). Complementary approaches may write code or call external tools to perform edits. Such approaches lie outside our scope: PAINTBENCH is a proxy benchmark designed to evaluate models that natively *output* in pixel-space. The native pixel-space class is increasingly the product of a research thread on *unified multimodal models* aimed at combining image understanding and generation in one backbone (Chameleon Team, 2024; Zhou et al., 2025; Tong et al., 2025; Zhao et al., 2025a; Tong et al., 2026), further motivated by evidence that image generators have become competent generalist vision systems on their own (Gabeur et al., 2026; Tong et al., 2024). PAINTBENCH tests the pixel-level precision side of that unified claim: whether models that can *describe* a scene can also *edit* it exactly.

3 BENCHMARKING PRECISE VISUAL EDITING

PAINTBENCH is a procedurally-generated deterministic evaluation of multimodal model capabilities on fundamental operations in precise single-answer visual editing. Here, we describe the design principles (Section 3.1), task categories (Section 3.2), and visual conditions (Section 3.3).

Each problem consists of an (input image, instruction, answer image) triple generated from a seed. Scenes consist of geometric shapes of varying types and colors rendered on solid-color or striped backgrounds. For each of the 20 tasks, we generate 12 problems for each of the 8 visual conditions, yielding 1,920 problems in total (20 tasks $\times$ 8 conditions $\times$ 12 problems; see Section C.3).

3.1 DESIGN PRINCIPLES

Four principles guide the design of PAINTBENCH. (1) **Determinism**: every problem has exactly one correct output image, produced by a deterministic transformation $A = f(I, t)$ from the input image I and instruction t; evaluation reduces to a pixel-level comparison of the model's output $\hat{A}$ against A and I, with no judge models, no perceptual proxies, and no ambiguity. (2) **Dynamic Generation**: problems are produced procedurally from random seeds, so fresh problem sets can be generated at will to prevent memorization or contamination. (3) **Controlled Difficulty**: tasks expose explicit parameters (canvas dimensions, object count, background texture, color palette) that vary scene conditions, enabling precise ablations (Section 5.3). (4) **Atomic Operations**: tasks target fundamental visual editing operations that serve as the building blocks of complex workflows.

3.2 TASK CATEGORIES

PAINTBENCH comprises 20 task types organized into four categories (Figure 1; full taxonomy in Table 4 and per-task descriptions in Section B.1). *Geometric Transformation* (translation, rotation, reflection, scaling, shearing) tests affine transformation of shapes. *Structural Manipulation* (construction, removal, copying, border, cropping) tests addition and removal of elements, and modification of scene composition. *Color Change* (recolor, flood fill, blending, gradient, point operations) tests manipulation of pixel color values. *Symbolic Reasoning* (comparison, ordering, pattern, counting, legend) tests edits that require spatial or numerical inference before execution. Generation details (including modes that vary the operation within each task) appear in Section B.

3.3 VISUAL CONDITIONS

Every task is evaluated across eight visual conditions, each changing exactly one scene parameter relative to the baseline condition (1024×1024, $n = 3$ shapes, standard color palette, single-color background; Figure 7, full enumeration in Table 10). This breakdown reveals model sensitivity to scene variations and ensures overall benchmark scores reflect a diverse mix of conditions.

4 PIXEL-LEVEL EVALUATION

Given an input image I and ground-truth answer image A, we define the *edit-region* $\mathcal{E}$ as pixels that differ between them, and the *preservation-region* $\mathcal{P}$ as pixels that are identical (Figure 2).

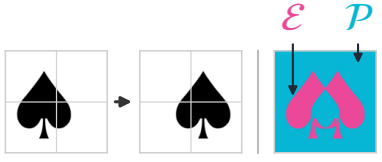


Figure 2: Edit ($\mathcal{E}$) and preservation ($\mathcal{P}$) regions for a simple translation.

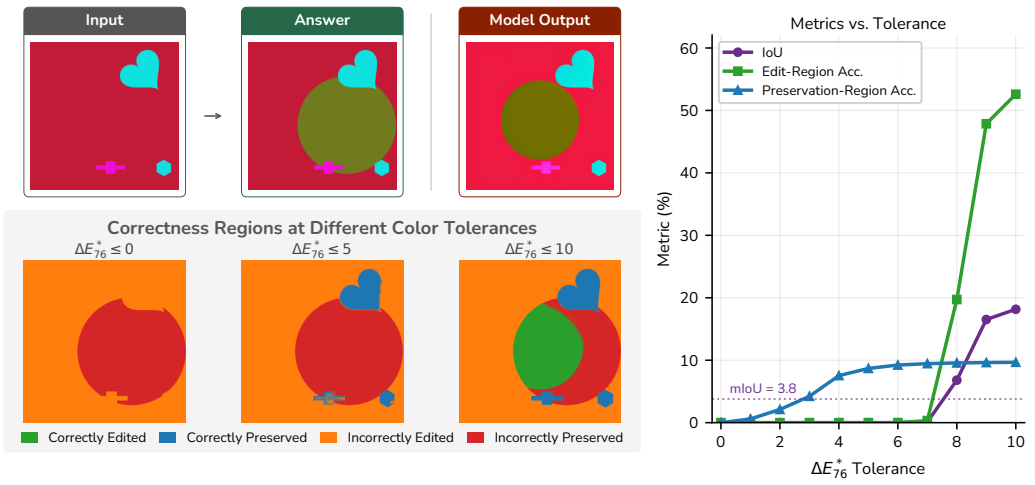


Figure 3: **Deterministic evaluation measures geometric and color accuracy at the pixel-level.** *Top left:* The model output is graded against the input and answer images. *Bottom left:* At a color tolerance t (defined by the ΔE^*_{76} convention; the lower the stricter), each pixel is classified as ***correctly* edited**, ***correctly* preserved**, ***incorrectly* edited**, or ***incorrectly* preserved**. *Right:* **IoU@t** increases with rising ΔE^*_{76} tolerance t due to increasing *edit-* and *preservation*-region accuracies (§4.2); averaging over $t \in \{0, \ldots, 10\}$ gives an overall score **mIoU** for one problem (§4.3).

4.1 Color Distance

We measure per-pixel color difference using ΔE^*_{76} (Robertson, 1977), the Euclidean distance in CIE L*a*b* color space: $\Delta E^*_{76}(o_i, a_i) = \sqrt{(L_o - L_a)^2 + (a_o - a_a)^2 + (b_o - b_a)^2}$, where L, a, b are the lightness and chromaticity coordinates of output pixel o_i and answer pixel a_i. Under a commonly used rule of thumb, $\Delta E^*_{76} \le 1$ is imperceptible, 1–2 is perceptible under close observation, 2–10 is perceptible at a glance, and >10 indicates distinct colors (Minaker et al., 2021; Schuessler, 2016). A pixel is *correct* if $\Delta E^*_{76}(o_i, a_i) \le t$ for a chosen color tolerance t; we evaluate at integer tolerances $t \in \{0, 1, \ldots, 10\}$, spanning exact pixel match ($t = 0$) to a lenient tolerance ($t = 10$).

4.2 Per-Problem Score: IoU@t

Each problem's pixels fall into four disjoint sets at tolerance t: correctly edited (CE) and incorrectly edited (IE) pixels in the edit-region $\mathcal{E}$, partitioned by whether $\Delta E^*_{76} \le t$; and correctly preserved (CP) and incorrectly preserved (IP) pixels in the preservation-region $\mathcal{P}$, similarly partitioned. We score each problem with the Jaccard index applied to pixel correctness:

$$\text{IoU@}t = \frac{|\text{CE}|}{|\text{CE}| + |\text{IE}| + |\text{IP}|} \tag{1}$$

This formulation jointly penalizes failure to execute the requested edit (increasing $|\text{IE}|$) and corruption of the preservation-region (increasing $|\text{IP}|$). IoU@t is robust to edit-region size, naturally handling the common case where the preservation-region is much larger than the edit-region.

4.3 Summary Metric: mIoU

To capture model performance across the full spectrum of color tolerances (and avoid arbitrarily committing to a single one), we summarize IoU@t by sweeping t across $\{0, 1, \ldots, 10\}$. This is analogous to COCO's Average Precision metric (Lin et al., 2014), which sweeps IoU thresholds rather than reporting one. The resulting summary metric **mIoU** (mean IoU) averages IoU@t over these 11 tolerances and all N problems:

$$\text{mIoU} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{11} \sum_{t=0}^{10} \text{IoU@}t(i) \tag{2}$$

We use mIoU throughout; tolerance-specific values are written as IoU@t.

Table 1: **PAINTBENCH mIoU (%) per task.** Best score per task in bold. *Category Avg.* rows are macro-averages over the 5 tasks in the category; *Benchmark Avg.* is the macro-average over 20 tasks with 95% bootstrap CIs (±; full intervals in Table 12).

Task	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Geometric Transformation											
Category Avg.	6.1 ±0.7	**11.1** ±0.9	6.2 ±0.8	3.4 ±0.5	2.4 ±0.4	3.3 ±0.5	2.4 ±0.3	2.2 ±0.4	1.4 ±0.3	0.1 ±0.1	0.0 ±0.0
Translation	12.3	**17.5**	9.6	5.3	3.6	4.5	2.7	3.9	3.0	0.1	0.0
Rotation	7.6	**13.2**	7.1	5.5	4.0	5.6	3.0	2.9	1.9	0.1	0.1
Reflection	4.4	**9.1**	5.1	3.0	1.8	4.0	2.7	2.5	1.1	0.1	0.0
Scaling	3.0	**7.8**	4.6	1.9	1.0	0.8	1.8	0.7	0.8	0.2	0.0
Shearing	3.1	**7.8**	4.4	1.4	1.4	1.6	1.8	1.2	0.5	0.0	0.0
Structural Manipulation											
Category Avg.	22.7 ±1.7	**24.5** ±1.5	14.0 ±1.5	10.3 ±1.3	10.0 ±1.4	8.2 ±1.1	7.9 ±1.1	7.1 ±1.2	7.2 ±1.0	0.8 ±0.3	0.9 ±0.4
Construction	**15.7**	14.3	4.8	5.3	1.0	3.4	5.3	2.4	3.1	0.6	1.2
Removal	45.8	**50.6**	38.2	31.7	27.5	21.1	24.1	18.7	22.6	2.3	3.1
Copying	**14.0**	13.9	12.8	6.0	4.9	1.4	0.8	3.2	4.5	0.1	0.0
Border	**18.9**	15.2	4.6	0.6	0.1	0.4	0.3	0.1	0.1	0.1	0.1
Cropping	19.1	**28.5**	9.5	7.8	16.5	14.7	9.2	10.9	5.4	1.1	0.1
Color Change											
Category Avg.	**17.2** ±1.6	13.8 ±1.5	6.4 ±1.2	5.4 ±1.1	2.6 ±0.8	2.7 ±0.8	1.6 ±0.5	1.8 ±0.6	2.4 ±0.7	0.2 ±0.1	0.2 ±0.1
Recolor	**30.4**	29.0	7.8	6.8	8.8	5.0	2.0	2.2	4.2	0.3	0.3
Flood Fill	24.8	**27.1**	11.2	16.3	2.3	4.9	2.6	4.7	3.8	0.3	0.2
Blending	5.3	**6.4**	2.6	1.2	1.1	0.7	1.7	0.9	1.2	0.1	0.1
Gradient	**13.0**	1.4	2.9	0.7	0.2	0.8	0.5	0.1	1.1	0.1	0.0
Point Operations	**12.3**	5.4	7.4	2.2	0.7	2.1	1.1	0.9	1.7	0.3	0.6
Symbolic Reasoning											
Category Avg.	**22.6** ±1.6	15.9 ±1.4	18.0 ±1.4	7.7 ±1.1	5.1 ±0.9	4.1 ±0.6	3.6 ±0.6	3.5 ±0.6	3.2 ±0.5	0.3 ±0.1	0.1 ±0.1
Comparison	**16.1**	10.7	14.3	12.9	4.2	6.2	6.0	7.5	8.2	0.4	0.5
Ordering	20.0	**21.0**	18.2	8.0	5.1	6.2	3.9	4.5	1.8	0.2	0.1
Pattern	13.4	**13.7**	8.7	8.0	4.4	5.4	2.3	0.9	3.2	0.4	0.0
Counting	**16.3**	14.9	14.8	8.4	5.5	1.6	1.9	2.2	2.3	0.3	0.0
Legend	**47.1**	19.4	34.2	1.1	6.1	1.1	3.9	2.4	0.3	0.1	0.0
Benchmark Avg.	**17.1** ±0.7	16.3 ±0.7	11.1 ±0.6	6.7 ±0.5	5.0 ±0.5	4.6 ±0.4	3.9 ±0.3	3.6 ±0.4	3.5 ±0.3	0.4 ±0.1	0.3 ±0.1

5 LEADING MODELS FAIL TO EXECUTE PRECISE EDITS

We evaluate 11 models on PAINTBENCH, organizing our analyses around four findings: operation difficulty and model specialization (Section 5.2); common failure modes (Section A.1); sensitivity to scene variations (Section 5.3); and a pervasive over-editing tendency (Section 5.4).

5.1 SETUP

We evaluate 11 models spanning distinct architectural families. NANO-BANANA-2 (Google, 2026), NANO-BANANA-1 (Google, 2025), and GPT-IMAGE-2 (OpenAI, 2026) are closed-weights native multimodal generators; the remaining eight are open-weights. QWEN-IMAGE-EDIT-2511 (Wu et al., 2025a) is a 20B instruction-tuned diffusion editor. FLUX.2-DEV (Black Forest Labs, 2025) is a 32B flow-matching generator, and FLUX.2-KLEIN-9B (Black Forest Labs, 2025) is a step-distilled variant from the same family targeting sub-second inference. FLUX.1-KONTEXT-DEV (Black Forest Labs et al., 2025) is a 12B rectified flow transformer specialized for instruction-based image editing. BAGEL (Deng et al., 2025) is a 7B mixture-of-transformer-experts model. HUNYUANIMAGE-3.0 (Tencent Hunyuan Foundation Model Team, 2025) is an 80B mixture-of-experts instruction-following image generator. LONGCAT-IMAGE-EDIT (Meituan LongCat Team et al., 2025) is a 6B long-context diffusion editor. INSTRUCTPIX2PIX (Brooks et al., 2023) is a 1B diffusion editor frequently cited as an older baseline. We generate one output per problem per model (see Section C).

5.2 OPERATION DIFFICULTY AND MODEL SPECIALIZATION

Table 1 reports mIoU per task and model across all 1,920 PAINTBENCH problems. The strongest model (NB-2) reaches only 17.1% mIoU; GPT-I2 follows at 16.3%, NB-1 at 11.1%, and open-weights models range from 6.7% (QWEN-IE) down to below 1% (HY-3, IP2P). Remarkably, despite being the largest open-weights model we evaluate (80B), HY-3 scores near zero on almost all tasks. Aggregate rankings tell only part of the story; the table reveals a clear difficulty gradient across operation types, and per-task scores expose pronounced model specializations.

Table 2: **Striped backgrounds and high object counts cause the largest mIoU drops.** All values are averaged over all 20 PAINTBENCH tasks. The baseline row shows absolute mIoU (%); all other rows report ΔmIoU (%) relative to baseline ($n = 3$ for most tasks, 1024×1024, solid background, standard palette). Cell shading: green = above baseline, red = below baseline, white = no change. Per-task breakdowns in Table 18.

Condition	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Baseline	**21.9**	20.9	13.4	7.2	6.4	5.2	4.3	3.7	3.9	0.4	0.4
Aspect Ratio (baseline: square 1024×1024)											
Horizontal (1024×576)	−2.7	−1.6	−1.9	+1.5	+0.2	+0.5	+1.2	+1.4	+1.5	+0.2	−0.1
Vertical (576×1024)	−0.9	−3.6	−0.1	+0.8	−1.4	+1.2	−0.6	−0.2	+1.8	+0.1	−0.2
Color Palette (baseline: standard palette)											
Nonstandard	+1.2	+1.9	+4.0	+1.4	−1.9	−0.6	+1.0	+3.0	−0.2	−0.1	−0.2
Background (baseline: solid)											
Striped	−11.1	−8.8	−6.0	−2.7	−2.4	−0.5	−1.9	−2.0	−1.9	0.0	−0.1
Object Count (baseline: $n = 3$ for most tasks; see Table 11)											
n_{med}	−4.0	−4.0	−1.6	−0.4	−1.4	−1.6	−0.4	−0.3	−0.8	0.0	+0.2
n_{high}	−8.9	−9.4	−5.7	−1.9	−2.2	−2.4	−1.0	−1.3	−1.7	−0.2	+0.1
n_{xhigh}	−11.9	−11.3	−7.1	−2.6	−2.3	−1.8	−1.8	−1.1	−1.6	−0.2	−0.3

Geometric and structural operations are consistently hard. The entire geometric transformation category is consistently difficult. No model exceeds 17.5% mIoU on any geometric task, with shearing and scaling especially difficult across the board ($\leq 7.8\%$ for both). Most structural manipulation tasks and formula-based color changes are also challenging: unlike single-color operations (recolor and flood fill), tasks such as gradient, blending, and point operations require different pixels to be colored differently, and scores are correspondingly low across all models.

Removal and single-color operations are more tractable. Removal reaches 50.6% (GPT-I2), flood fill 27.1% (GPT-I2), and recolor 30.4% (NB-2). Open-weights models are strongest on removal, reaching 31.7% for QWEN-IE and 27.5% for BAGEL. These tasks all involve filling a connected region with a single color or removing content entirely, which is structurally simpler than the per-pixel computation that gradient, blending, or point operations demand.

🎨 Overall performance is low (best model at 17.1% mIoU), with difficulty tracking the underlying primitive: geometric transformation, formula-based color change, and most structural manipulation tasks are consistently hard, while removal and single-color operations are more tractable, though still far from solved.

NANO-BANANA-2 and GPT-IMAGE-2 specialize in complementary categories. The two leading models split the four PAINTBENCH categories down the middle: GPT-I2 leads geometric transformation (11.1% vs. 6.1%) and structural manipulation (24.5% vs. 22.7%), while NB-2 leads color change (17.2% vs. 13.8%) and symbolic reasoning (22.6% vs. 15.9%). GPT-I2's geometric transformation advantage holds across every task in the category: translation (17.5% vs. 12.3%), rotation (13.2% vs. 7.6%), reflection (9.1% vs. 4.4%), scaling (7.8% vs. 3.0%), and shearing (7.8% vs. 3.1%). Within structural manipulation, GPT-I2 leads on removal (50.6% vs. 45.8%) and cropping (28.5% vs. 19.1%). Meanwhile, NB-2 scores substantially higher than GPT-I2 on gradient (13.0% vs. 1.4%) and point operations (12.3% vs. 5.4%). A notable outlier is legend, where NB-2 scores 47.1% vs. GPT-I2's 19.4%, a 28-point spread despite comparable overall scores. We hypothesize this reflects different training-data composition or fine-tuning emphases between the models.

🎨 PAINTBENCH diagnoses task-specific profiles that benchmark averages conceal: GPT-I2 and NB-2 specialize in complementary categories, and individual models exhibit pronounced strengths and weaknesses on specific tasks that their overall score does not predict.

5.3 BRITTLENESS TO SCENE VARIATION

Every PAINTBENCH task is evaluated under eight visual conditions, enabling direct measurement of how scene variations affect model performance. Table 2 reports mIoU averaged over all 20 tasks for each condition; per-task breakdowns appear in Table 18.

Striped backgrounds and high object counts cause the largest drops. Striped backgrounds knock NANO-BANANA-2 down 11.1 points and GPT-IMAGE-2 down 8.8; all other models also decline, with the exception of HUNYUANIMAGE-3.0, which remains near zero under both conditions. Performance degrades similarly as object count increases: GPT-IMAGE-2 drops 11.3 points from baseline to n_{xhigh} and NANO-BANANA-2 drops 11.9, with n_{med} and n_{high} causing proportionally smaller but still substantial declines across almost all models.

Models fail at exactly reproducing nonstandard colors, despite flat mIoU. At the mIoU level, the nonstandard palette appears to produce no consistent directional effect: roughly half the models slightly improve and half slightly decline. But exact pixel-match ($\Delta E^*_{76} = 0$) accuracies tell a different story. Both edit- and preservation-region accuracies are *much* lower under the nonstandard palette for *every* model (Figure 8). To illustrate, BAGEL's preservation accuracy plummets from 19.2% to 0.3%, and NB-2 from 13.6% to 0.9%. Even QWEN-IE, the most proficient at reproducing nonstandard colors, achieves only 3.6% preservation accuracy versus 9.6% baseline. We hypothesize this reflects training-data composition: standard colors are far more prevalent.

🎨 Models are brittle to scene variation: striped backgrounds and high object counts cause large and consistent mIoU drops, and models struggle at exactly reproducing nonstandard colors despite flat headline scores.

5.4 MODELS OVER-EDIT RELATIVE TO THE TARGET REGION

Beyond the scene parameters we explicitly control for, models change *far* more pixels between input and output than the edit-region requires. We quantify this as the ratio of changed pixel count (at color tolerance $\Delta E^*_{76} \leq 5$[1]) divided by edit-region size; a perfect edit implies a ratio of 1. Figure 9 plots this ratio (left) and mIoU (right) as a function of edit-region size, bucketed from $< 32^2$ to $\geq 256^2$ pixels (roughly 0.1% to 6% of the 1024^2 canvas). Median ratios vary dramatically across edit-region size, ranging from $\sim$1–8$\times$ for the largest edit-regions ($\geq 256^2$ pixels) all the way to $\sim$50–1,400$\times$ for the smallest ($< 32^2$ pixels). This discrepancy is starkly reflected in mIoU: NANO-BANANA-2 rises from a mIoU of only 0.9% at $< 32^2$ pixels to 28.7% at $\geq 256^2$ pixels, and GPT-IMAGE-2 from 1.7% to 24.1%. All models exhibit a pronounced increase in mIoU as edit-region size increases.

🎨 Models over-edit by 1–8$\times$ the edit-region area for large regions and by 50–1,400$\times$ for the smallest, driving sharp mIoU declines on problems with small edit-regions.

6 TINYGRAFIXBENCH: GENERALIZATION BEYOND SYNTHETIC SHAPES

TINYGRAFIXBENCH applies PAINTBENCH's procedurally generated, deterministically evaluated framework to data visualization editing, testing whether PAINTBENCH's fundamental operations generalize to chart editing tasks.

Benchmark design. TINYGRAFIXBENCH comprises 600 problems across five chart types (bar chart, scatter plot, line chart, heatmap, network graph), with four tasks per chart type corresponding to four fundamental editing operations: construction, transformation, removal, and recoloring (Figure 10). Charts are rendered with Matplotlib at 1024×768 pixels using deterministic seeds and evaluated with the same mIoU protocol as PAINTBENCH (see Section B.7 for details; full per-task results in Table 21).

[1] $\Delta E^*_{76} \leq 5$ provides a principled median of changed pixel counts across all ΔE^*_{76} tolerances in $[0, 10]$.

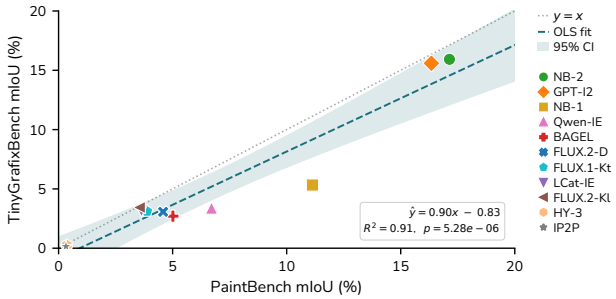


Figure 4: **PAINTBENCH and TINYGRAFIXBENCH mIoU scores are strongly linearly correlated across models.** Each point is one model. The OLS fit (dashed teal, with 95% CI) closely tracks $y = x$ (dotted gray); regression yields $R^2 = 0.91$, $p < 0.001$.

TINYGRAFIXBENCH is harder, but scores track PAINTBENCH. Overall performances on TINYGRAFIXBENCH are slightly lower than on PAINTBENCH, potentially reflecting the additional difficulty of understanding and editing chart imagery. However, model scores between the benchmarks exhibit a *strong* correlation ($R^2 = 0.91$, $p < 0.001$; Figure 4). NANO-BANANA-2 leads on TINYGRAFIXBENCH at 15.9% and GPT-IMAGE-2 follows close behind at 15.6%, consistent with their relative standings on PAINTBENCH; the other models are at 5.3% (NB-1) or below.

🎨 Model scores on TINYGRAFIXBENCH are strongly correlated with those on PAINTBENCH ($R^2 = 0.91$, $p < 0.001$), suggesting that PAINTBENCH captures capabilities that generalize to applied visual editing tasks.

7 DISCUSSION

We introduce PAINTBENCH and TINYGRAFIXBENCH on the premise that a meaningful class of visual editing tasks that contain unique correct outputs has been underserved by evaluation frameworks built for subjective, open-ended generation. By constructing problems procedurally from random seeds, both benchmarks support pixel-level evaluation without bias-prone judge models or perceptual proxies, while protecting against contamination. Because PAINTBENCH enables controllable difficulty and scene variation, practitioners can generate custom fine-grained task sets, turning the benchmark from a static snapshot into a configurable diagnostic instrument.

Our experiments reveal that current image editing models cannot reliably execute basic raster operations despite strong open-ended generation performance. Geometric transformation, formula-based color change, and most structural manipulation tasks are effectively unsolved across all models; even the most tractable removal and recoloring tasks are far from reliably executed. Yet amidst the generally low scores, distinct model specializations in different tasks and task categories emerge. Our analysis shows that models are brittle to scene variation in various forms: high shape counts, striped backgrounds, nonstandard colors, and small edit-regions. Promisingly, the strong correlation between PAINTBENCH and TINYGRAFIXBENCH scores suggests that PAINTBENCH captures capabilities that generalize to applied visual editing tasks built on similar primitives.

Limitations. Depending on real-world application, the deterministic evaluation metric may warrant adjustment. For some tasks, our metric may be particularly strict, especially with small or thin edit-regions that are easy to miss. Still other tasks may require an even stricter pass-rate-style metric (e.g., percentage of problems with IoU@2 $\geq 99.9\%$) rather than mIoU. Furthermore, precise editing tasks with non-unique answers (such as those involving text rendering or non-unique ways to fill the background exposed by a transformed shape) may require more complex deterministic evaluation metrics. Nevertheless, we encourage practitioners to carefully consider the feasibility of using a deterministic metric for the task at hand, and design or adopt one that best captures the most important considerations.

AI USE STATEMENT

In this work, we used generative AI tools for implementation (writing the procedural problem generators and evaluation code), and as an assistant in research methodology design and results interpretation (the benchmark design, core methodological decisions, and interpretation of results are the authors' own). Because the procedural generators were written with AI assistance, generative AI contributed indirectly to synthetic data generation; however, no benchmark data was produced by a generative model directly: every PAINTBENCH and TINYGRAFIXBENCH problem (input image, instruction, and answer image) is produced deterministically from a random seed by the generator code (Section B), which the authors reviewed. We have not used generative AI tools for hypothesis refinement or qualitative analysis, and developing theoretical models, formulating mathematical claims, proving theorems, data cleaning, and translation are not applicable to this work. Additionally, we used generative AI tools for figure creation, code editing, literature analysis, brainstorming, paper drafting, literature summarization, and reference formatting. We have reviewed all AI-assisted work: all problems regenerate identically from their seeds, all reported numbers are produced by scripts from the evaluation outputs, and every citation was manually checked against its source. Separately, the image editing models evaluated in this work are themselves generative AI systems; they are the object of study (Section 5) rather than tools used to conduct the research. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

PAINTBENCH and TINYGRAFIXBENCH consist entirely of procedurally generated synthetic images (geometric shapes and Matplotlib charts); they involve no human subjects, personal data, or scraped web content. Open-weights models were run under their respective licenses, and proprietary models were accessed through official APIs under the providers' terms of service (Section C). More precise image editing is dual-use: the same capabilities that support design and visualization workflows could make manipulated images or misleading data visualizations harder to detect. PAINTBENCH measures these capabilities rather than providing them, and its deterministic, pixel-level evaluation can equally support tracking and auditing how precisely models alter visual content.

REPRODUCIBILITY STATEMENT

All problems are generated deterministically from random seeds; the generation procedure, shape library, color system, and seed design are documented in Section B, and the benchmark configuration in Section C. The evaluation metric is fully specified in Section 4, and evaluated models, inference parameters, and licenses are listed in Section C. Bootstrap confidence intervals and full per-task results are provided in Section D. Anonymized code for problem generation, model inference, and evaluation is available at https://anonymous.4open.science/r/PaintBench, and the exact PAINTBENCH and TINYGRAFIXBENCH test sets used in this paper are released at https://huggingface.co/datasets/PaintBenchICLR2027/PaintBench.

REFERENCES

Samyadeep Basu, Mehrdad Saberi, Shweta Bhardwaj, Atoosa Malemir Chegini, Daniela Massiceti, Maziar Sanjabi, Shell Xu Hu, and Soheil Feizi. EditVal: Benchmarking Diffusion Based Text-Guided Image Editing Methods. *arXiv preprint arXiv:2310.02426*, 2023.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025. URL https://bfl.ai/blog/flux-2.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space. *arXiv preprint arXiv:2506.15742*, 2025.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *CVPR*, 2023.

Chameleon Team. Chameleon: Mixed-Modal Early-Fusion Foundation Models. *arXiv preprint arXiv:2405.09818*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*, 2021.

François Chollet. On the Measure of Intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging Properties in Unified Multimodal Pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Bradley Efron and Robert J Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1994.

Valentin Gabeur, Shangbang Long, Songyou Peng, Paul Voigtlaender, Shuyang Sun, Yanan Bao, Karen Truong, Zhicheng Wang, Wenlei Zhou, Jonathan T Barron, et al. Image Generators are Generalist Vision Learners. *arXiv preprint arXiv:2604.20329*, 2026.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An Object-Focused Framework for Evaluating Text-to-Image Alignment. In *NeurIPS*, 2023.

Google. Introducing Gemini 2.5 Flash Image. https://developers.googleblog.com/en/introducing-gemini-2-5-flash-image/, 2025. URL https://developers.googleblog.com/en/introducing-gemini-2-5-flash-image/.

Google. Nano Banana 2 (Gemini 3.1 Flash Image). https://deepmind.google/models/gemini-image/flash/, 2026. URL https://deepmind.google/models/gemini-image/flash/.

Feng Han, Yibin Wang, Chenglin Li, Zheming Liang, Dianyi Wang, Yang Jiao, Zhipeng Wei, Chao Gong, Cheng Jin, and Jiaqi Wang. UniREditBench: A Unified Reasoning-based Image Editing Benchmark. *arXiv preprint arXiv:2511.01295*, 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*, 2017.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench: A Comprehensive Benchmark for Open-World Compositional Text-to-Image Generation. In *NeurIPS*, 2023.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *CVPR*, 2017.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. GenEval 2: Addressing Benchmark Drift in Text-to-Image Evaluation. *arXiv preprint arXiv:2512.16853*, 2025.

Divake Kumar, Sina Tayebati, Devashri Naik, Ranganath Krishnan, and Amit Ranjan Trivedi. VLM Judges Can Rank but Cannot Score: Task-Dependent Uncertainty in Multimodal Evaluation. *arXiv preprint arXiv:2604.25235*, 2026.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

Runzhou Liu, Hailey Weingord, Sejal Mittal, Prakhar Dungarwal, Anusha Nandula, Bo Ni, Samyadeep Basu, Hongjie Chen, Nesreen K Ahmed, Li Li, et al. Human-Aligned MLLM Judges for Fine-Grained Image Editing Evaluation: A Benchmark, Framework, and Analysis. *arXiv preprint arXiv:2602.13028*, 2026.

Xinxin Liu, Zhaopan Xu, Ming Li, Kai Wang, Yong Jae Lee, and Yuzhang Shang. Can World Simulators Reason? Gen-ViRe: A Generative Visual Reasoning Benchmark. *arXiv preprint arXiv:2511.13853*, 2025.

Yiwei Ma, Jiayi Ji, Ke Ye, Weihuang Lin, Zhibin Wang, Yonghan Zheng, Qiang Zhou, Xiaoshuai Sun, and Rongrong Ji. I2EBench: A Comprehensive Benchmark for Instruction-Based Image Editing. In *NeurIPS*, 2024.

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

Samuel A Minaker, Ryan H Mason, and David R Chow. Optimizing Color Performance of the Ngenuity 3-Dimensional Visualization System. *Ophthalmology Science*, 2021.

Weili Nie, Zhiding Yu, Lei Mao, Ankit B Patel, Yuke Zhu, and Anima Anandkumar. Bongard-LOGO: A New Benchmark for Human-Level Concept Learning and Reasoning. In *NeurIPS*, 2020.

OpenAI. ChatGPT Images 2.0 System Card. https://deploymentsafety.openai.com/chatgpt-images-2-0/introduction, 2026. URL https://deploymentsafety.openai.com/chatgpt-images-2-0/introduction.

Alan R Robertson. The CIE 1976 Color-Difference Formulae. *Color Research & Application*, 1977.

Zachary Schuessler. Delta E 101. https://zschuessler.github.io/DeltaE/learn/, 2016. URL https://zschuessler.github.io/DeltaE/learn/.

Tencent Hunyuan Foundation Model Team. HunyuanImage 3.0 Technical Report. *arXiv preprint arXiv:2509.23951*, 2025.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A Fully Open, Vision-Centric Exploration of Multimodal LLMs. In *NeurIPS*, 2024.

Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. MetaMorph: Multimodal Understanding and Generation via Instruction Tuning. In *ICCV*, 2025.

Shengbang Tong, David Fan, John Nguyen, Ellis Brown, Gaoyue Zhou, Shengyi Qian, Boyang Zheng, Théophane Vallaeys, Junlin Han, Rob Fergus, Naila Murray, Marjan Ghazvininejad, Mike Lewis, Nicolas Ballas, Amir Bar, Michael Rabbat, Jakob Verbeek, Luke Zettlemoyer, Koustuv Sinha, Yann LeCun, and Saining Xie. Beyond Language Modeling: An Exploration of Multimodal Pretraining. In *ICML*, 2026.

Maijunxian Wang, Ruisi Wang, Juyi Lin, Ran Ji, Thaddäus Wiedemer, Qingying Gao, Dezhi Luo, Yaoyao Qian, Lianyu Huang, Zelong Hong, et al. A Very Big Video Reasoning Suite. *arXiv preprint arXiv:2602.20159*, 2026.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-Token Prediction is All You Need. *arXiv preprint arXiv:2409.18869*, 2024.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE TIP*, 2004.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-Image Technical Report. *arXiv preprint arXiv:2508.02324*, 2025a.

Yongliang Wu, Zonghui Li, Xinting Hu, Xinyu Ye, Xianfang Zeng, Gang Yu, Wenbo Zhu, Bernt Schiele, Ming-Hsuan Yang, and Xu Yang. KRIS-Bench: Benchmarking Next-Level Intelligent Image Editing Models. In *NeurIPS*, 2025b.

Siwei Yang, Mude Hui, Bingchen Zhao, Yuyin Zhou, Nataniel Ruiz, and Cihang Xie. Complex-Edit: CoT-Like Instruction Generation for Complexity-Controllable Image Editing Benchmark. *TMLR*, 2026. URL https://openreview.net/forum?id=lL1JR6dxG8.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, Li Yuan, et al. ImgEdit: A Unified Image Editing Dataset and Benchmark. In *NeurIPS*, 2025.

Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. RAVEN: A Dataset for Relational and Analogical Visual Reasoning. In *CVPR*, 2019.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing. In *NeurIPS*, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, 2018.

Shanshan Zhao, Xinjie Zhang, Jintao Guo, Jiakui Hu, Lunhao Duan, Minghao Fu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, et al. Unified Multimodal Understanding and Generation Models: Advances, Challenges, and Opportunities. *arXiv preprint arXiv:2505.02567*, 2025a.

Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, et al. Envisioning Beyond the Pixels: Benchmarking Reasoning-Informed Visual Editing. In *NeurIPS*, 2025b.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the Next Token and Diffuse Images with One Multi-Modal Model. In *ICLR*, 2025.

APPENDIX

This appendix provides benchmark construction details, experimental details, qualitative examples, and extended results supporting the main paper:

- (§A) **Additional Figures and Analyses**: generation pipeline, failure modes, and additional results on operation difficulty, scene variation, over-editing, and TINYGRAFIXBENCH; future directions.
- (§B) **Benchmark Construction**: shape library, color system, scene generation, seed design, and evaluation pipeline.
- (§C) **Experimental Details**: model specifications, inference parameters, and benchmark generation pipeline configuration.
- (§D) **Extended Results**: full per-task and per-mode results for PAINTBENCH and TINYGRAFIXBENCH.
- (§E) **Model Output Galleries**: per-problem galleries for PAINTBENCH and TINYGRAFIXBENCH, showing model outputs alongside ground-truth answers.

A ADDITIONAL FIGURES AND ANALYSES

A.1 FOUR COMMON FAILURE MODES

Beyond aggregate scores, the per-tolerance metric curves diagnose *how* a model fails on individual problems (Figure 6). **Color imprecision** appears when the edit is structurally correct but the model's output colors deviate from the target: edit-region accuracy is near zero at $\Delta E^*_{76} = 0$ but climbs steeply with tolerance, reaching high values only at lenient thresholds. **Execution omission** appears when the model fails to attempt the edit at all: edit-region accuracy stays near zero across the entire tolerance range, while preservation accuracy stays high. **Structural catastrophe** appears when the model's output bears little resemblance to either the input or the answer: all three metrics collapse to near-zero across all tolerances. **Structural imprecision** appears when the model attempts the right edit in roughly the right place but misaligns the affected region: edit-region accuracy plateaus at a moderate value that does not improve with looser color tolerance. These curve shapes provide a compact diagnostic that pinpoints failure mode from a single per-problem evaluation, surfacing patterns that aggregate mIoU scores obscure.

A.2 OPERATION DIFFICULTY AND MODEL SPECIALIZATION: ADDITIONAL RESULTS

Symbolic reasoning presents a mixed picture. Pattern is consistently hard, reaching only 13.7% (GPT-I2). Comparison (16.1% for NB-2) and counting (16.3% for NB-2) are also fairly challenging. Ordering is moderately difficult, reaching 21.0% (GPT-I2). Legend is relatively easier for closed-weights models (up to 47.1% for NB-2) but not for open-weights models (up to 6.1% for BAGEL).

Individual models depart from their own average on specific tasks. Despite leading in geometric transformation and structural manipulation, GPT-I2 scores only 1.4% on gradient, far below its 16.3% average and NB-2's 13.0%. Among open-weights models, QWEN-IE is especially proficient at flood fill (16.3%, outperforming the closed-weights NB-1; 11.4 points ahead of the next open-weights model) and comparison (12.9%). BAGEL and FLUX.2-D are also especially proficient at cropping (16.5% and 14.7%, respectively, outperforming the closed-weights NB-1). The border task has a sharp capability cliff: only NB-2 (18.9%), GPT-I2 (15.2%), and NB-1 (4.6%) score substantially; all other models score near zero.

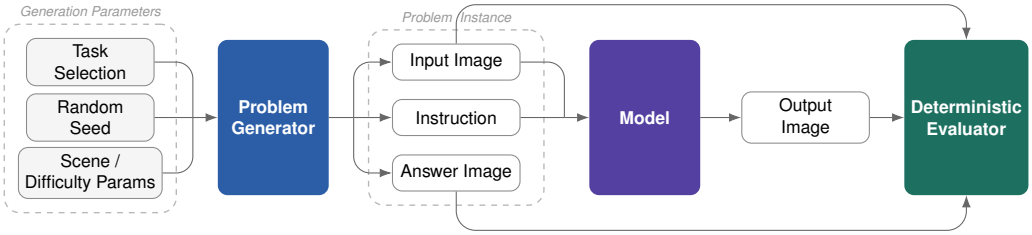


Figure 5: **Procedurally generated problems are evaluated against pixel-exact ground truth.** A random seed, task mode, and scene parameters (number of shapes, color palette, image dimensions, background style) produce a problem consisting of an input image, instruction, and answer image. The model output is then deterministically compared pixel-wise to the answer and input images, without relying on bias-prone judge models.

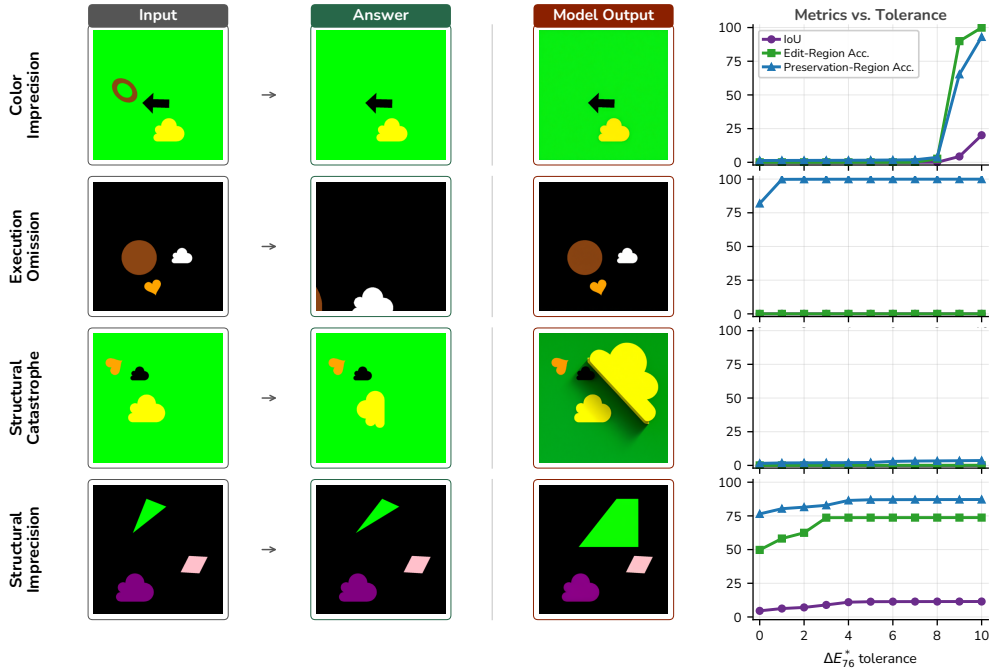


Figure 6: **Four common failure modes diagnosed by color-tolerance metric curves.** Each row shows an input, answer, and model output, along with the metric curves across ΔE^*_{76} tolerances. Curve shape pinpoints the failure mode: *color imprecision* reaches high accuracy only at high tolerances; *execution omission* keeps edit-region accuracy near zero; *structural catastrophe* collapses all three metrics to near-zero; *structural imprecision* plateaus at moderate edit-region accuracy.

A.3 Brittleness to Scene Variation: Additional Results

Non-square aspect ratios reduce closed-weights performance modestly and inconsistently. The three closed-weights models all decline on non-square canvases, but with no consistent directional preference: Nano-Banana-2 falls 2.7 points on horizontal and 0.9 on vertical; GPT-Image-2 falls 1.6 on horizontal and 3.6 on vertical (the only model with a larger vertical-axis penalty); Nano-Banana-1 falls 1.9 and 0.1 points respectively. Open-weights models show no consistent directional effect, with most slightly improving.

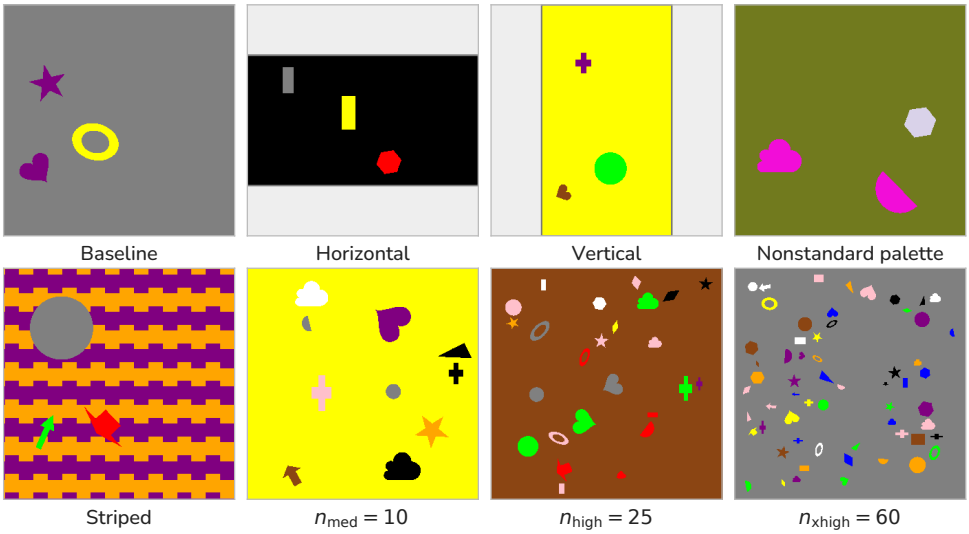


Figure 7: **Visual conditions isolate one scene parameter at a time.** Each panel shows an input image for the removal task. Starting from the baseline ($n = 3$, 1024×1024, solid background, standard palette), we vary aspect ratio, palette, background texture, or object count ($n \in \{10, 25, 60\}$).

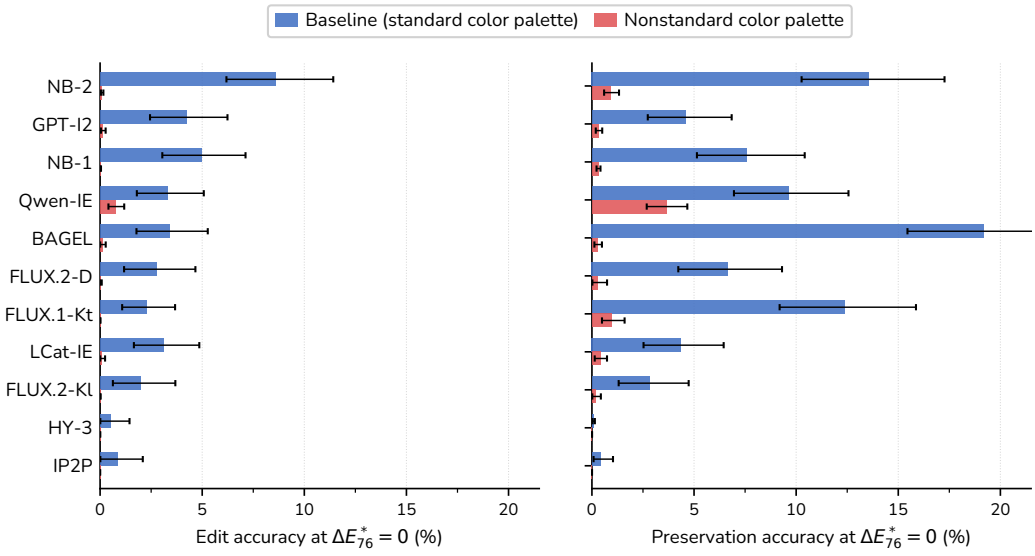


Figure 8: **Nonstandard colors substantially reduce exact pixel-match accuracy.** Edit- and preservation-region accuracies at exact pixel-match ($\Delta E^*_{76} = 0$) under baseline (standard palette) and nonstandard palettes are displayed for different models. The large accuracy gap between the two palettes reveals that models struggle to exactly reproduce nonstandard colors.

A.4 Over-Editing: Additional Results

A.5 TinyGrafixBench: Additional Results

Model-specific patterns. Per-task results in Table 3 reveal distinct model-specific patterns. A clear capability gap separates Nano-Banana-2 from Nano-Banana-1 on several tasks where Nano-Banana-2 performs well: add bar (32.3% vs. 0.3%), recolor bar (29.1% vs. 0.6%), shade interval (28.4% vs. 0.3%), and mask cells (24.1% vs. 1.2%). Among open-weights models, task differentiation is notable: BAGEL leads on filter series (10.2%, above any closed-weights model),

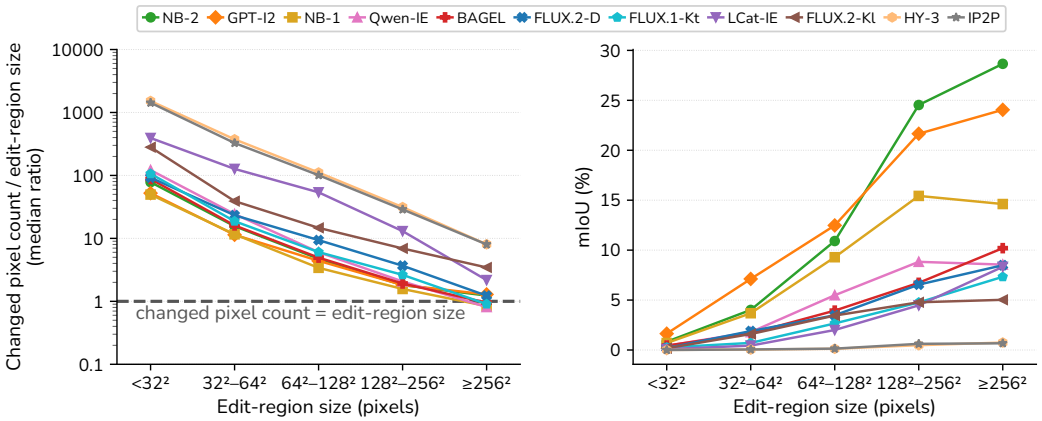


Figure 9: **Models *over-edit* relative to edit-region area, especially for smaller regions, driving worse mIoU performance.** Left: ratio of changed pixel count (at $\Delta E^*_{76} \leq 5$) to edit-region size (median ratio) vs. edit-region size. Right: mIoU (%) vs. edit-region size. Bins range from $< 32^2$ to $\geq 256^2$ pixels ($< 0.1\%$ to $\geq 6\%$ of the 1024^2 canvas). Both metrics degrade sharply for smaller edit-regions, and the pattern holds across all models (each bin contains ≥ 80 problems per model).



Figure 10: **TINYGRAFIXBENCH applies PAINTBENCH's procedural, deterministic methodology to data visualization.** Examples from `scatter`, `network`, and `heatmap` chart types (see all in Section E.2).

FLUX.1-KONTEXT-DEV on swap axes (15.7%), LONGCAT-IMAGE-EDIT on shift heatmap (18.0%), and FLUX.2-KLEIN-9B on sort bars (11.6%). Meanwhile, tasks requiring precise placement of small elements (*e.g.*, draw best-fit line, add node, draw segments) are especially hard for all models.

Table 3: **TINYGRAFIXBENCH mIoU (%) per task.** Best score per task in bold. *Plot Type Avg.* rows are macro-averages over the 4 tasks per chart type; TINYGRAFIXBENCH *Avg.* is the macro-average over 5 chart types with 95% bootstrap CIs (±; full intervals in Table 13; task descriptions in Table 7).

Task	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-KI	HY-3	IP2P
Bar Chart											
Plot Type Avg.	**38.9** ±2.3	34.8 ±2.3	11.5 ±1.4	4.8 ±1.5	2.7 ±1.1	2.3 ±0.7	1.3 ±0.4	2.9 ±0.9	2.9 ±0.6	0.2 ±0.2	0.1 ±0.1
Add Bar	**32.3**	31.9	0.3	0.6	0.0	0.0	0.0	0.2	0.0	0.0	0.0
Sort Bars	64.4	**65.8**	31.3	10.4	4.1	8.8	3.5	8.9	11.6	0.1	0.4
Remove Bar	**29.8**	13.3	13.8	8.0	6.8	0.0	1.6	2.1	0.1	0.7	0.1
Recolor Bar	**29.1**	28.2	0.6	0.1	0.0	0.3	0.0	0.5	0.1	0.0	0.0
Scatter Plot											
Plot Type Avg.	4.2 ±0.6	**6.8** ±0.7	3.7 ±0.2	1.3 ±0.2	2.1 ±0.6	4.2 ±0.4	4.3 ±0.3	1.0 ±0.3	2.1 ±0.4	0.1 ±0.0	0.3 ±0.2
Draw Best Fit Line	0.8	**1.1**	0.5	0.7	0.8	0.9	0.5	0.2	0.9	0.0	0.1
Swap Axes	9.0	**16.0**	13.1	3.1	6.9	14.7	15.7	3.4	5.4	0.3	0.9
Remove Outlier	0.4	0.3	0.3	0.5	0.5	**0.9**	0.7	0.3	0.8	0.0	0.2
Recolor Class	6.6	**9.6**	0.9	0.7	0.1	0.4	0.5	0.1	1.2	0.0	0.1
Line Chart											
Plot Type Avg.	11.5 ±1.6	**16.0** ±1.5	4.7 ±0.4	4.4 ±0.7	4.9 ±0.8	3.8 ±0.7	4.5 ±0.3	3.2 ±0.4	4.9 ±0.9	0.0 ±0.0	0.1 ±0.0
Draw Segments	**1.6**	1.0	0.4	1.0	0.6	0.9	0.4	0.3	0.6	0.0	0.1
Normalize Series	6.2	**15.5**	11.0	11.5	8.3	8.9	10.8	8.6	10.0	0.0	0.1
Filter Series	9.8	9.7	7.1	1.6	**10.2**	4.7	6.9	3.2	7.2	0.1	0.2
Shade Interval	28.4	**37.9**	0.3	3.4	0.5	0.6	0.0	0.5	1.9	0.0	0.0
Heatmap											
Plot Type Avg.	**20.2** ±2.8	15.5 ±2.5	4.0 ±0.7	4.9 ±1.3	2.0 ±0.5	3.2 ±1.2	3.5 ±0.6	7.9 ±1.4	4.8 ±0.8	0.9 ±0.3	0.0 ±0.0
Add Cell	2.6	**7.6**	0.4	0.6	0.0	1.0	0.4	0.2	1.1	0.0	0.0
Shift Heatmap	**41.1**	30.5	13.5	14.0	7.4	6.8	10.1	18.0	16.5	2.1	0.1
Mask Cells	**24.1**	17.8	1.2	4.6	0.1	4.7	3.4	13.3	0.2	1.2	0.0
Change Colormap	**13.1**	6.2	0.8	0.4	0.4	0.3	0.0	0.0	1.5	0.2	0.0
Network											
Plot Type Avg.	**4.8** ±0.6	**4.8** ±0.8	2.7 ±0.4	1.6 ±0.4	1.8 ±0.5	1.9 ±0.4	1.9 ±0.2	0.8 ±0.3	2.4 ±0.6	0.1 ±0.0	0.1 ±0.1
Add Node	0.7	**1.2**	0.5	0.6	1.0	0.3	0.3	0.1	0.6	0.0	0.1
Swap Nodes	**5.0**	**5.1**	3.8	2.1	1.6	3.1	2.8	0.5	3.4	0.2	0.2
Remove Node	**9.1**	8.6	6.3	3.5	4.6	4.0	4.6	2.5	5.6	0.1	0.3
Recolor Node	4.4	**4.5**	0.2	0.2	0.1	0.1	0.0	0.0	0.1	0.0	0.0
TINYGRAFIXBENCH Avg.	**15.9** ±0.8	15.6 ±0.8	5.3 ±0.3	3.4 ±0.4	2.7 ±0.3	3.1 ±0.3	3.1 ±0.2	3.2 ±0.3	3.4 ±0.3	0.3 ±0.1	0.2 ±0.1

A.6 FUTURE DIRECTIONS

This work focuses on 2D raster editing, but the procedurally generated, deterministically evaluated framework we advocate for can extend to a broader range of tasks with unique correct edits: scientific visualization, engineering drawing, physics and game simulation, 3D scene manipulation, and more. Our framework delivers an infinitude of PAINTBENCH problems; the infinitude of tasks awaiting this broader paradigm is a torch for future researchers to carry.

B BENCHMARK CONSTRUCTION

PAINTBENCH and TINYGRAFIXBENCH are generated entirely procedurally. Table 4 lists the 20 PAINTBENCH tasks (grouped into four categories, with task-mode breakdowns); Table 7 in Section B.7 lists the 20 TINYGRAFIXBENCH tasks. The remainder of this section documents the design choices behind both benchmarks and the shared evaluation pipeline; high-level configuration (problem counts, palette definitions, visual condition parameters) lives in Section C.

B.1 PAINTBENCH: PER-TASK DESCRIPTIONS

The four PAINTBENCH task categories are described in detail below; Table 4 above lists the 35 task-modes.

Geometric Transformation tests affine transformations of shapes. *Translation* moves a shape by a specified displacement (`amount` mode) or so that one of its named control points aligns with a named reference point elsewhere in the scene (`align` mode); the reference may be on another shape or on the canvas. *Rotation* rotates a shape by a specified angle, about a named control point on the shape itself (`local` mode) or about an external pivot (`external` mode); the external pivot is either a named reference point elsewhere in the scene or a random canvas coordinate. *Reflection* mirrors a shape across one of eight axes of its bounding box (four edges, two center lines, two diagonals; `local` mode) or across a line defined by two external points (`external` mode). *Scaling* resizes a

Table 4: **PAINTBENCH task taxonomy.** 4 categories, 20 task types, 35 task-modes. Single-mode tasks are listed without a mode qualifier.

Category	Task	Modes
Geometric Transformation	Translation	amount, align
	Rotation	local, external
	Reflection	local, external
	Scaling	amount, match
	Shearing	—
Structural Manipulation	Construction	circle, line, polygon
	Removal	attribute, location
	Copying	—
	Border	—
	Cropping	straight, tilted
Color Change	Recolor	color_code, dropper
	Flood Fill	background, foreground
	Blending	—
	Gradient	background, foreground
	Point Ops	brightness, grayscale, invert
Symbolic Reasoning	Comparison	—
	Ordering	—
	Pattern	grid, circular
	Counting	shape, color
	Legend	—

shape by a specified multiplicative factor (`amount` mode) or to match a bounding-box dimension of another shape (`match` mode). *Shearing* applies a horizontal or vertical shear relative to a fixed bounding-box edge or center line.

Structural Manipulation tests addition and removal of elements, and modification of scene composition. *Construction* places a new shape of specified type (line, circle, or polygon), color, size, and position. *Removal* deletes a shape identified by an attribute such as color or shape type (`attribute` mode) or by its canvas position (`location` mode). *Copying* duplicates a specified shape to a new canvas location. *Border* adds a colored border around a specified shape. *Cropping* extracts and upsamples a subregion of the canvas centered at a control point, either aligned to canvas axes (`straight` mode) or at an arbitrary angle (`tilted` mode).

Color Change tests manipulation of pixel color values. *Recolor* requires changing all shapes matching a description to a target color, specified either by hex code (`color_code` mode) or by naming an existing scene color to match (`dropper` mode). *Flood Fill* fills a connected region bounded by a shape with a new color; modes select whether the edit-region is the background (`background`) or foreground (`foreground`). *Blending* alpha-blends a specified color at a given opacity over a target shape region. *Gradient* applies a linear gradient inside a parallelogram region; modes select background or foreground edit-region. *Point Operations* applies a per-pixel intensity transformation (brightness adjustment, grayscale conversion, or color inversion) to selected shapes.

Symbolic Reasoning tests edits that require spatial or numerical inference before execution. *Comparison* requires identifying and removing the shape at a specified size rank (*e.g.*, the second-largest). *Ordering* rearranges all shapes of a given type along an axis in ascending or descending size order. *Pattern* completes a missing cell in a two-dimensional repeating grid, inferred from the surrounding context; modes vary between rectangular (`grid`) and circular (`circular`) arrangements. *Counting* adjusts a tally strip to match the number of shapes meeting a criterion; the model must determine the count by inspection, as the instruction does not state it; modes select whether to count by shape type (`shape`) or color (`color`). *Legend* applies a set of color substitutions and shape removals specified by a key already present in the image; the model must read the key and recolor or remove the corresponding shapes.

B.2 PAINTBENCH: SHAPE LIBRARY

PAINTBENCH scenes are composed from a vocabulary of 12 shape types, each tagged with two constraints that govern its appearance. *Rotatable* shapes may be randomly oriented; non-rotatable shapes either look identical under rotation (circle) or are semantically axis-aligned (rectangle, cross, cloud). *Width/height-free* shapes allow independent width and height scaling, so their aspect ratio varies across problems; fixed-ratio shapes always use their canonical proportions.

Table 5: **Shape library properties.** "R" = rotatable; "AR-free" = width and height may be set independently.

Shape	R	AR-free	Shape	R	AR-free
Circle	–	–	Arrow	✓	✓
Rectangle	–	✓	Heart	✓	–
Cloud	–	–	Star	✓	–
Hexagon	✓	–	Semicircle	✓	–
Triangle	✓	–	Cross	–	✓
Ring	✓	✓	Diamond	✓	✓

Each shape type also exposes a set of named control points (e.g. "center," "tip," "30-degree vertex," "arc midpoint"). Task generators use these to express instructions in terms of named, semantically meaningful locations. Every placed shape additionally inherits nine axis-aligned bounding-box control points (four corners, four edge midpoints, and the bounding-box center), which the geometric tasks draw on alongside the intrinsic control points when constructing instructions. The canvas itself contributes nine analogous reference points for instructions that target absolute image locations.

B.3 PAINTBENCH: COLOR SYSTEM

Two 11-color palettes are defined. The *standard* palette uses the exact web-color hex codes for common color names (*e.g.*, `#FF0000` for red, `#0000FF` for blue). The *nonstandard* palette uses perceptually distinct variants with uncommon hex codes (*e.g.*, gold at `#E4BA18` rather than `#FFD700`).

At generation time, the active palette is shuffled deterministically using the problem seed. The first shuffled color becomes the background fill; the second is a "holdout" color reserved for two-color striped backgrounds. The remaining nine colors populate the object color pool. The background and holdout colors are excluded from the object pool, ensuring that no shape visually merges with the background.

Table 6: **PaintBench color palettes.** The standard palette uses common web hex codes; the nonstandard palette uses perceptually distinct variants with uncommon codes.

Standard palette		Nonstandard palette	
red	#FF0000	crimson	#C31B37
orange	#FFA500	tangerine-colored	#F47B16
yellow	#FFFF00	gold	#E4BA18
green	#00FF00	olive-colored	#717A1E
blue	#0000FF	cyan	#0FE1DF
purple	#800080	lavender	#D9D2E9
pink	#FFC0CB	magenta	#F20DD8
brown	#8B4513	tan-colored	#CBAA85
black	#000000	jet black	#101211
gray	#808080	silver	#BBBCBA
white	#FFFFFF	ivory white	#F8F6E8

B.4 PAINTBENCH: BACKGROUND RENDERING

Main PAINTBENCH problems use solid single-color backgrounds. The *striped* visual condition replaces the solid fill with a two-color striped pattern alternating the background and holdout colors; parameters (orientation $\in \{0°, 45°, 90°\}$, band width $\in \{6\%, 8\%, 10\%\}$ of canvas width, and waveform sampled uniformly from `line`, `sine`, `square`, `triangle`, `sawtooth`) are randomized per problem, producing a diverse but fully deterministic set of striped patterns.

B.5 PAINTBENCH: SCENE GENERATION

A scene consists of multiple shapes drawn on a background. The same scene-generation routine is used across all PAINTBENCH tasks: shapes are placed sequentially, and for each shape the generator samples a type, color, size, aspect ratio (for AR-free shapes), rotation (for rotatable shapes), and canvas position, then rejects any candidate whose axis-aligned bounding box overlaps an already-placed shape; a 4-pixel gap margin is enforced so shape boundaries are always separated by at least a few pixels.

Shape size is sampled from a density-dependent range $[\ell_{\min}, \ell_{\max}]$ expressed as fractions of the shorter canvas dimension:

$$\ell_{\min} = \max\left(0.02,\ \frac{0.18}{\sqrt{n_{\text{mid}}}}\right), \qquad \ell_{\max} = \min\left(0.40,\ \frac{0.55}{\sqrt{n_{\text{mid}}}}\right),$$

where n_{mid} is the midpoint of the $[n_{\min}, n_{\max}]$ count range. This $1/\sqrt{n}$ scaling keeps individual shapes readable as scene density increases. For AR-free shapes, the aspect ratio is sampled log-uniformly from $[0.4, 2.5]$; rectangle, ring, and diamond shapes additionally exclude the near-square band $[0.8, 1.25]$ to prevent visual ambiguity with tilted squares.

To enforce visual diversity, no two shapes in the same scene may share the same (type, color) combination, and at most $\lceil n/3 \rceil$ shapes may share the same color.

B.6 PAINTBENCH: SEED DESIGN

Seeds are derived deterministically via SHA-256 from the string

```
"paintbench|{task}|{cond}|{mode}|{slot}|{attempt}",
```

making them immune to Python hash randomization and stable across machines. Every problem is uniquely identified by its (task, visual condition, mode, slot) tuple, and each combination is assigned its own independent seed. For each (task, condition, mode, slot), the generator searches sequentially over attempt indices until a valid scene is produced; an attempt that fails validation (for instance, due to shape placement collisions) is discarded. TINYGRAFIXBENCH uses its own seed namespace (Section B.7), so no seed-space interactions exist between the two benchmarks.

B.7 TINYGRAFIXBENCH: BENCHMARK DESIGN

TINYGRAFIXBENCH applies the same deterministic-edit framework to a different visual domain: Matplotlib-rendered analytical charts instead of synthetic shape scenes. Five chart types (`bar_chart`, `heatmap`, `line_chart`, `network`, `scatter_plot`) each expose four editing tasks, giving $5 \times 4 = 20$ task-modes; Table 7 summarizes the task catalog and Figure 10 in Section 6 shows three illustrative chart types (full per-chart galleries in Section E.2). All figures are rendered at 1024×768 pixels (160 dpi, 6.4×4.8 inches) using a bundled DejaVuSans font, so chart pixels are byte-identical across machines. The on-disk layout (input PNG, answer PNG, instruction JSON per problem) is identical to PAINTBENCH, so the same inference and evaluation pipeline applies without modification.

State-then-render architecture. Each chart module factors generation into three stages. A seeded state-builder produces a single base chart description (data, colors, labels, title, ranges) for a given slot. Each task function then takes that base state, mutates a copy to produce the desired edit, and returns *two* states (input and answer) plus a natural-language instruction. A single deterministic

Table 7: **TINYGRAFIXBENCH task catalog.** Each chart type's four tasks span the four editing operations: construction, transformation, removal, and recoloring. Brief descriptions in gray.

Chart type	Construction	Transformation	Removal	Recoloring
Bar chart	Add Bar add missing bar	Sort Bars sort bars ascending or descending	Remove Bar remove bar and label	Recolor Bar recolor one bar
Scatter plot	Draw Best Fit Line draw OLS best-fit line	Swap Axes swap x and y coordinates	Remove Outlier remove max-residual point	Recolor Class recolor points and best-fit line
Line chart	Draw Segments connect gaps using line segments	Normalize Series stretch and shift series vertically to fit in range	Filter Series clip to y values above or below threshold	Shade Interval shade area under curve within interval
Heatmap	Add Cell fill empty cell with value-corresponding color	Shift Heatmap shift cells in a direction	Mask Cells mask cells above or below threshold	Change Colormap change colormap gradient
Network	Add Node add node and incident edges	Swap Nodes swap two node positions	Remove Node remove node and incident edges	Recolor Node recolor node in graph and legend

renderer per chart type emits both the input and answer figures. This factorization is essential for evaluation: any pixel difference between input and answer comes entirely from the state edit, not from rendering nondeterminism, so small ground-truth deltas (e.g. removing one outlier point) are recoverable to the exact pixel.

Visual style sampling. Bg/text color pairs are sampled to guarantee luminance contrast (30% dark-bg/light-text, 70% light-bg/dark-text). Object colors (bars, classes, nodes, colormap endpoints) are sampled uniformly in CIE L*a*b* space, rejecting any draw within $\Delta E^*_{76} \leq 20$ of an `avoid` list that always includes the background; multi-color palettes are built one color at a time with the running set added to the avoid list, so all colors in a scene are mutually perceptually distinct. Titles, axis labels, and bar/node labels are random gibberish strings of 1–3 letter sequences each, so models cannot lean on real-world label semantics ("height of building" would prime a different distribution than "`Qkx Lpvm`"). A per-problem magnitude factor sampled from $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ scales all numerical ranges, so axis values span seven orders of magnitude across the benchmark.

Numerical-textual consistency. Every numerical value that appears in an instruction is rounded to three significant figures *before* being used both in the instruction string and in the answer state's render parameters. Without this step, the value displayed by Matplotlib (e.g. `0.123`) and the value the model is told to draw (e.g. `0.12345`) could drift, producing a small, persistent edit-region error invisible to the dataset author.

Unambiguity by construction. Each chart-edit task is constructed so a unique correct answer exists. Bar values are resampled until every pair differs by at least 5% of the y-axis maximum, so the sorted order is visually unambiguous. Line-chart gaps are placed with ≥ 2 visible vertices between them, so every input segment renders as a line (not a degenerate dot). Heatmap base states always contain at least one empty cell (NaN), so `add_cell` has a target. For scatter plots, the class with a best-fit line is constructed by drawing points along a line, then pushing one randomly chosen point by 3.5 noise standard deviations along the sign of its natural residual and clipping it back to the axis; this guarantees the maximum-residual point is the same one in both input and answer, so the "remove outlier" task has a unique solution.

Seed namespace. A SHA-256 seed derived from "`tinygrafixbench|{graph}|{task}|{slot}`" determines all chart parameters and the target transformation; no seed search is needed (every `generate_task` is constructed to always succeed), so the slot index alone identifies a problem.

B.8 Evaluation Pipeline

The evaluation pipeline is shared across PaintBench and TinyGrafixBench. Given input I, answer A, and model output O (all $W \times H$ RGB images):

Output normalization. O is rescaled (preserving aspect ratio) and center-cropped to A's resolution. We use nearest-neighbor interpolation rather than bilinear or bicubic: smooth interpolation would synthesize intermediate colors at edit boundaries that do not exist in I or A.

Change mask. $M_{\text{edit}}[p] = \mathbf{1}[I[p] \neq A[p]]$ identifies the E pixels that changed from input to answer. The remaining $P = WH - E$ pixels are *preservation* pixels (background and unchanged shapes that the model should leave untouched).

CIE76 distance. O and A are converted from sRGB to CIE L*a*b* (D65 illuminant, standard IEC 61966-2-1 piecewise linearization), and the per-pixel CIE76 distance is $\Delta E^*_{76}[p] = \|O_{\text{lab}}[p] - A_{\text{lab}}[p]\|_2$.

Per-tolerance metrics. For each tolerance $t \in \{0, 1, \ldots, 10\}$, pixel p is declared correct iff $\Delta E^*_{76}[p] \leq t$. Three metrics are computed (consistent with the CE / IE / CP / IP partition used in Section 4.2):

$$\text{Edit accuracy} = \frac{|\{p : M_{\text{edit}}[p] = 1 \wedge \Delta E^*_{76}[p] \leq t\}|}{E},$$

$$\text{Preservation accuracy} = \frac{|\{p : M_{\text{edit}}[p] = 0 \wedge \Delta E^*_{76}[p] \leq t\}|}{P},$$

$$\text{IoU} = \frac{|\{p : M_{\text{edit}}[p] = 1 \wedge \Delta E^*_{76}[p] \leq t\}|}{E + |\{p : M_{\text{edit}}[p] = 0 \wedge \Delta E^*_{76}[p] > t\}|}.$$

IoU penalizes both missed edits and erroneous modifications to preservation-regions, analogous to intersection over union for segmentation masks.

Mean-tolerance IoU. The primary reported metric is the mean IoU over all 11 tolerances $t \in \{0, \ldots, 10\}$, averaging over a range of color tolerance levels rather than committing to a single tolerance. Tolerances 0–10 span from exact pixel match ($t = 0$) to a lenient tolerance ($t = 10$).

Table 8: **Model specifications.** Eleven models evaluated on PAINTBENCH and TINYGRAFIXBENCH, spanning proprietary native multimodal generators and open-weights flow-matching and diffusion editors. All models perform image editing conditioned on a natural-language instruction.

Model	Architecture	Params	Open weights	Reference
NANO-BANANA-2	Native multimodal generator	—	—	(Google, 2026)
NANO-BANANA-1	Native multimodal generator	—	—	(Google, 2025)
GPT-IMAGE-2	Native multimodal generator	—	—	(OpenAI, 2026)
QWEN-IMAGE-EDIT-2511	Diffusion-based editor	20B	✓	(Wu et al., 2025a)
LONGCAT-IMAGE-EDIT	Diffusion-based editor	6B	✓	(Meituan LongCat Team et al., 2025)
BAGEL	Mixture-of-Transformers generator	7B	✓	(Deng et al., 2025)
FLUX.1-KONTEXT-DEV	Rectified flow editor	12B	✓	(Black Forest Labs et al., 2025)
FLUX.2-DEV	Flow-matching generator	32B	✓	(Black Forest Labs, 2025)
FLUX.2-KLEIN-9B	Flow-matching generator (distilled)	9B	✓	(Black Forest Labs, 2025)
HUNYUANIMAGE-3.0	MoE instruction-following generator	80B (13B active)	✓	(Tencent Hunyuan Foundation Model Team, 2025)
INSTRUCTPIX2PIX	Diffusion-based editor (DDPM)	1B	✓	(Brooks et al., 2023)

C EXPERIMENTAL DETAILS

C.1 MODELS

Table 8 summarizes the eleven models evaluated, spanning proprietary native multimodal generators (NANO-BANANA-2, NANO-BANANA-1, GPT-IMAGE-2) and open-weights flow-matching and diffusion editors (FLUX.1-KONTEXT-DEV, FLUX.2-DEV, FLUX.2-KLEIN-9B, QWEN-IMAGE-EDIT-2511, LONGCAT-IMAGE-EDIT, BAGEL, INSTRUCTPIX2PIX, HUNYUANIMAGE-3.0). Open-weights models are accessed under their respective published licenses:

QWEN-IMAGE-EDIT-2511, LONGCAT-IMAGE-EDIT	Apache 2.0
BAGEL	Apache 2.0
INSTRUCTPIX2PIX	MIT
HUNYUANIMAGE-3.0	Tencent Hunyuan Community License
FLUX.1-KONTEXT-DEV	FLUX.1 [dev] Non-Commercial License v1.1.1
FLUX.2-DEV	FLUX [dev] Non-Commercial License v2.0
FLUX.2-KLEIN-9B	FLUX Non-Commercial License v2.1

Proprietary models (NANO-BANANA-2, NANO-BANANA-1, GPT-IMAGE-2) are accessed via official commercial APIs under the providers' terms of service.

C.2 INFERENCE PARAMETERS

Table 9 lists the generation parameters for the eight locally-run models. The proprietary API-only models (NANO-BANANA-2, NANO-BANANA-1, GPT-IMAGE-2) use the provider's default API sampling and have no locally-controllable inference parameters. All locally-run models are evaluated with a fixed random seed. Inference was run on NVIDIA H200 GPUs via a Slurm cluster.

C.3 BENCHMARK CONFIGURATION

Construction-side details (shape vocabulary, palette definitions, scene-generation rules, seed scheme) appear in Section B; this subsection records the configuration values used in our experiments.

Problem counts. The PAINTBENCH test set contains 1,920 problems (20 tasks $\times$ 8 visual conditions $\times$ 12 problems per task-condition cell), rendered at either 1024×1024 (baseline and most conditions) or 1024×576 / 576×1024 (the *horizontal* and *vertical* aspect-ratio conditions). The baseline condition uses a single-color background drawn from the standard palette and $n = 3$ shapes for most tasks (see Table 11 for task-specific values). Visual conditions each change exactly one parameter; see Table 10 for the full enumeration. TINYGRAFIXBENCH contributes 600 problems (20 tasks $\times$ 30 problems) at 1024×768 resolution.

Visual conditions. Eight visual conditions are baked into every task of PAINTBENCH, each changing exactly one parameter relative to the baseline. Each condition receives its own independent set of seeds, so problems across conditions are drawn from distinct random scenes.

Table 9: **Inference parameters.** Generation settings for the eight locally-run models, taken from the per-run `inference_metrics` sidecar JSONs. The proprietary API-only models (NANO-BANANA-2, NANO-BANANA-1, GPT-IMAGE-2) use the provider's default API sampling and are omitted.

Model	Steps	CFG Scale
QWEN-IMAGE-EDIT-2511	50	4.0[†]
LONGCAT-IMAGE-EDIT	50	4.0[†]
BAGEL	50	4.0[‡]
FLUX.1-KONTEXT-DEV	28	3.5
FLUX.2-DEV	50	4.0
FLUX.2-KLEIN-9B	50	4.0
HUNYUANIMAGE-3.0	8[¶]	—
INSTRUCTPIX2PIX	100	7.5 / 1.5[§]

[†]Pipeline default for the QWEN-IE / LCAT-IE family (`true_cfg_scale`); not overridden in our runs.
[‡]BAGEL uses dual-CFG: `cfg_text_scale`= 4.0, `cfg_img_scale`= 2.0.
[¶]HY-3 uses 8-step distilled sampling via its own `generate_image()` pipeline; classifier-free guidance is not separately configurable.
[§]INSTRUCTPIX2PIX uses two guidance scales: text CFG = 7.5, image CFG = 1.5.

Table 10: **Visual conditions.** Each condition changes exactly one scene parameter relative to the baseline. All 20 tasks are rendered for all 8 conditions at 12 problems each, for $20 \times 8 \times 12 = 1{,}920$ problems total.

Condition	Axis	Canvas	n	Palette	Background
baseline	—	1024×1024	3	standard	solid
horizontal	aspect ratio	1024×576	3	standard	solid
vertical	aspect ratio	576×1024	3	standard	solid
nonstandard	palette	1024×1024	3	nonstandard	solid
striped	background type	1024×1024	3	standard	striped
n_{med}	object count	1024×1024	10	standard	solid
n_{high}	object count	1024×1024	25	standard	solid
n_{xhigh}	object count	1024×1024	60	standard	solid

Object counts by task. Most tasks use $n = 3$ at baseline ($n_{\text{low}} = 3$); ablation modes use $n_{\text{med}} = 10$, $n_{\text{high}} = 25$, and $n_{\text{xhigh}} = 60$. Three task groups use adjusted ranges suited to the structure of their task; Table 11 lists the exact values and justifications.

Natural-language instructions. Instructions are generated automatically from each problem's transformation parameters and fully specify the target shape(s), the operation, and all parameters needed to produce the unique answer. Each problem is saved as an input PNG, an answer PNG, and an instruction JSON sidecar that also records the seed and any task-specific metadata, enabling downstream analysis of problem characteristics beyond those reported in this paper.

Table 11: **Object counts for different tasks (n).** Most tasks use default levels; four use adjusted ranges. Object count is the only variable changed by the n_{med}, n_{high}, and n_{xhigh} conditions.

Group	**Tasks**	**baseline**	n_{med}	n_{high}	n_{xhigh}	**Justification**
Default	all other 17 tasks	3	10	25	60	Standard range covering sparse to very dense scenes.
Comparison, Ordering	comparison, ordering	3	5	7	9	Ranking n shapes requires precise size discrimination; upper cap is reduced to keep the spatial-reasoning task visually tractable.
Pattern	pattern	1	3	6	10	n indexes grid cells; starts at 1 (a single populated cell to infer from) and caps at 10 to preserve legible grid structure.
Counting	counting	5	10	25	60	Baseline raised to $n = 5$ to ensure a non-trivial count; upper range matches the default.

Table 12: **Full 95% bootstrap CIs for PAINTBENCH Category and Benchmark macro-averages.** Companion to Table 1; bootstrap procedure described in Section D.1.

	Geom. Trans.			Struct. Manip.			Color Change			Symbolic Reas.			Benchmark		
Model	Avg	Lo	Hi	Avg	Lo	Hi	Avg	Lo	Hi	Avg	Lo	Hi	Avg	Lo	Hi
NB-2	6.1	5.4	6.8	22.7	21.1	24.4	17.2	15.6	18.8	22.6	21.0	24.2	17.1	16.4	17.8
GPT-I2	11.1	10.2	12.0	24.5	23.0	26.0	13.8	12.4	15.3	15.9	14.5	17.4	16.3	15.7	17.0
NB-1	6.2	5.4	6.9	14.0	12.5	15.4	6.4	5.2	7.6	18.0	16.7	19.5	11.1	10.5	11.7
Qwen-IE	3.4	2.9	3.9	10.3	9.1	11.6	5.4	4.4	6.5	7.7	6.7	8.8	6.7	6.2	7.2
BAGEL	2.4	2.0	2.8	10.0	8.6	11.5	2.6	1.9	3.5	5.1	4.2	6.0	5.0	4.5	5.5
FLUX.2-D	3.3	2.8	3.9	8.2	7.1	9.4	2.7	2.0	3.5	4.1	3.5	4.7	4.6	4.2	5.0
FLUX.1-Kt	2.4	2.1	2.7	7.9	6.8	9.1	1.6	1.1	2.1	3.6	3.1	4.2	3.9	3.5	4.2
LCat-IE	2.2	1.9	2.6	7.1	5.9	8.3	1.8	1.2	2.4	3.5	2.9	4.2	3.6	3.3	4.0
FLUX.2-Kl	1.4	1.2	1.7	7.2	6.1	8.2	2.4	1.7	3.1	3.2	2.6	3.7	3.5	3.2	3.9
HY-3	0.1	0.0	0.1	0.8	0.5	1.2	0.2	0.2	0.3	0.3	0.2	0.4	0.4	0.3	0.5
IP2P	0.0	0.0	0.0	0.9	0.5	1.4	0.2	0.1	0.4	0.1	0.0	0.3	0.3	0.2	0.5

Table 13: **Full 95% bootstrap CIs for TINYGRAFIXBENCH Plot Type and Benchmark macro-averages.** Companion to Table 3; bootstrap procedure described in Section D.1.

	Bar Chart			Scatter Plot			Line Chart			Heatmap			Network			Benchmark		
Model	Avg	Lo	Hi	Avg	Lo	Hi	Avg	Lo	Hi	Avg	Lo	Hi	Avg	Lo	Hi	Avg	Lo	Hi
NB-2	38.9	36.6	41.2	4.2	3.6	4.8	11.5	9.9	13.1	20.2	17.4	23.1	4.8	4.3	5.4	15.9	15.1	16.7
GPT-I2	34.8	32.5	37.1	6.8	6.1	7.4	16.0	14.5	17.5	15.5	13.1	18.1	4.8	4.0	5.7	15.6	14.8	16.4
NB-1	11.5	10.1	12.9	3.7	3.5	4.0	4.7	4.3	5.1	4.0	3.3	4.8	2.7	2.4	3.1	5.3	5.0	5.7
Qwen-IE	4.8	3.3	6.4	1.3	1.1	1.5	4.4	3.7	5.2	4.9	3.7	6.2	1.6	1.2	2.0	3.4	3.0	3.8
BAGEL	2.7	1.7	3.9	2.1	1.5	2.7	4.9	4.1	5.7	2.0	1.5	2.5	1.8	1.3	2.4	2.7	2.4	3.0
FLUX.2-D	2.3	1.6	3.0	4.2	3.9	4.6	3.8	3.1	4.4	3.2	2.1	4.5	1.9	1.5	2.3	3.1	2.8	3.4
FLUX.1-Kt	1.3	0.9	1.7	4.3	4.1	4.6	4.5	4.2	4.9	3.5	2.9	4.1	1.9	1.7	2.2	3.1	2.9	3.3
LCat-IE	2.9	2.1	3.9	1.0	0.7	1.3	3.2	2.7	3.6	7.9	6.5	9.3	0.8	0.5	1.1	3.2	2.8	3.5
FLUX.2-Kl	2.9	2.3	3.6	2.1	1.7	2.5	4.9	4.1	5.9	4.8	4.0	5.7	2.4	1.9	3.0	3.4	3.1	3.8
HY-3	0.2	0.1	0.4	0.1	0.1	0.1	0.0	0.0	0.1	0.9	0.6	1.2	0.1	0.0	0.1	0.3	0.2	0.3
IP2P	0.1	0.1	0.2	0.3	0.2	0.6	0.1	0.1	0.1	0.0	0.0	0.1	0.1	0.1	0.3	0.2	0.1	0.2

D EXTENDED RESULTS

D.1 FULL BOOTSTRAP CONFIDENCE INTERVALS

Tables 12 to 14 provide the complete asymmetric percentile bootstrap confidence intervals for the Category, Plot Type, Visual Condition, and Benchmark macro-averages shown in Tables 1 to 3.

Bootstrap procedure. CIs are computed by resampling per-problem IoU values with replacement ($B = 10{,}000$ iterations, seed 0, percentile method; (Efron & Tibshirani, 1994)). All CIs in this paper share a single hierarchical methodology that mirrors each main-table average's aggregation hierarchy: per-task-mode resample → per-task pooled mean. *Category* / *Plot Type* / *Visual Condition* rows then macro-average across the inner units within that group: 5 tasks for PAINTBENCH categories, 4 subtasks for TINYGRAFIXBENCH plot types, and the 20 tasks contributing to a given condition for PAINTBENCH visual conditions. *Benchmark Avg.* rows perform one further macro step across the outer units: 4 categories for PAINTBENCH and 5 chart types for TINYGRAFIXBENCH, so each benchmark mean is a doubly-macro average in which every category / chart type contributes equally. The benchmark task lists are treated as fixed (definitional) rather than a random sample, so each interval should be read as "how much would this macro-average shift under a different draw of per-problem instances," not "how much would it shift under a different choice of tasks." Intervals are only slightly asymmetric (max 0.13% deviation from symmetry in our data), so the main paper's half-width ±X.X averages the two true endpoints; full *Lo* and *Hi* columns are reported here for readers who want the exact bracket bounds. To conserve space, the visual conditions table (Table 14) uses a narrow layout with three sub-rows per condition (**Mean**, *Lo*, *Hi*).

Table 14: **Full 95% bootstrap CIs for visual conditions.** Companion to Table 2; bootstrap procedure described in Section D.1. Object-count conditions (n_{med}, n_{high}, n_{xhigh}) use the task-group-specific levels listed in Table 11.

Condition	Stat	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Baseline	Mean	**21.9**	20.9	13.4	7.2	6.4	5.2	4.3	3.7	3.9	0.4	0.4
	Lo	19.7	19.0	11.6	5.8	5.0	4.1	3.4	2.8	3.0	0.2	0.2
	Hi	24.1	22.8	15.3	8.6	8.0	6.4	5.3	4.6	4.8	0.6	0.7
Horizontal	Mean	19.2	**19.3**	11.5	8.7	6.7	5.7	5.5	5.1	5.4	0.6	0.3
	Lo	17.2	17.5	9.9	7.3	5.4	4.7	4.6	4.0	4.3	0.3	0.1
	Hi	21.2	21.3	13.2	10.1	8.0	6.9	6.5	6.3	6.5	0.9	0.6
Vertical	Mean	**21.0**	17.4	13.3	8.0	5.0	6.4	3.8	3.5	5.7	0.5	0.2
	Lo	19.1	15.7	11.7	6.7	3.8	5.2	3.2	2.6	4.5	0.3	0.1
	Hi	22.9	19.1	15.1	9.2	6.4	7.8	4.4	4.5	6.9	0.7	0.4
Nonstandard	Mean	**23.2**	22.9	17.5	8.6	4.5	4.7	5.3	6.7	3.7	0.3	0.2
	Lo	21.2	21.0	15.6	7.1	3.5	3.7	4.3	5.5	2.8	0.1	0.0
	Hi	25.1	24.7	19.3	10.2	5.6	5.7	6.4	7.9	4.6	0.6	0.5
Striped	Mean	10.8	**12.2**	7.4	4.5	4.0	4.7	2.4	1.7	2.0	0.4	0.3
	Lo	9.7	11.0	6.3	3.8	3.2	3.9	1.9	1.3	1.5	0.2	0.1
	Hi	11.9	13.5	8.6	5.4	4.9	5.7	3.0	2.2	2.5	0.6	0.5
n_{med}	Mean	**17.9**	17.0	11.9	6.8	5.0	3.6	3.9	3.4	3.1	0.4	0.6
	Lo	16.1	15.2	10.1	5.4	3.8	2.8	2.9	2.4	2.3	0.1	0.1
	Hi	19.8	18.8	13.7	8.3	6.4	4.5	5.0	4.5	4.0	0.8	1.1
n_{high}	Mean	**13.1**	11.5	7.7	5.3	4.2	2.8	3.3	2.4	2.2	0.2	0.5
	Lo	11.6	10.0	6.5	4.1	3.0	2.2	2.4	1.5	1.5	0.1	0.1
	Hi	14.7	12.9	8.9	6.6	5.5	3.6	4.3	3.5	3.0	0.4	1.0
n_{xhigh}	Mean	**10.0**	9.6	6.3	4.6	4.2	3.4	2.5	2.6	2.3	0.2	0.1
	Lo	8.5	8.1	5.3	3.4	2.8	2.3	1.5	1.5	1.4	0.1	0.0
	Hi	11.6	11.2	7.5	5.8	5.6	4.7	3.5	3.8	3.4	0.4	0.1

D.2 Full Results Tables

The following tables provide complete per-task, and per-mode breakdowns. Table 15 covers all 35 PaintBench task-modes (three sub-tables: mean IoU, edit-region accuracy, and preservation-region accuracy); Table 18 covers all 8 visual conditions across all 20 PaintBench tasks; Table 21 covers all 20 TinyGrafixBench task-modes. Per-cell CIs are not shown to keep these full-results tables compact; aggregate-level bootstrap CIs (categories, chart types, visual conditions, benchmark averages) are reported in Section D.1.

Table 15: **PAINTBENCH Mean IoU (%) per task-mode.** Best per row in bold. Aggregate-level bootstrap CIs are reported in Section D.1.

Category	Task	Mode	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Geometric Transformation													
Category Avg.			6.1 ±0.7	**11.1** ±0.9	6.2 ±0.8	3.4 ±0.5	2.4 ±0.4	3.3 ±0.5	2.4 ±0.3	2.2 ±0.4	1.4 ±0.3	0.1 ±0.1	0.0 ±0.0
	Translation	align	10.0	**16.7**	9.2	5.1	4.0	5.3	2.1	3.7	3.6	0.0	0.0
		amount	14.6	**18.3**	10.1	5.5	3.1	3.7	3.3	4.2	2.4	0.1	0.0
	Rotation	external	11.3	**19.1**	9.6	7.4	5.5	8.2	3.8	4.3	2.2	0.1	0.1
		local	3.8	**7.3**	4.6	3.6	2.6	3.1	2.2	1.4	1.5	0.0	0.1
	Reflection	external	4.3	**8.6**	5.3	3.2	1.6	3.8	3.4	2.3	1.4	0.2	0.0
		local	4.5	**9.5**	4.9	2.7	2.0	4.1	1.9	2.7	0.7	0.0	0.0
	Scaling	amount	3.9	**10.2**	5.0	2.0	1.0	1.1	2.3	0.9	0.9	0.1	0.0
		match	2.2	**5.4**	4.2	1.8	1.1	0.6	1.3	0.4	0.7	0.3	0.0
	Shearing	—	3.1	**7.8**	4.4	1.4	1.4	1.6	1.8	1.2	0.5	0.0	0.0
Structural Manipulation													
Category Avg.			22.7 ±1.7	**24.5** ±1.5	14.0 ±1.5	10.3 ±1.3	10.0 ±1.4	8.2 ±1.1	7.9 ±1.1	7.1 ±1.2	7.2 ±1.0	0.8 ±0.3	0.9 ±0.4
	Construction	circle	18.7	**21.3**	8.2	7.3	2.0	2.9	8.6	3.4	4.3	1.0	2.5
		line	**2.2**	1.5	0.3	0.4	0.0	0.1	0.1	0.2	0.1	0.0	0.0
		polygon	**26.2**	20.1	5.9	8.4	0.9	7.2	7.1	3.5	5.0	0.8	1.1
	Removal	attribute	46.2	**54.0**	41.0	40.7	30.9	28.4	29.1	19.1	26.5	2.4	2.1
		location	45.5	**47.3**	35.3	22.8	24.0	13.9	19.1	18.3	18.7	2.2	4.2
	Copying	—	**14.0**	13.9	12.8	6.0	4.9	1.4	0.8	3.2	4.5	0.1	0.0
	Border	—	**18.9**	15.2	4.6	0.6	0.1	0.4	0.3	0.1	0.1	0.1	0.1
	Cropping	straight	25.5	**30.7**	8.6	7.1	18.1	15.8	11.3	9.2	2.2	1.0	0.2
		tilted	12.7	**26.3**	10.3	8.5	14.8	13.6	7.2	12.6	8.5	1.2	0.0
Color Change													
Category Avg.			**17.2** ±1.6	13.8 ±1.5	6.4 ±1.2	5.4 ±1.1	2.6 ±0.8	2.7 ±0.8	1.6 ±0.5	1.8 ±0.6	2.4 ±0.7	0.2 ±0.1	0.2 ±0.1
	Recolor	color code	**34.2**	30.3	10.1	13.6	8.4	8.8	4.1	3.7	8.4	0.6	0.4
		dropper	26.6	**27.6**	5.5	0.1	9.3	1.1	0.0	0.7	0.0	0.1	0.2
	Flood Fill	background	**32.6**	31.3	14.2	21.2	1.3	6.3	3.6	6.5	3.9	0.2	0.1
		foreground	17.1	**22.9**	8.2	11.5	3.4	3.6	1.6	3.0	3.7	0.4	0.3
	Blending	—	5.3	**6.4**	2.6	1.2	1.1	0.7	1.7	0.9	1.2	0.1	0.1
	Gradient	background	**25.3**	2.2	5.7	1.3	0.5	1.3	0.9	0.1	2.1	0.2	0.1
		foreground	**0.7**	**0.7**	0.1	0.1	0.0	0.2	0.2	0.0	0.1	0.0	0.0
	Point Operations	brightness	**11.1**	3.7	1.3	0.3	0.7	2.0	0.7	0.2	1.6	0.5	0.0
		grayscale	10.3	5.6	**12.1**	4.5	1.2	2.9	1.7	2.4	3.4	0.3	1.7
		invert	**15.6**	6.9	8.7	1.7	0.2	1.4	0.8	0.0	0.0	0.0	0.0
Symbolic Reasoning													
Category Avg.			**22.6** ±1.6	15.9 ±1.4	18.0 ±1.4	7.7 ±1.1	5.1 ±0.9	4.1 ±0.6	3.6 ±0.6	3.5 ±0.6	3.2 ±0.5	0.3 ±0.1	0.1 ±0.1
	Comparison	—	**16.1**	10.7	14.3	12.9	4.2	6.2	6.0	7.5	8.2	0.4	0.5
	Ordering	—	20.0	**21.0**	18.2	8.0	5.1	6.2	3.9	4.5	1.8	0.2	0.1
	Pattern	circular	**17.1**	14.7	10.7	11.8	4.0	5.8	3.0	1.3	3.9	0.6	0.0
		grid	9.7	**12.6**	6.6	4.3	4.8	5.0	1.5	0.4	2.5	0.3	0.0
	Counting	color	**14.2**	12.6	12.9	6.8	4.1	1.4	1.2	1.8	1.6	0.2	0.0
		shape	**18.4**	17.2	16.7	10.0	7.0	1.9	2.7	2.6	3.0	0.4	0.0
	Legend	—	**47.1**	19.4	34.2	1.1	6.1	1.1	3.9	2.4	0.3	0.1	0.0
Benchmark Avg.			**17.1** ±0.7	16.3 ±0.7	11.1 ±0.6	6.7 ±0.5	5.0 ±0.5	4.6 ±0.4	3.9 ±0.3	3.6 ±0.4	3.5 ±0.3	0.4 ±0.1	0.3 ±0.1

Table 16: **PAINTBENCH Edit-Region Accuracy (%) per task-mode.** Best per row in bold. Aggregate-level bootstrap CIs are reported in Section D.1.

Category	Task	Mode	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Geometric Transformation													
Category Avg.			35.9	**37.4**	23.5	21.6	17.3	16.3	19.2	21.2	12.7	3.0	2.4
	Translation	align	**41.7**	40.8	21.9	24.7	18.9	17.0	14.6	21.4	15.6	2.0	0.6
		amount	**43.4**	41.1	19.6	20.9	9.1	8.5	13.4	19.8	10.7	2.2	0.5
	Rotation	external	**58.0**	53.3	29.3	33.3	35.1	29.9	22.1	32.1	15.5	0.8	1.4
		local	**34.4**	34.1	25.4	26.9	31.4	24.1	23.9	20.7	21.0	5.3	7.1
	Reflection	external	19.7	**30.1**	18.6	19.1	14.7	18.0	18.8	19.7	15.3	1.7	1.2
		local	22.7	**30.5**	20.5	19.8	14.8	17.2	15.3	21.4	11.1	3.5	2.7
	Scaling	amount	38.3	**44.4**	26.2	18.0	7.3	11.8	22.5	19.7	9.4	3.0	3.8
		match	**32.0**	30.2	24.1	18.0	7.6	10.3	22.2	16.8	8.3	4.9	3.3
	Shearing	—	34.3	**34.9**	24.8	17.8	17.3	12.8	19.5	20.5	10.1	3.2	1.6
Structural Manipulation													
Category Avg.			48.9	**49.0**	26.1	21.5	18.5	18.0	17.0	17.4	17.5	4.8	3.3
	Construction	circle	35.2	**39.4**	13.6	11.5	4.1	4.2	15.1	5.4	8.2	4.9	4.4
		line	**8.0**	5.3	0.9	3.9	0.3	2.8	1.1	1.8	1.6	2.5	3.5
		polygon	**52.0**	40.1	10.7	16.1	1.1	18.0	23.1	10.5	11.6	4.6	5.2
	Removal	attribute	**78.6**	76.6	65.2	73.2	54.5	58.2	62.6	31.3	61.1	5.6	6.2
		location	**76.1**	73.9	58.7	49.6	42.7	33.1	42.4	36.4	35.5	6.5	11.6
	Copying	—	**56.1**	52.5	34.0	23.7	18.8	7.3	3.1	32.8	21.3	6.7	0.3
	Border	—	**45.0**	43.9	13.7	1.6	5.4	9.2	3.7	0.7	2.2	4.1	2.7
	Cropping	straight	43.0	**47.9**	10.8	8.7	19.9	20.2	15.5	11.7	3.8	3.3	0.4
		tilted	25.9	**42.2**	14.0	11.9	16.2	18.8	10.2	16.0	13.8	3.1	0.1
Color Change													
Category Avg.			**36.5**	29.6	13.4	14.3	10.8	12.9	7.8	8.6	9.3	6.0	6.4
	Recolor	color code	**66.7**	59.3	20.4	32.9	25.1	28.2	14.6	18.1	23.9	23.0	14.6
		dropper	**52.7**	51.5	14.6	4.3	23.0	7.6	4.2	5.3	4.2	7.0	7.9
	Flood Fill	background	**67.9**	60.5	32.9	44.1	20.1	43.7	24.8	20.4	25.1	6.1	5.2
		foreground	74.2	62.0	24.0	41.9	28.6	26.6	20.9	26.5	23.2	9.9	17.2
	Blending	—	**9.5**	8.7	4.5	2.3	2.0	1.4	3.0	2.2	2.3	0.9	2.2
	Gradient	background	**32.7**	17.7	7.1	4.2	2.0	3.4	1.9	0.6	4.2	1.8	1.2
		foreground	6.1	**9.6**	2.0	1.6	0.4	2.8	0.6	0.2	1.3	0.9	0.9
	Point Operations	brightness	**17.9**	5.9	2.4	0.6	2.4	3.5	1.8	1.2	3.2	3.2	0.0
		grayscale	14.9	9.0	17.8	10.7	3.1	14.5	4.7	12.6	6.1	8.3	**18.6**
		invert	**35.4**	12.1	16.5	3.1	1.3	2.8	1.2	1.4	0.0	3.5	0.1
Symbolic Reasoning													
Category Avg.			**57.0**	42.3	45.6	30.6	20.7	21.4	23.3	18.7	17.4	4.2	1.5
	Comparison	—	36.8	21.9	**46.2**	28.4	17.5	32.2	40.9	29.6	39.5	4.6	3.6
	Ordering	—	55.3	**56.3**	47.1	31.1	18.7	27.5	20.8	27.5	6.8	2.9	3.5
	Pattern	circular	**48.4**	41.3	34.3	40.0	24.6	32.4	14.8	12.6	19.0	11.7	0.2
		grid	**65.1**	58.3	27.6	21.4	24.6	29.1	10.4	7.9	18.5	7.8	0.0
	Counting	color	74.5	52.6	56.1	60.4	27.0	10.3	28.8	17.5	17.3	2.5	0.1
		shape	71.3	58.5	58.1	60.0	39.9	18.5	29.9	23.8	24.3	3.3	0.7
	Legend	—	**63.3**	27.8	46.9	2.4	9.4	2.2	13.0	5.3	1.0	0.8	0.1
Benchmark Avg.			**44.6**	39.6	27.2	22.0	16.8	17.1	16.8	16.5	14.2	4.5	3.4

Table 17: **PAINTBENCH Preservation-Region Accuracy (%) per task-mode.** Best per row in bold. Aggregate-level bootstrap CIs are reported in Section D.1.

Category	Task	Mode	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Geometric Transformation													
Category Avg.			77.7	72.2	74.9	**78.2**	72.9	64.3	70.7	50.4	53.9	12.6	4.7
	Translation	align	75.8	68.1	64.9	**76.9**	70.6	63.9	67.9	46.8	56.9	7.1	3.3
		amount	**78.8**	72.2	75.0	77.1	77.6	59.9	70.8	56.1	59.2	12.0	4.6
	Rotation	external	80.5	73.7	77.5	**81.7**	74.2	65.7	71.1	53.5	58.8	16.9	6.1
		local	78.1	68.7	74.2	**79.7**	68.2	62.0	68.4	47.4	60.0	16.3	7.0
	Reflection	external	79.8	72.5	74.9	**80.1**	75.6	66.3	73.5	55.5	58.1	13.9	7.2
		local	77.0	71.8	**78.2**	77.2	75.6	64.4	68.3	48.8	51.4	7.8	2.2
	Scaling	amount	78.0	73.2	**78.2**	75.6	71.8	63.8	74.8	49.4	60.8	18.0	7.4
		match	74.5	73.0	76.0	**79.4**	75.3	66.9	72.3	48.4	56.5	12.5	2.8
	Shearing	—	**77.3**	74.3	75.2	77.1	70.0	65.0	70.0	49.1	38.8	10.7	3.0
Structural Manipulation													
Category Avg.			75.7	66.4	74.2	**78.1**	69.4	61.2	70.1	47.4	51.9	10.1	8.8
	Construction	circle	68.8	66.7	74.6	**76.1**	70.4	59.6	62.7	46.8	49.9	15.0	1.1
		line	**80.5**	73.8	70.6	76.8	74.2	56.7	69.7	41.2	55.9	6.6	0.5
		polygon	74.3	64.6	72.2	**75.4**	73.7	57.9	54.6	49.6	54.0	2.7	0.7
	Removal	attribute	80.0	76.6	73.7	**80.5**	73.5	69.0	69.8	41.3	63.2	11.6	12.6
		location	**82.6**	80.2	79.2	78.5	81.7	67.5	73.2	49.1	69.3	14.1	20.7
	Copying	—	78.7	59.2	78.0	77.1	**80.6**	66.3	76.5	55.0	56.5	10.3	15.0
	Border	—	80.1	64.3	74.0	**80.3**	50.9	48.6	66.1	49.7	44.2	9.6	1.8
	Cropping	straight	68.4	62.3	70.8	**77.2**	63.7	61.3	74.2	39.1	35.4	8.4	12.6
		tilted	58.7	61.1	69.5	**78.2**	66.3	68.3	73.5	43.7	43.4	10.9	6.7
Color Change													
Category Avg.			**79.0**	61.4	72.0	67.4	51.4	46.2	60.5	34.1	47.9	6.1	1.9
	Recolor	color code	**78.8**	61.0	62.5	74.3	66.0	37.5	67.0	37.3	45.2	2.5	0.7
		dropper	76.9	63.6	70.0	**78.3**	69.2	50.9	70.4	45.1	57.4	11.5	4.7
	Flood Fill	background	**82.2**	69.4	76.3	73.1	28.6	22.7	40.8	40.1	26.9	4.4	0.2
		foreground	80.3	73.6	78.2	**80.9**	59.3	51.2	62.8	45.7	47.3	11.3	0.4
	Blending	—	**81.7**	69.3	74.3	76.5	59.6	53.3	61.3	30.2	48.2	5.6	0.7
	Gradient	background	**77.5**	7.8	70.3	20.5	28.2	38.1	55.6	23.6	42.2	2.4	3.5
		foreground	**72.0**	61.1	67.9	35.4	30.5	48.7	55.2	14.4	48.4	1.5	1.0
	Point Operations	brightness	79.4	64.8	73.5	**79.8**	63.7	67.2	69.0	48.6	65.6	7.6	2.1
		grayscale	78.6	69.7	74.1	**79.1**	34.6	23.8	55.0	25.8	50.2	4.7	6.3
		invert	**80.4**	74.3	71.9	79.6	71.4	67.9	72.2	37.6	57.4	11.4	1.5
Symbolic Reasoning													
Category Avg.			79.8	69.8	77.3	**80.8**	77.6	65.7	71.5	48.3	52.9	12.6	8.5
	Comparison	—	79.2	68.7	68.9	**79.6**	77.0	62.4	68.1	45.7	57.2	7.7	10.0
	Ordering	—	**80.9**	73.5	80.8	80.8	80.2	64.5	72.4	52.3	54.3	15.6	4.3
	Pattern	circular	79.7	75.0	79.6	**81.9**	71.9	67.8	70.2	42.1	54.6	14.1	11.6
		grid	75.2	59.2	76.1	75.9	**77.2**	66.1	69.3	33.4	43.5	12.0	19.2
	Counting	color	80.0	67.0	78.3	**80.8**	77.7	69.4	68.7	45.2	58.6	7.3	2.7
		shape	79.3	71.4	77.9	**81.5**	76.2	70.0	71.5	49.1	56.5	11.2	2.3
	Legend	—	81.7	70.6	80.9	**83.6**	79.3	65.1	77.0	58.7	46.4	17.2	10.1
Benchmark Avg.			**78.0**	67.5	74.6	76.2	67.8	59.3	68.2	45.1	51.7	10.3	5.9

Table 18: **Visual conditions: Mean IoU (%).** All values averaged over all 20 PAINTBENCH tasks. Best per row in bold. Object-count conditions (n_{med}, n_{high}, n_{xhigh}) use the task-group-specific levels listed in Table 11.

Condition	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Baseline	**21.9**	20.9	13.4	7.2	6.4	5.2	4.3	3.7	3.9	0.4	0.4
Horizontal	19.2	**19.3**	11.5	8.7	6.7	5.7	5.5	5.1	5.4	0.6	0.3
Vertical	**21.0**	17.4	13.3	8.0	5.0	6.4	3.8	3.5	5.7	0.5	0.2
Nonstandard	**23.2**	22.9	17.5	8.6	4.5	4.7	5.3	6.7	3.7	0.3	0.2
Striped	10.8	**12.2**	7.4	4.5	4.0	4.7	2.4	1.7	2.0	0.4	0.3
n_{med}	**17.9**	17.0	11.9	6.8	5.0	3.6	3.9	3.4	3.1	0.4	0.6
n_{high}	**13.1**	11.5	7.7	5.3	4.2	2.8	3.3	2.4	2.2	0.2	0.5
n_{xhigh}	**10.0**	9.6	6.3	4.6	4.2	3.4	2.5	2.6	2.3	0.2	0.1

Table 19: **Visual conditions: Edit-Region Accuracy (%).** All values averaged over all 20 PAINTBENCH tasks. Best per row in bold. Object-count conditions (n_{med}, n_{high}, n_{xhigh}) use the task-group-specific levels listed in Table 11.

Condition	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Baseline	**48.1**	42.8	28.1	21.6	17.8	18.5	13.3	17.1	15.2	4.9	4.2
Horizontal	**47.7**	44.1	30.0	27.6	19.2	21.1	28.6	22.3	20.3	6.5	4.2
Vertical	**49.7**	42.5	32.8	23.3	15.8	20.9	26.4	17.7	19.6	5.5	3.0
Nonstandard	**45.6**	42.1	29.8	21.0	14.4	13.6	14.5	20.9	13.0	3.3	1.5
Striped	**42.6**	40.7	25.6	25.4	23.0	23.4	18.2	15.9	15.0	5.0	3.2
n_{med}	**44.8**	37.5	27.5	21.5	17.0	13.6	13.0	16.5	11.8	4.2	4.4
n_{high}	**41.7**	34.9	24.2	18.4	15.0	13.1	12.1	9.9	10.3	3.3	2.7
n_{xhigh}	**36.4**	31.9	19.2	17.1	12.5	13.0	8.8	11.4	8.5	3.4	3.9

Table 20: **Visual conditions: Preservation-Region Accuracy (%).** All values averaged over all 20 PAINTBENCH tasks. Best per row in bold. Object-count conditions (n_{med}, n_{high}, n_{xhigh}) use the task-group-specific levels listed in Table 11.

Condition	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Baseline	**77.4**	68.2	72.1	75.5	69.0	61.1	70.6	43.5	49.1	8.8	5.9
Horizontal	**78.1**	73.4	71.7	77.9	74.1	58.1	64.2	50.1	58.4	11.5	5.5
Vertical	**79.8**	70.4	73.8	78.3	75.3	59.3	63.5	45.2	64.1	11.9	5.7
Nonstandard	76.1	68.9	76.4	**79.3**	52.7	54.3	70.3	57.4	49.2	6.1	3.1
Striped	**75.6**	63.9	74.0	70.4	67.6	62.7	59.1	37.7	40.5	12.7	9.7
n_{med}	**79.5**	68.5	74.6	76.6	67.0	60.0	71.6	43.5	53.5	10.2	6.9
n_{high}	**79.7**	64.4	76.6	77.0	68.9	60.9	74.2	42.1	49.0	10.5	4.2
n_{xhigh}	**78.2**	62.1	77.7	74.2	68.0	58.3	72.0	41.1	49.7	11.0	6.4

Table 21: **TinyGrafixBench Mean IoU (%) per task-mode.** Best per row in bold. Aggregate-level bootstrap CIs are reported in Section D.1.

Chart Type	Task	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Bar Chart												
	Plot Type Avg.	**38.9** ±2.3	34.8 ±2.3	11.5 ±1.4	4.8 ±1.5	2.7 ±1.1	2.3 ±0.7	1.3 ±0.4	2.9 ±0.9	2.9 ±0.6	0.2 ±0.2	0.1 ±0.1
	Add Bar	**32.3**	31.9	0.3	0.6	0.0	0.0	0.0	0.2	0.0	0.0	0.0
	Sort Bars	64.4	**65.8**	31.3	10.4	4.1	8.8	3.5	8.9	11.6	0.1	0.4
	Remove Bar	**29.8**	13.3	13.8	8.0	6.8	0.0	1.6	2.1	0.1	0.7	0.1
	Recolor Bar	**29.1**	28.2	0.6	0.1	0.0	0.3	0.0	0.5	0.1	0.0	0.0
Scatter Plot												
	Plot Type Avg.	4.2 ±0.6	**6.8** ±0.7	3.7 ±0.2	1.3 ±0.2	2.1 ±0.6	4.2 ±0.4	4.3 ±0.3	1.0 ±0.3	2.1 ±0.4	0.1 ±0.0	0.3 ±0.2
	Draw Best Fit Line	0.8	**1.1**	0.5	0.7	0.8	0.9	0.5	0.2	0.9	0.0	0.1
	Swap Axes	9.0	**16.0**	13.1	3.1	6.9	14.7	15.7	3.4	5.4	0.3	0.9
	Remove Outlier	0.4	0.3	0.3	0.5	0.5	**0.9**	0.7	0.3	0.8	0.0	0.2
	Recolor Class	6.6	**9.6**	0.9	0.7	0.1	0.4	0.5	0.1	1.2	0.0	0.1
Line Chart												
	Plot Type Avg.	11.5 ±1.6	**16.0** ±1.5	4.7 ±0.4	4.4 ±0.7	4.9 ±0.8	3.8 ±0.7	4.5 ±0.3	3.2 ±0.4	4.9 ±0.9	0.0 ±0.0	0.1 ±0.0
	Draw Segments	**1.6**	1.0	0.4	1.0	0.6	0.9	0.4	0.3	0.6	0.0	0.1
	Normalize Series	6.2	**15.5**	11.0	11.5	8.3	8.9	10.8	8.6	10.0	0.0	0.1
	Filter Series	9.8	9.7	7.1	1.6	**10.2**	4.7	6.9	3.2	7.2	0.1	0.2
	Shade Interval	28.4	**37.9**	0.3	3.4	0.5	0.6	0.0	0.5	1.9	0.0	0.0
Heatmap												
	Plot Type Avg.	**20.2** ±2.8	15.5 ±2.5	4.0 ±0.7	4.9 ±1.3	2.0 ±0.5	3.2 ±1.2	3.5 ±0.6	7.9 ±1.4	4.8 ±0.8	0.9 ±0.3	0.0 ±0.0
	Add Cell	2.6	**7.6**	0.4	0.6	0.0	1.0	0.4	0.2	1.1	0.0	0.0
	Shift Heatmap	**41.1**	30.5	13.5	14.0	7.4	6.8	10.1	18.0	16.5	2.1	0.1
	Mask Cells	**24.1**	17.8	1.2	4.6	0.1	4.7	3.4	13.3	0.2	1.2	0.0
	Change Colormap	**13.1**	6.2	0.8	0.4	0.4	0.3	0.0	0.0	1.5	0.2	0.0
Network												
	Plot Type Avg.	**4.8** ±0.6	**4.8** ±0.8	2.7 ±0.4	1.6 ±0.4	1.8 ±0.5	1.9 ±0.4	1.9 ±0.2	0.8 ±0.3	2.4 ±0.6	0.1 ±0.0	0.1 ±0.1
	Add Node	0.7	**1.2**	0.5	0.6	1.0	0.3	0.3	0.1	0.6	0.0	0.1
	Swap Nodes	**5.0**	**5.1**	3.8	2.1	1.6	3.1	2.8	0.5	3.4	0.2	0.2
	Remove Node	**9.1**	8.6	6.3	3.5	4.6	4.0	4.6	2.5	5.6	0.1	0.3
	Recolor Node	4.4	**4.5**	0.2	0.2	0.1	0.1	0.0	0.0	0.1	0.0	0.0
TinyGrafixBench Avg.		**15.9** ±0.8	15.6 ±0.8	5.3 ±0.3	3.4 ±0.4	2.7 ±0.3	3.1 ±0.3	3.1 ±0.2	3.2 ±0.3	3.4 ±0.3	0.3 ±0.1	0.2 ±0.1

Table 22: **TinyGrafixBench Edit-Region Accuracy (%) per task-mode.** Best per row in bold. Aggregate-level bootstrap CIs are reported in Section D.1.

Chart Type	Task	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Bar Chart												
	Plot Type Avg.	**70.5**	58.1	28.1	10.9	6.4	3.3	6.8	6.8	4.0	2.0	0.4
	Add Bar	**59.9**	53.1	1.2	0.9	0.0	0.1	0.0	0.7	0.0	0.0	0.0
	Sort Bars	74.0	**74.4**	40.7	12.8	5.1	10.8	5.8	13.3	14.8	0.1	0.7
	Remove Bar	**81.9**	53.7	67.8	29.8	20.4	0.1	21.5	11.2	0.4	7.7	0.9
	Recolor Bar	**66.2**	51.3	2.9	0.2	0.0	2.0	0.0	1.9	0.6	0.0	0.0
Scatter Plot												
	Plot Type Avg.	22.1	21.2	23.5	17.7	13.5	23.7	**34.2**	14.0	18.6	0.8	7.7
	Draw Best Fit Line	5.7	**6.0**	5.6	5.4	5.3	4.6	5.2	2.6	5.1	0.6	0.7
	Swap Axes	23.0	35.3	41.3	8.1	13.3	24.7	**41.8**	12.3	13.6	1.0	2.4
	Remove Outlier	29.0	12.5	40.5	53.1	35.2	61.4	**85.9**	40.1	52.1	1.4	27.0
	Recolor Class	30.6	**30.9**	6.8	4.1	0.3	4.1	3.9	1.1	3.8	0.3	0.8
Line Chart												
	Plot Type Avg.	40.2	**44.2**	33.3	22.0	25.3	16.6	32.3	21.7	23.7	1.0	1.0
	Draw Segments	**20.0**	11.2	9.4	16.0	6.9	8.1	8.3	7.1	9.3	1.0	1.7
	Normalize Series	18.3	41.4	45.7	**46.3**	22.7	29.0	45.1	35.4	28.6	0.6	0.5
	Filter Series	70.9	71.7	**76.9**	16.1	70.8	27.9	75.8	43.1	52.7	2.2	1.7
	Shade Interval	51.6	**52.5**	1.2	9.5	0.8	1.5	0.0	1.2	4.4	0.0	0.1
Heatmap												
	Plot Type Avg.	**27.3**	21.0	5.3	7.3	2.1	4.9	8.4	14.1	5.7	1.6	0.1
	Add Cell	7.7	**15.4**	2.8	5.3	0.1	2.6	10.5	3.9	2.4	0.5	0.0
	Shift Heatmap	**43.6**	32.2	15.5	15.7	7.8	7.1	13.9	20.7	18.0	2.7	0.1
	Mask Cells	**41.9**	28.6	2.1	7.8	0.2	9.5	9.1	31.6	0.3	2.9	0.0
	Change Colormap	**15.9**	7.8	0.9	0.4	0.5	0.5	0.0	0.1	2.2	0.4	0.1
Network												
	Plot Type Avg.	**33.6**	22.5	22.7	8.3	7.6	10.3	26.0	6.6	12.1	0.8	1.5
	Add Node	6.3	**8.6**	6.0	4.5	5.0	2.9	4.5	2.2	4.3	0.4	1.8
	Swap Nodes	20.8	15.7	24.0	7.9	3.8	10.8	**26.4**	3.1	12.0	1.6	1.2
	Remove Node	60.0	41.1	57.6	19.0	21.0	26.7	**71.9**	20.9	30.5	1.2	2.9
	Recolor Node	**47.4**	24.5	3.4	1.7	0.5	0.8	1.4	0.2	1.5	0.1	0.0
TinyGrafixBench Avg.		**38.7**	33.4	22.6	13.2	11.0	11.8	21.6	12.6	12.8	1.2	2.1

Table 23: **TinyGrafixBench Preservation-Region Accuracy (%) per task-mode.** Best per row in bold. Aggregate-level bootstrap CIs are reported in Section D.1.

Chart Type	Task	NB-2	GPT-I2	NB-1	Qwen-IE	BAGEL	FLUX.2-D	FLUX.1-Kt	LCat-IE	FLUX.2-Kl	HY-3	IP2P
Bar Chart												
	Plot Type Avg.	78.5	70.8	75.1	**79.5**	72.4	58.8	58.5	43.6	63.9	23.5	10.9
	Add Bar	**79.1**	71.8	76.2	78.7	72.6	51.8	63.1	54.2	67.6	20.7	6.5
	Sort Bars	**78.5**	73.8	75.1	77.8	72.3	66.9	57.7	53.5	64.3	23.2	6.1
	Remove Bar	78.2	65.9	73.6	**82.0**	70.3	67.2	55.6	38.2	70.2	22.6	19.7
	Recolor Bar	78.4	71.5	75.7	**79.4**	74.1	49.4	57.7	28.8	53.6	27.6	11.2
Scatter Plot												
	Plot Type Avg.	79.6	73.3	81.0	**83.9**	75.5	64.4	80.0	38.0	74.0	15.1	17.9
	Draw Best Fit Line	80.4	72.0	79.4	**83.3**	77.8	72.5	76.2	39.5	75.6	9.0	10.8
	Swap Axes	77.4	74.8	82.3	**84.0**	81.4	73.1	81.0	37.8	75.5	18.2	9.5
	Remove Outlier	81.5	75.1	82.0	**84.6**	73.0	73.6	81.0	44.7	79.7	18.4	27.6
	Recolor Class	79.0	71.5	80.4	**83.5**	70.0	38.3	81.9	30.2	65.0	15.0	23.9
Line Chart												
	Plot Type Avg.	**77.6**	71.4	74.5	75.3	75.1	61.0	71.5	59.0	68.3	5.1	13.4
	Draw Segments	79.6	71.6	79.5	**81.0**	78.9	69.8	77.3	60.9	71.5	3.9	11.3
	Normalize Series	**81.1**	73.2	80.1	79.6	75.9	63.6	79.8	63.7	76.4	2.1	4.9
	Filter Series	77.3	70.3	75.4	75.9	77.0	67.8	**77.6**	61.8	70.6	9.1	23.5
	Shade Interval	**72.3**	70.4	62.7	64.6	68.7	42.8	51.1	49.4	54.7	5.2	13.8
Heatmap												
	Plot Type Avg.	68.3	65.3	70.4	**72.0**	61.6	54.6	28.5	34.7	53.5	12.3	12.9
	Add Cell	77.1	74.2	72.6	**78.1**	68.7	62.0	31.1	31.1	63.6	12.4	12.6
	Shift Heatmap	**75.2**	73.2	68.4	71.2	74.7	71.3	32.5	46.5	55.0	21.1	9.0
	Mask Cells	63.2	58.9	**69.0**	67.7	58.0	53.8	31.5	26.1	62.5	10.0	25.8
	Change Colormap	58.0	55.1	**71.7**	71.2	45.2	31.2	18.8	35.0	32.7	5.6	4.4
Network												
	Plot Type Avg.	80.1	71.2	81.9	**84.5**	79.5	58.8	75.0	36.8	75.7	10.8	14.8
	Add Node	76.5	69.9	81.6	**83.6**	78.2	63.6	74.8	43.9	74.1	4.4	21.0
	Swap Nodes	80.2	73.1	82.3	**85.4**	79.3	71.3	73.3	31.0	78.3	14.8	25.7
	Remove Node	81.6	74.9	82.3	**85.3**	81.1	68.6	74.3	46.0	75.9	13.7	7.4
	Recolor Node	82.1	67.0	81.6	**83.7**	79.2	31.9	77.6	26.3	74.6	10.1	5.0
TinyGrafixBench Avg.		76.8	70.4	76.6	**79.0**	72.8	59.5	62.7	42.4	67.1	13.4	14.0

E Model Output Galleries

This section presents model output galleries for PaintBench (Section E.1) and TinyGrafixBench (Section E.2), showing model outputs alongside ground-truth answers for representative problems.

E.1 Per-Problem Galleries: PaintBench

Figures 11 to 14 show one representative problem per PaintBench category alongside the outputs of all eleven models. Each figure shows the input image, instruction, answer, and one output per model.

E.2 Per-Problem Galleries: TinyGrafixBench

Figures 15 to 19 show one representative problem per chart type across the five TinyGrafixBench chart families. Each figure shows the input image, instruction, answer, and one output per model.

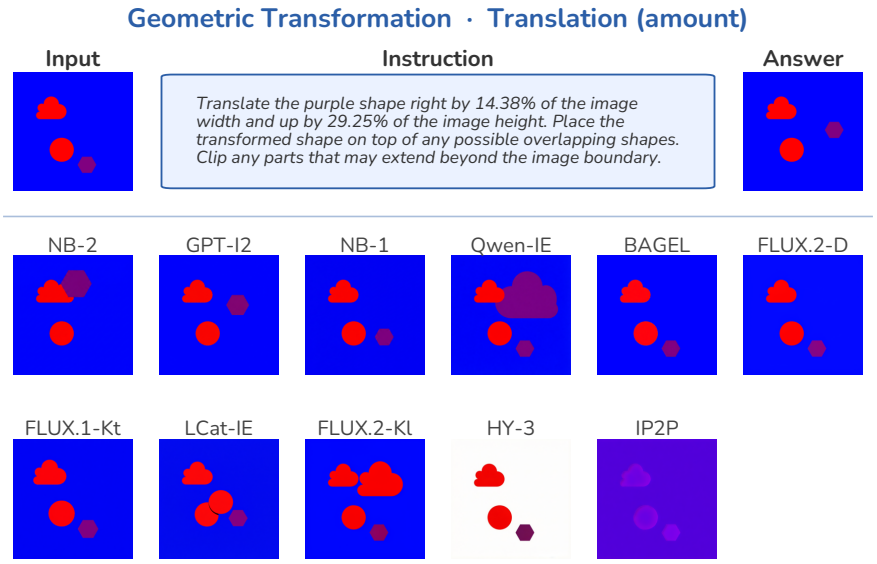


Figure 11: **PAINTBENCH gallery: Geometric Transformation.**

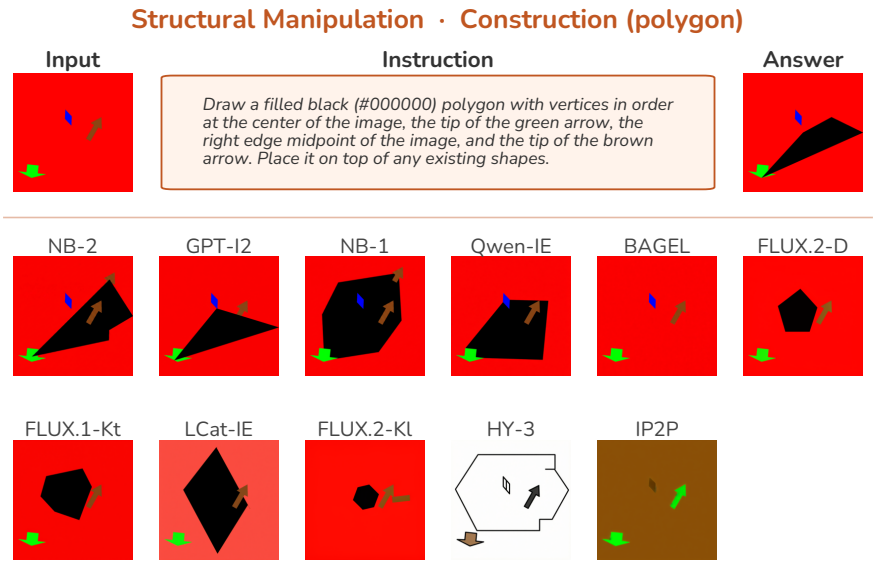


Figure 12: **PAINTBENCH gallery: Structural Manipulation (construction task, polygon mode).** Input, ground-truth answer, and outputs from all eleven models.

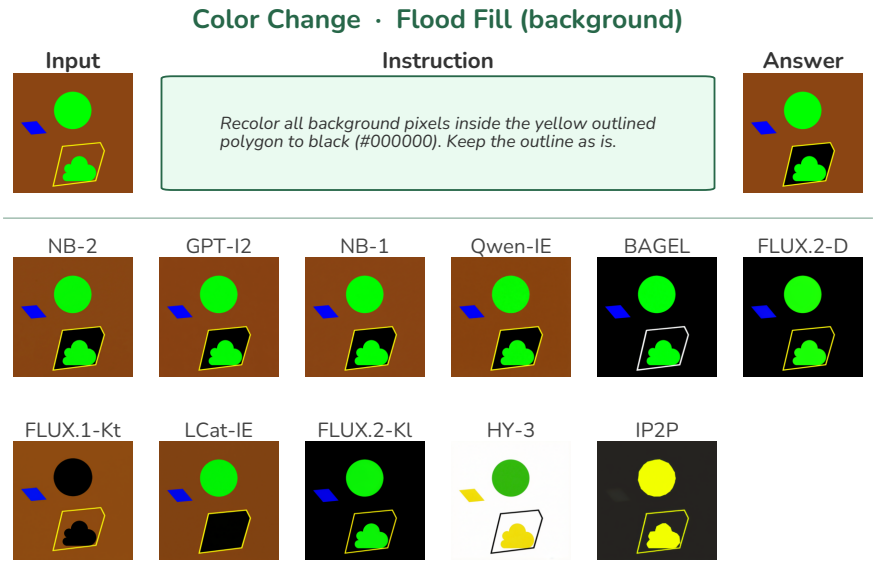


Figure 13: **PAINTBENCH gallery: Color Change (flood fill task, background mode).** Input, ground-truth answer, and outputs from all eleven models.

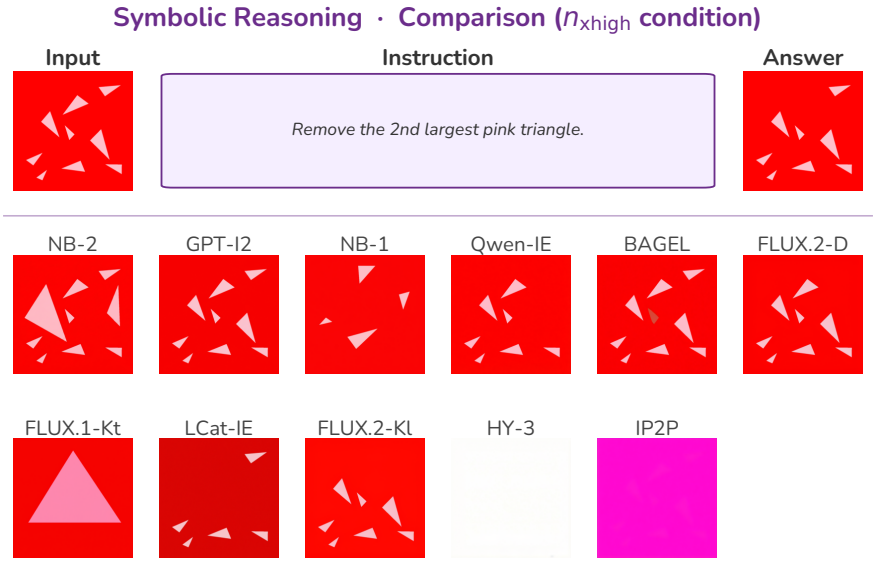


Figure 14: **PAINTBENCH gallery: Symbolic Reasoning (comparison task, $n_{\mathbf{xhigh}}$ condition).** Input, ground-truth answer, and outputs from all eleven models. *N.B.,* HY-3 *outputs a white image.*

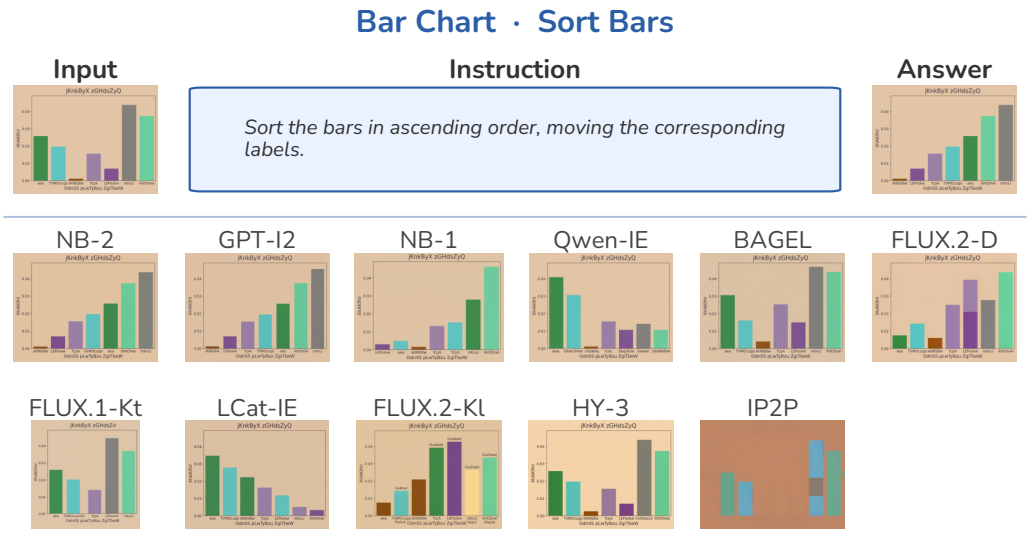


Figure 15: **TINYGRAFIXBENCH gallery: Bar Chart (sort bars task).**

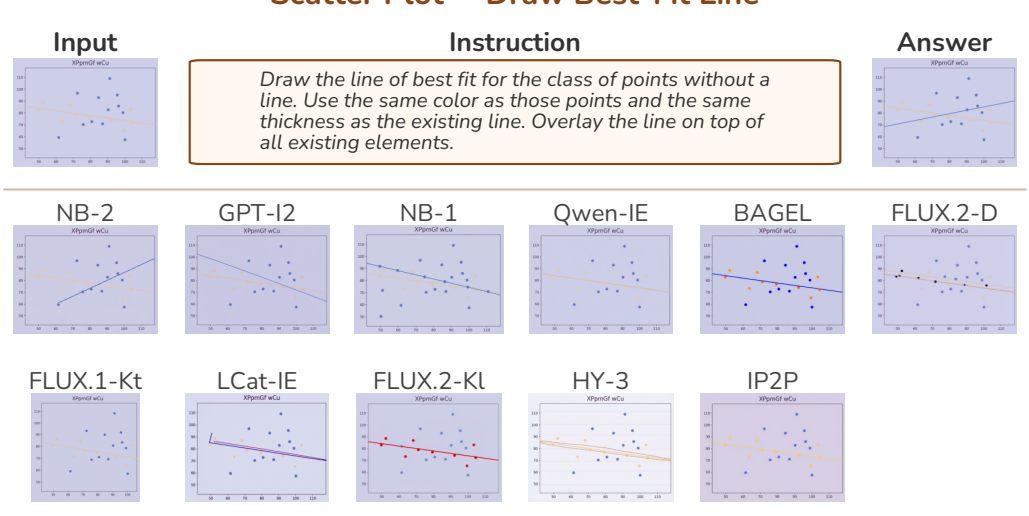


Figure 16: **TINYGRAFIXBENCH gallery: Scatter Plot (draw best-fit line task).**

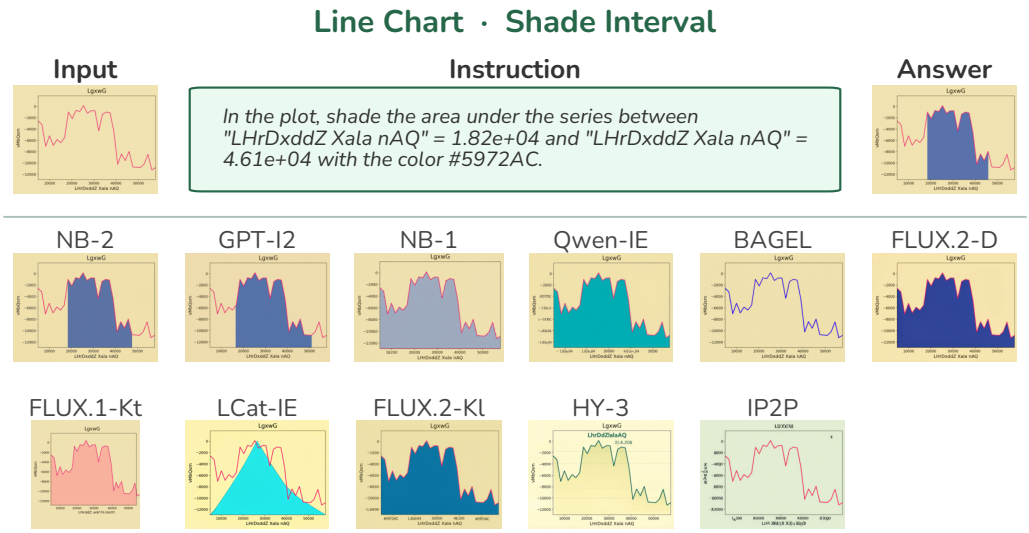


Figure 17: **TINYGRAFIXBENCH gallery: Line Chart (shade interval task).**

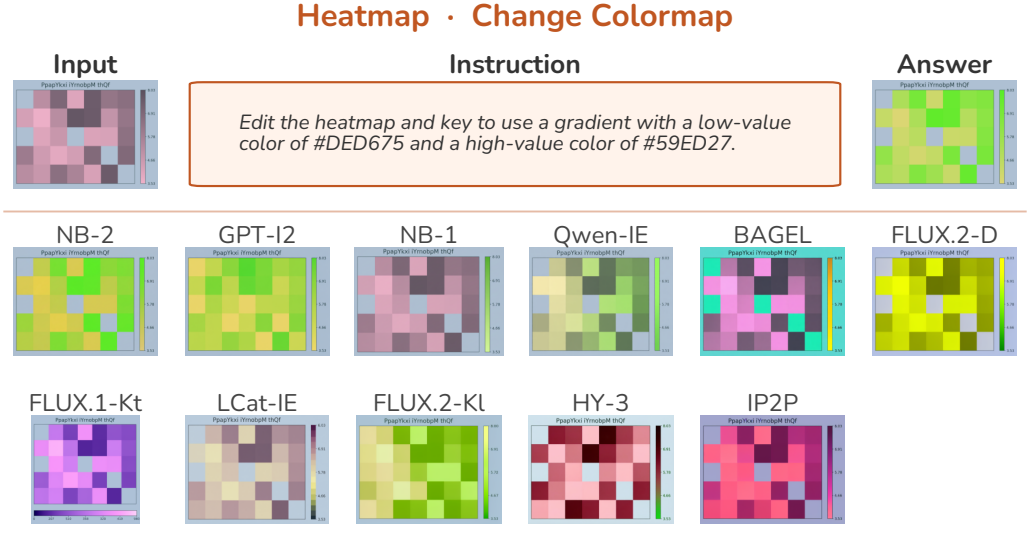


Figure 18: **TINYGRAFIXBENCH gallery: Heatmap (change colormap task).**

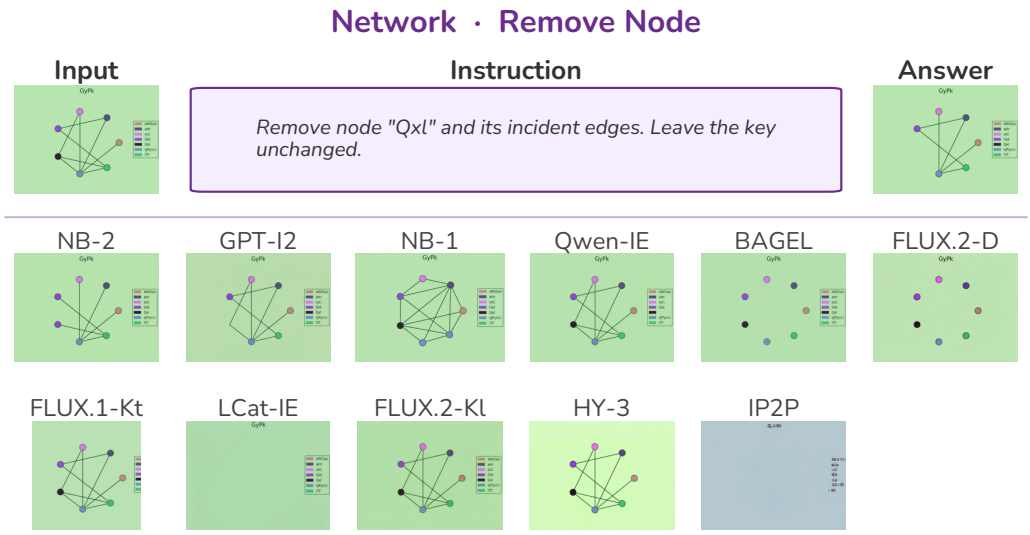


Figure 19: **TINYGRAFIXBENCH gallery: Network (remove node task).**