# PPTBench: Can Coding Agents Reconstruct the Visual World through Structured, Editable Slides?

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

## Abstract

Coding agents are increasingly moving beyond text-based software tasks to reconstruct visual targets through code. This capability, visual coding, requires agents to translate their understanding of visual targets into executable code. Slides provide a natural testbed for this capability, combining rich visual structure with objects that can be programmatically created and edited. To measure this capability, we introduce **PPTBench**, a benchmark for reconstructing scientific flow diagrams as editable PowerPoint slides. PPTBench covers 500 scientific flow-diagram tasks across 10 presentation domains, drawn from real research papers, and is evaluated with a four-stage agentic judge covering artifact validity, process and connector fidelity, rendering quality, and visual fidelity. Evaluation of ten models across 36 model–harness–effort configurations shows a substantial gap in reliable visual coding: the best configuration achieves 77.34, while the median across configurations is 24.38. Fine-grained analysis shows that agents can generally produce valid slide files, but struggle to produce high-quality reconstructions that faithfully recover the semantics and visual structure of the target. Further analysis shows that increasing reasoning effort primarily improves hard-gate passage rather than mean detail quality on each configuration's passing tasks, while configurations with more inspection tend to achieve higher overall scores. PPTBench establishes a measurable testbed for studying visual coding and advancing agents toward more reliable visual creation.

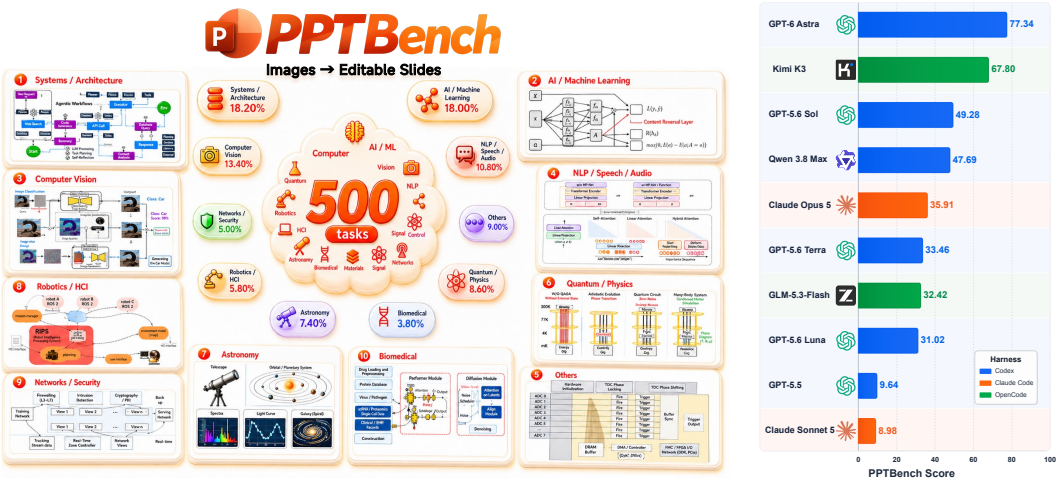



Figure 1: PPTBench task domains and representative model scores on 500 scientific flow diagrams.



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.

# 1 INTRODUCTION

Coding agents are beginning to act in the visual world. Instead of producing software evaluated mainly through APIs or terminals, they are increasingly asked to create webpages, interfaces, games, 3D scenes, diagrams, and documents (Li et al., 2025b; Chi et al., 2026a; Hu et al., 2024; Belouadi et al., 2024b; Ge et al., 2025), requiring them to think through code as a medium for expressing visual ideas. These tasks require more than generating valid programs: agents must infer the visual structure of a target and faithfully represent it in an executable form. This creates a fundamental challenge in aligning visual understanding with programmatic representation.

Slides provide an ideal testbed for the visual coding task above. A slide is not simply an image, but an editable composition whose elements have distinct types and spatial relationships. Its shapes, text, diagrams, and layout form a compact visual language for organizing and communicating information. Slides are also a pervasive medium for human knowledge: people use them to explain ideas, present research, teach concepts, and communicate decisions. Reconstructing a slide therefore goes beyond reproducing its appearance; it requires an agent to recover how visual elements are organized to convey meaning and express that structure in an editable artifact.

Existing benchmarks evaluate different parts of this capability. Slide-generation suites (Zheng et al., 2025; Yang et al., 2026; Chen et al., 2026) start from documents or requests and evaluate open-ended designs. Chart-to-code and screenshot-to-markup benchmarks (Yang et al., 2025; Wu et al., 2024; Tang et al., 2025a; Zhang et al., 2026a; Si et al., 2024; Laurençon et al., 2024; Li et al., 2025b) reconstruct fixed targets as code or markup. Office benchmarks such as PPTC (Guo et al., 2023) evaluate instructed edits to PPTX files, while Slide2Code (Tang et al., 2025b) directly evaluates image-to-editable-slide reconstruction. The earlier benchmark named PPTBench (Huang et al., 2025) spans slide detection, understanding, modification, and generation, including generation from screenshots. Our PPTBench focuses on scientific process diagrams and tests whether native editable reconstructions preserve their process and connector structure. Its evaluation gates artifact validity, process fidelity, and legibility before scoring visual detail, separating structural failures from appearance errors.

To rigorously measure and advance this capability, we introduce **PPTBench**, a benchmark for visual coding through editable slide reconstruction. Its 500 tasks are drawn from scientific flow diagrams in real arXiv papers, with each task requiring an agent to reconstruct a given diagram as a single-page PowerPoint presentation composed of native, editable objects. The benchmark combines automated checks with human screening to ensure that the tasks are visually meaningful and structurally suitable for reconstruction. To enable automated evaluation, PPTBench employs a four-stage Agentic Judge. It first checks file validity and native editable content, screening raster images for copies of the reference. It then evaluates whether the reconstructed objects preserve the semantics of the target, whether the rendered slide is visually and textually legible, and whether individual elements match the target in their geometry and textual content. The four stages form a gated evaluation, so fundamental failures cannot be hidden by strong performance on other aspects. The score is computed from structured findings, and blinded validation shows substantial agreement with human judgments.

We evaluate ten models across 36 model–harness–effort configurations on all 500 tasks, yielding 18,000 scored reconstructions. The best configuration, GPT-6 Astra with high reasoning effort, reaches 77.34 out of 100, while the configuration median is 24.38, revealing a substantial gap in reliable visual coding. Agents usually produce usable slides, but only 31.74% of reconstructions pass all gates, with semantic recovery as the dominant bottleneck. Text accounts for 53.0% of detail deductions, with unintended wrapping as the largest source. We further find that reasoning effort mainly improves hard-gate pass rates, while inspection frequency is more strongly associated with higher scores than revision frequency. These findings distinguish file validity from faithful reconstruction of a diagram's process and visual structure.

In summary, the primary contributions of this paper are:

1. We identify the core challenge of visual coding as recovering visual structure in executable representations, and choose editable slides as a suitable medium for measuring this capability.
2. We introduce PPTBench with 500 diverse scientific flow-diagram reconstruction tasks drawn from real arXiv papers, together with an automated four-stage Agentic Judge that verifies artifact validity, semantic correctness, rendering quality, and fine-grained visual fidelity.

3. Through extensive experiments on frontier models, we uncover systematic limitations of current agents, such as gaps in semantic recovery and visual fidelity, and provide directions for advancing visual coding agents.

## 2 THE PPTBENCH BENCHMARK

PPTBench asks whether an agent can reconstruct a scientific flow diagram as a native, editable PPTX slide from a single image. We first define the task and artifact contract, then describe the benchmark construction and the four-stage evaluation protocol.

### 2.1 TASK FORMULATION

A diagram reconstruction task provides an agent with a reference image $I$ and requires it to generate a single-slide PPTX artifact $P = A(I)$. The reference image $I$ is a rasterized rendering of a flow diagram from a real research paper and serves as the agent's only observation of the target. The generated artifact must recover both the visual appearance and structural semantics of the diagram while remaining natively editable. Specifically, the output should be composed of native PowerPoint objects, except for explicitly approved raster elements that cannot be faithfully represented with native primitives.

The native-object requirement prevents trivial solutions such as directly embedding the reference image into the slide, ensuring that the task evaluates structural reconstruction rather than image copying. The evaluator renders the generated PPTX with a fixed rendering environment, producing a rendered image $\hat{I} = R(P)$, and compares it with the reference image $I$.

### 2.2 BENCHMARK CONSTRUCTION AND COVERAGE

This section describes where candidate diagrams come from, how they are prepared and screened into a fixed set of reconstruction tasks, and which scientific domains the PPTBench covers. Figure 2 summarizes the construction process, from source retrieval and preprocessing to human screening and task assembly.

**Figure sources.** We query arXiv metadata and download source archives, then parse LaTeX `figure` environments to locate the authors' original graphics files. Captions, file paths, and asset properties help us identify flow, system-architecture, and module diagrams. Using published figures exposes agents to the irregular layouts, mixed notation, and visual conventions that researchers use to explain their work. We retain each candidate's paper identifier, title, version, figure number, and related metadata so that every task can be traced back to its source.

**Preprocessing.** We render each candidate directly from its original asset and record the source identity and rendering parameters for reproducibility. Embedded raster blocks are inspected separately. Self-contained elements that cannot reasonably be recreated with native PowerPoint primitives may be approved for reuse, while the rest of the diagram must be rebuilt as editable objects. We then remove repeated figures and alternate exports within each paper. Filtering and preprocessing leave 1,000 candidates for human screening (Figure 2).

**Human screening.** Annotators review the 1,000 candidates and retain 500 using five exclusion criteria. We remove figures outside the flow and architecture categories and diagrams whose content and logic are too simple to distinguish model capabilities. We also exclude references with insufficient resolution or illegible text, as well as those with cropping errors, source corruption, or rendering defects. Finally, annotators must be able to consistently identify the key connections, arrow directions, and node boundaries. If the reference itself leaves an arrow's direction unclear, a reconstruction cannot be judged reliably against it. Together, these criteria select diagrams that are challenging enough to reveal differences between systems and clear enough to evaluate. The resulting 500 tasks form the fixed evaluation set.

**Task inputs.** Each task provides a reference image and, where needed, manually approved raster assets. Agents inspect and measure the reference, then produce a single-slide reconstruction with

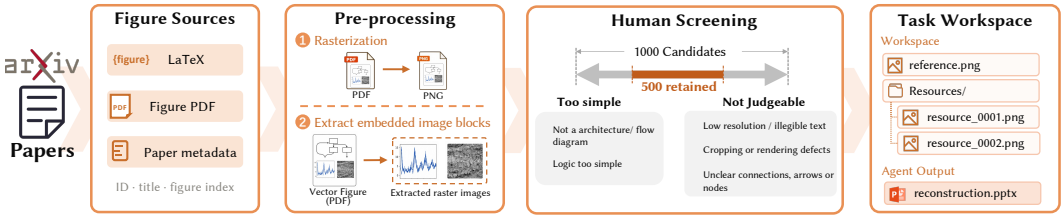



Figure 2: PPTBench construction: source retrieval, preprocessing, human screening, and task assembly. Screening retains 500 of 1,000 candidate diagrams.

editable text, shapes, and connectors. Approved raster assets must appear unchanged in their intended locations. The agent receives task-specific inputs without source-paper provenance or access to other tasks.

**Benchmark composition and coverage.** PPTBench draws from multiple scientific fields to evaluate reconstruction across different visual structures and conventions. Its 500 tasks span 50 arXiv primary categories, grouped into 10 representative domains in Figure 1. For example, systems workflows use branches and feedback loops whose connections determine the process, while neural architectures combine repeated modules with dense mathematical labels. Quantum circuits introduce a different spatial convention, where gate placement along aligned wires encodes the order of operations. These examples require agents to recover both the visual arrangement and the relationships it expresses across different diagram conventions. Detailed category counts and source examples are provided in Appendix B.

### 2.3 Evaluation Protocol

Each reconstruction receives a score from 0 to 100, combining three validity gates with visual detail quality (Figure 3):

$$S = \underbrace{G_{\text{art}}}_{\text{Stage 0}} \cdot \underbrace{G_{\text{sem}}}_{\text{Stage 1}} \cdot \underbrace{G_{\text{rt}}}_{\text{Stage 2}} \cdot \underbrace{(S_{\text{layout}} + S_{\text{text}} + S_{\text{graphics}})}_{\text{Stage 3}}. \tag{1}$$

Artifact validation is deterministic, and GPT-5.6 Luna High evaluates Stages 1–3 in three independent rounds. The binary gate $G_{\text{art}}$ records artifact validity. $G_{\text{sem}} = 1$ requires at least two rounds to pass Stage 1; $G_{\text{rt}} = 1$ requires at least two of those rounds to also pass Stage 2. Each gate is zero otherwise, so a candidate must pass both judged gates in at least two rounds to receive a detail score. Findings from passing rounds are merged and converted into scores by deterministic rules. Layout, text, and local graphics contribute up to 30, 40, and 30 points, respectively, with normalized detail quality $Q = (S_{\text{layout}} + S_{\text{text}} + S_{\text{graphics}})/100$.

The judge can write cropping, color-segmentation, and geometric-measurement scripts to inspect the reference and reconstruction. Visual inspection checks the depicted process even when equivalent diagrams use different PowerPoint object structures. Section 3.4 evaluates gate decisions against blinded human labels and independent repetitions.

**Stage 0: artifact validity.** The evaluator checks that the PPTX can be parsed and rendered, contains exactly one slide and at least three native editable components, and has no external relationships, embedded fonts, attachments, or ActiveX payloads. Missing or invalid artifacts receive zero. The validator checks object bounds and screens raster images for copies of the reference. Valid slides are rendered and paired with the reference after cropping outer whitespace and normalizing scale and centering. Full-resolution images remain available for inspection; Appendix C.2 gives the image preparation details.

**Stage 1: semantic correctness.** The judge compares nodes, labels, and directed connections, checking for missing, added, or incorrectly merged steps, wrong endpoints or directions, altered branches, merges, loops, or cross-panel links, and label errors that change meaning. It also traces local connector segments and intermediate arrowheads, so a missing segment or moved arrowhead can fail this stage even when the overall process looks plausible. A confirmed discrepancy ends the round, with the affected node or connection recorded as the reason.

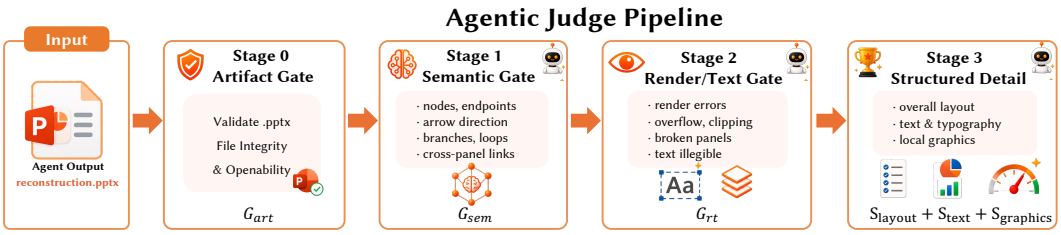



Figure 3: The four-stage Agentic Judge: artifact validation ($G_{\text{art}}$), semantic correctness ($G_{\text{sem}}$), rendering and text quality ($G_{\text{rt}}$), and visual detail. Stages 1–3 run in three independent rounds. At least two rounds must pass both judged gates before merged detail findings determine the score (Eq. (1)).

**Stage 2: rendering and text quality.** After the semantic gate passes, the judge checks legibility. This gate triggers when black regions, distortion, tearing, severe ghosting, or blank blocks affect more than 30% of the figure, or when more than 50% of text boxes overflow, overlap, or break improperly. A failure ends the round. Localized defects proceed to Stage 3 for graded deductions.

**Stage 3: fine-grained visual quality.** The judge compares layout and composition, text and typography, and local graphics such as nodes and connectors. It records defects and their extent without assigning points. Findings from passing rounds are merged by issue type within each category and dimension, retaining the largest reported affected fraction. A fixed rubric then deducts points according to severity and extent, up to each dimension's score cap. Outer whitespace, global centering, and absolute content scale do not affect the score. Appendix I gives the finding schema, merging rules, and deduction formulas, and Appendix H provides scored examples.

## 3 Diagnosing Visual Coding Capabilities on PPTBench

### 3.1 Setup and Metrics

We evaluate ten models across 36 model–harness–effort configurations (Table 1). Each configuration runs all 500 tasks under the task and evaluation protocol in Section 2, yielding 18,000 scored reconstructions. Models use Codex, Claude Code, or OpenCode as listed in the table, so results reflect both the model and its harness. The evaluation covers the full benchmark, including the tasks used for protocol development and judge validation.

Our main metric is mean $S$ over all 500 tasks, with invalid or gated outputs scored as zero. Artifact validity and gate-pass rate measure the fractions passing Stage 0 and all gates, respectively. Score on passed tasks is the mean score computed only over reconstructions that pass all gates; multiplying it by the gate-pass fraction gives mean $S$. Layout, text, and local-graphics scores also average over all tasks, with zeros for gated outputs, and sum to mean $S$. We report generation cost in USD for each configuration's 500-task evaluation, excluding judging. Appendix C.1 gives execution and cost-accounting details; Appendix E compares configuration and model-family weighting.

### 3.2 Overall Performance

**The best configuration scores 77.34, while the median scores 24.38.** Table 1 reports all 36 model–harness–effort configurations, ranked by mean score. GPT-6 Astra at high leads with 77.34, followed by Astra xhigh (72.60) and Astra max (68.76). Eight of 36 configurations exceed 40 points, and the weakest, GPT-5.6 Luna at none, scores 1.53. Across all 18,000 model–task pairs, 461 (2.56%) pass the gates without any recorded detail finding.

**Cost varies widely even among similarly performing systems.** Reported generation cost for 500 tasks ranges from $20.67 to $1,598.36, with a median of about $151.50. GPT-6 Astra averages 156,884 tokens per task at high effort and 849,174 at max. Qwen 3.8 Max scores within 1.59 points of Sol Max for approximately one-third of its cost ($307 versus $988). GLM-5.3-Flash reaches 32.42 for $52.53, while GPT-6 Astra High costs about $321 and Astra Max is the most expensive

Table 1: Overall performance of frontier agents on PPTBench across 36 model–harness–effort configurations, ranked by mean score over 500 reconstruction tasks.

| Model | Harness | Effort | Mean score ↑ /100 | Gate pass rate ↑ (%) | Score on passed tasks ↑ /100 | Layout /30 | Text /40 | Graphics /30 | Cost ↓ ($) |
|---|---|---|---|---|---|---|---|---|---|
| GPT-6 Astra | Codex | high | **77.34** | **80.8** | 95.7 | **24.02** | **30.02** | **23.30** | 321 |
| GPT-6 Astra | Codex | xhigh | 72.60 | 74.8 | **97.1** | 22.31 | 28.59 | 21.70 | 967 |
| GPT-6 Astra | Codex | max | 68.76 | 71.0 | 96.8 | 21.15 | 27.04 | 20.57 | 1,598 |
| GPT-6 Astra | Codex | medium | 68.16 | 71.6 | 95.2 | 21.25 | 26.36 | 20.54 | 208 |
| Kimi K3 | OpenCode | high | 67.80 | 73.8 | 91.9 | 21.49 | 26.33 | 19.97 | 1,158 |
| GPT-6 Astra | Codex | low | 59.42 | 63.0 | 94.3 | 18.65 | 22.99 | 17.78 | 179 |
| GPT-5.6 Sol | Codex | max | 49.28 | 52.6 | 93.7 | 15.57 | 18.93 | 14.78 | 988 |
| Qwen 3.8 Max | Claude Code | xhigh | 47.69 | 53.2 | 89.7 | 15.51 | 17.99 | 14.19 | 307 |
| GPT-5.6 Sol | Codex | xhigh | 36.24 | 40.0 | 90.6 | 11.73 | 13.50 | 11.01 | 495 |
| Claude Opus 5 | Claude Code | xhigh | 35.91 | 43.6 | 82.4 | 11.42 | 15.59 | 8.90 | 685 |
| Claude Opus 5 | Claude Code | max | 33.54 | 40.8 | 82.2 | 10.77 | 14.46 | 8.32 | 719 |
| GPT-5.6 Terra | Codex | max | 33.46 | 37.4 | 89.5 | 10.99 | 12.07 | 10.40 | 283 |
| Claude Opus 5 | Claude Code | high | 33.12 | 40.8 | 81.2 | 10.62 | 14.06 | 8.43 | 365 |
| GLM-5.3-Flash | OpenCode | max | 32.42 | 36.8 | 88.1 | 10.45 | 12.61 | 9.36 | 53 |
| GPT-5.6 Luna | Codex | max | 31.02 | 36.4 | 85.2 | 10.70 | 10.43 | 9.89 | 166 |
| GPT-5.6 Sol | Codex | high | 28.28 | 31.8 | 88.9 | 9.28 | 10.27 | 8.73 | 285 |
| GPT-5.6 Terra | Codex | xhigh | 25.84 | 29.6 | 87.3 | 8.67 | 9.25 | 7.92 | 125 |
| GPT-5.6 Luna | Codex | xhigh | 24.63 | 29.0 | 84.9 | 8.52 | 8.27 | 7.83 | 87 |
| Claude Opus 5 | Claude Code | medium | 24.13 | 31.2 | 77.3 | 8.17 | 9.36 | 6.60 | 132 |
| GPT-5.6 Sol | Codex | medium | 22.20 | 25.4 | 87.4 | 7.45 | 8.08 | 6.67 | 179 |
| GPT-5.6 Luna | Codex | high | 19.47 | 23.8 | 81.8 | 6.89 | 6.44 | 6.15 | 57 |
| Claude Opus 5 | Claude Code | low | 17.90 | 23.6 | 75.8 | 6.02 | 6.89 | 4.98 | 73 |
| GPT-5.6 Terra | Codex | high | 16.89 | 19.4 | 87.0 | 5.67 | 6.08 | 5.14 | 81 |
| GPT-5.6 Sol | Codex | low | 11.12 | 13.0 | 85.6 | 3.80 | 3.88 | 3.45 | 123 |
| GPT-5.6 Terra | Codex | medium | 10.10 | 12.0 | 84.1 | 3.43 | 3.55 | 3.12 | 62 |
| GPT-5.5 | Codex | medium | 9.64 | 11.4 | 84.6 | 3.27 | 3.39 | 2.98 | 262 |
| Claude Sonnet 5 | Claude Code | xhigh | 8.98 | 12.8 | 70.1 | 2.75 | 3.97 | 2.26 | 214 |
| Claude Sonnet 5 | Claude Code | high | 8.76 | 12.4 | 70.7 | 2.82 | 3.74 | 2.20 | 137 |
| GPT-5.6 Terra | Codex | low | 8.60 | 9.8 | 87.8 | 2.87 | 3.14 | 2.60 | 58 |
| GPT-5.6 Luna | Codex | medium | 8.15 | 10.0 | 81.5 | 2.93 | 2.65 | 2.57 | 31 |
| Claude Sonnet 5 | Claude Code | medium | 5.99 | 9.2 | 65.1 | 1.80 | 2.65 | 1.54 | 99 |
| GPT-5.6 Sol | Codex | none | 4.99 | 5.8 | 86.0 | 1.65 | 1.88 | 1.46 | 117 |
| Claude Sonnet 5 | Claude Code | low | 4.03 | 6.4 | 63.0 | 1.16 | 1.79 | 1.08 | 61 |
| GPT-5.6 Terra | Codex | none | 3.55 | 4.2 | 84.4 | 1.20 | 1.32 | 1.03 | 45 |
| GPT-5.6 Luna | Codex | low | 3.03 | 3.6 | 84.3 | 1.02 | 1.08 | 0.93 | 21 |
| GPT-5.6 Luna | Codex | none | 1.53 | 1.8 | 85.3 | 0.50 | 0.55 | 0.48 | 22 |

configuration. These are comparisons among the evaluated model–harness configurations under the frozen price snapshot.

### 3.3 Agent Behavior and Failure Modes

**Valid decks are common, but preserving the depicted process remains difficult.** Across all 18,000 pairs, 5,714 (31.74%) pass all gates. Figure 4 shows how many rounds reject each reconstruction; two or more rejections produce a zero score. The per-model breakdown in Appendix C.3 identifies semantics as the dominant failure stage. A deck can preserve its palette and panel layout while reversing arrows or changing branch targets (Appendix D). Gate-pass rates span 1.8–80.8%, whereas mean scores on passed tasks span 63.0–97.1. Mean score tracks gate passage closely ($r = 0.997$), consistent with the multiplicative score and the larger variation in gate success.

**Text fitting accounts for most residual deductions.** Among gate-passing cases, text contributes 53.0% of the 64,210.58 penalty points, local graphics 34.3%, and layout 12.8%. The largest individual deduction is `unexpected_wrap`: 5,724 points across 1,864 cases, with a mean affected fraction of 0.577. Clipping, overflow, and unintended line breaks expose a gap between declared text-frame geometry and rendered output. The OOXML text frame specifies a container, while the renderer determines how its contents wrap. Inspecting native objects alone can therefore miss defects that become visible after rendering. Appendix D reports issue frequencies and representative gate failures.

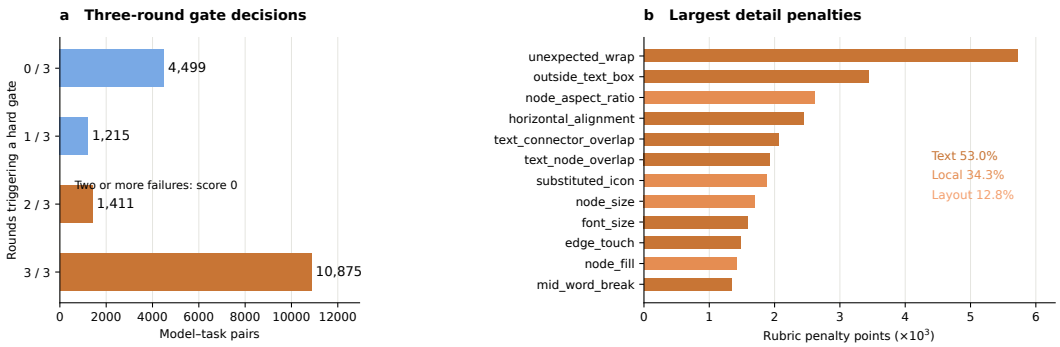



Figure 4: Failure analysis over 18,000 model–task pairs. Left: number of rounds triggering a hard gate. Right: the twelve largest detail penalties among gate-passing cases; dimension shares include all issue types.

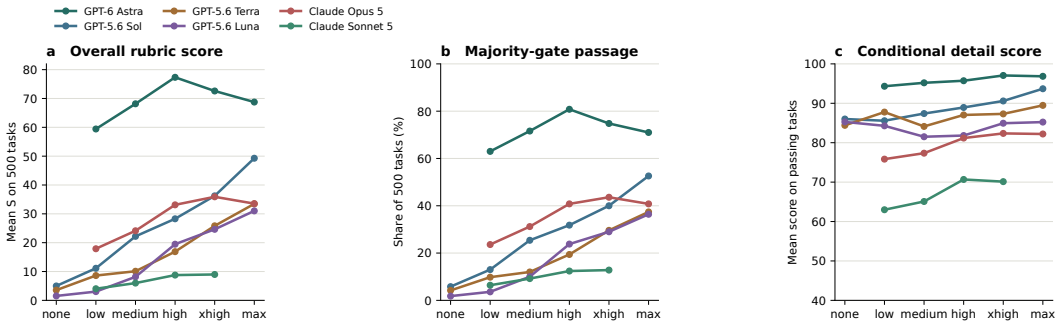



Figure 5: Reasoning-effort sweeps across six model families: mean score over all tasks (left), gate-pass rate (center), and mean score on passed tasks (right).

**More reasoning primarily improves gate passage, with diminishing or non-monotonic gains.** GPT-6 Astra gains 17.92 points from low to high, with a 17.8-point gate-pass increase and only a 1.4-point rise in score on passed tasks. Sol gains 44.29 points from none to max as its gate-pass rate rises by 46.8 points. Terra and Luna show the same broad pattern (Figure 5). Additional effort is not uniformly beneficial: Opus peaks at xhigh (35.91) and falls to 33.54 at max, while Sonnet changes from 8.76 at high to 8.98 at xhigh and loses artifact validity (98.4% to 95.6%). More compute chiefly changes how often systems recover the process; it does not consistently eliminate local visual defects.

Conditional detail means can involve different passing tasks. On the 289 tasks shared by Astra High and Max, Max improves detail by 0.98 points while its overall gate-pass rate falls by 9.8 percentage points. Appendix E reports these matched comparisons.

**Inspecting a reconstruction correlates with quality more strongly than editing it.** For the 27 configurations with parsed trajectories, mean score correlates with review turns at Pearson $r = 0.881$ and Spearman $\rho = 0.897$, versus $r = 0.255$ and $\rho = 0.462$ for revision turns (Appendix F, Figure 9). A review renders or inspects the candidate; a revision edits the deck. This association is compatible with the text-fitting failures above, but does not establish that forced inspection improves quality: reasoning effort affects both inspection and performance, and equally frequent reviews yield different scores across families. For example, at max effort, Sol averages 6.75 reviews per task and scores 49.28, while Luna averages 6.48 and scores 31.02; Terra reviews 3.15 times and scores 33.46. Within-family review–score correlations range from 0.922 to 0.998, but revision counts also correlate positively within these effort sweeps (Table 10). The pooled contrast thus describes differences in agent behavior without isolating the effect of visual inspection.

### 3.4 Judge Agreement and Ranking Stability

**The judge substantially agrees with blinded human gate decisions.** Three annotators independently label 200 cases, with four reconstructions per task across 50 tasks. Their decisions are unanimous

Table 2: Reported human agreement on 200 cases. Gate failure is the positive class.

**Human–human pairwise**

| Pair | Gate agr. | Cohen's $\kappa$ |
|---|---|---|
| Judge 1–Judge 2 | **90.5%** | **0.781** |
| Judge 1–Judge 3 | 89.0% | 0.766 |
| Judge 2–Judge 3 | 83.5% | 0.645 |

**Luna vs. human majority** ($N = 200$)

| | Human majority | |
|---|---|---|
| Luna majority | Gate | Pass |
| Gate | 120 | 15 |
| Pass | 12 | 53 |

**Headline metrics**

| Metric | Value |
|---|---|
| Human three-way unanimity | 81.5% |
| Fleiss' $\kappa$ (three human raters) | 0.728 |
| Gate agreement | 86.5% (173/200) |
| Case-level Wilson 95% CI | 81.1%–90.6% |
| Cohen's $\kappa$ | 0.696 |
| Precision (gate) | 88.9% |
| Recall (gate) | 90.9% |
| $F_1$ (gate) | 89.9% |
| Luna extra gates (false positives) | 15 |
| Luna missed gates (false negatives) | 12 |

Table 3: Repeatability of single-round and three-round gate decisions on 200 cases.

**Single-round consistency** (3 repeats)

| Judge | Unanimous gate rate | Fleiss' $\kappa$ | Flipped cases |
|---|---|---|---|
| Luna High | **87.5%** | **0.809** | **25** |
| Sol Medium | 86.5% | 0.797 | 27 |
| Terra High | 81.0% | 0.714 | 38 |

**Three-round majority, repeated**
Luna High, rounds 1–3 vs. 4–6

| | |
|---|---|
| Agreement | 96.0% |
| Cohen's $\kappa$ | 0.908 |
| Gate $F_1$ | 0.971 |
| Disagreements | 8 / 200 |

in 81.5% of cases. The three-round Luna High majority agrees with the human majority on 86.5% (Wilson 95% CI 81.1–90.6%), with 15 extra gates and 12 missed gates (Table 2). Human unanimity and agreement between majority votes measure different quantities, so the former is not an exact ceiling for the latter. Appendix G.1 gives the sampling, blinding, and judge-selection procedure.

**Independent three-round majorities agree on 96.0% of gate decisions.** Luna High is unanimous across three individual rounds in 87.5% of cases. Two independently collected three-round majorities agree on 96.0%, with 8 disagreements among 200 cases (Table 3). Appendix G.2 separately measures Stage 3 variation within Astra's recorded three-round evaluations.

**Detail findings retain evidence from individual judge rounds.** Single-round support is 6.8% for `unexpected_wrap` and 73.3% for `node_spacing` (Appendix D.1). Human issue-set agreement is reported separately from gate agreement in Appendix G.1. Across four detail-scoring variants, rank correlations with the original ordering remain $\tau = 0.971$–$0.997$ (Appendix G.4). Appendix G.3 compares one-pass, majority, and unanimous voting on Astra.

**Five hundred tasks yield stable broad performance tiers.** Across 10,000 paired task-bootstrap resamples, larger task sets stabilize the ordering (Table 4). At 500 tasks, mean Kendall's $\tau$ is 0.956 and Astra High ranks first in 97.4% of resamples. Appendix G.5 gives the resampling protocol and paired confidence intervals for neighboring configurations.

Table 4: Ranking stability from 10,000 paired task-bootstrap resamples per benchmark size.

| | Kendall's $\tau$ | | Exact top-$k$ set reproduced (%) | | | | 95% score CI width | |
|---|---|---|---|---|---|---|---|---|
| Tasks | mean | 5–95% | $k = 1$ | $k = 2$ | $k = 3$ | $k = 5$ | mean | maximum |
| 50 | 0.871 | 0.829–0.911 | 60.0 | 39.1 | 26.6 | 66.4 | 18.7292 | 26.2842 |
| 100 | 0.907 | 0.873–0.937 | 74.8 | 53.9 | 33.3 | 83.6 | 13.2508 | 18.2886 |
| 200 | 0.932 | 0.905–0.956 | 87.7 | 69.0 | 37.3 | 95.4 | 9.3986 | 12.8205 |
| 300 | 0.944 | 0.921–0.965 | 93.1 | 77.4 | 41.7 | 98.7 | 7.6752 | 10.6089 |
| 400 | 0.951 | 0.927–0.971 | 96.1 | 83.4 | 44.6 | 99.4 | 6.6420 | 9.1836 |
| 500 | **0.956** | 0.933–0.975 | **97.4** | **88.0** | **46.2** | **99.8** | **5.9488** | **8.2862** |

## 4 Related Work

**Presentation, poster and office documents.** Slide generation includes document-to-deck systems (Fu et al., 2022; Sun et al., 2021), LLM agents (Zheng et al., 2025; Yang et al., 2026; Chen et al., 2026), and scientific posters (Pang et al., 2025). These tasks evaluate content, layout, and style against open-ended targets. Office benchmarks specify edits to existing files (Guo et al., 2023; Wang et al., 2024b). The earlier PPTBench (Huang et al., 2025) covers detection, understanding, modification, and generation, using task-specific correctness metrics and a holistic generation score. Slide2Code (Tang et al., 2025b) directly evaluates reference-image reconstruction as editable PPTX and is the closest task setting. Chart-to-code (Yang et al., 2025; Wu et al., 2024; Tang et al., 2025a; Zhang et al., 2026a; Li et al., 2026) and screenshot-to-markup (Si et al., 2024) also fix the reference, with plotting code or markup as output. Our PPTBench concentrates on scientific process diagrams, checks native-object validity, and gates process and connector fidelity before applying localized visual deductions. These stages distinguish failure to preserve the depicted process from errors in appearance. Table 5 in Appendix A compares these settings.

**Understanding, parsing and generating diagrams.** Figure question answering tests chart and document understanding (Masry et al., 2022; Mathew et al., 2022), while CharXiv (Wang et al., 2024c) evaluates charts from real scientific papers. Structured prediction recovers diagram constituents and relations (Kembhavi et al., 2016; Sun et al., 2022), and flowchart benchmarks test reasoning over process structure (Pan et al., 2024; Singh et al., 2024). Diagram generation from text or papers (Zala et al., 2024; Huang et al., 2026) also motivates structural evaluation through graph comparison (Liang & You, 2025) and inverse parsing (Zhang et al., 2026b). PPTBench supplies one reference image and evaluates both its appearance and depicted process in the reconstructed slide. The process and connector gate checks visible nodes, connections, routed segments, and arrowheads without requiring the OOXML tree to match a reference graph.

Image-to-TikZ and SVG reconstruction (Belouadi et al., 2024b; Rodriguez et al., 2025) and Image2Struct (Roberts et al., 2024b) also recover structured representations from visual targets. PPTBench studies this reconstruction problem under native PowerPoint constraints, on scientific diagrams whose visible process can fail even when the rendered layout is close. Its contribution is the task collection and staged evaluation of that setting. The separation of artifact, gate, and detail results makes this difference experimentally inspectable.

**Model-based judges and agentic execution.** LLM and multimodal judges approximate human evaluation (Zheng et al., 2023; Chen et al., 2024a; Lee et al., 2024). Agent-as-a-Judge adds tool use (Zhuge et al., 2024). PPTBench's judge uses image-analysis scripts and returns gates and localized findings for deterministic scoring, with reported gate agreement against blinded human labels and independent repetitions (Section 3.4). Evaluated agents combine reasoning and tool use (Yao et al., 2023), code execution (Wang et al., 2024a), and revision from feedback (Shinn et al., 2023; Madaan et al., 2023). Appendix A provides extended comparisons.

## 5 Conclusion

We introduced PPTBench to evaluate how coding agents translate visual targets into structured, editable slides. Its 500 scientific flow-diagram tasks and four-stage judge assess artifact validity, process semantics, rendering quality, and visual detail. Across ten models and 36 configurations, agents generally produce valid slides but struggle to preserve the depicted process and visual structure, with the best mean score reaching 77.34. These findings highlight the challenge of recovering objects and their relationships from visual input and expressing them faithfully through code. Future work can investigate explicit structural reasoning and verification through rendered feedback, and extend evaluation to richer visual artifacts such as interfaces and 3D scenes. More broadly, PPTBench supports the development of agents that turn visual intent into meaningful artifacts that people can inspect, edit, and reuse.

## REFERENCES


Jonas Belouadi, Anne Lauscher, and Steffen Eger. AutomaTikZ: Text-guided synthesis of scientific vector graphics with TikZ. In *International Conference on Learning Representations (ICLR)*, 2024a. arXiv:2310.00367.

Jonas Belouadi, Simone Paolo Ponzetto, and Steffen Eger. DeTikZify: Synthesizing graphics programs for scientific figures and sketches with TikZ. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. arXiv:2405.15306.

Tony Beltramelli. pix2code: Generating code from a graphical user interface screenshot. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS)*, 2018. doi: 10.1145/3220134.3220135.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yinuo Liu, Yaochen Wang, Huichi Zhou, Qihui Zhang, Pan Zhou, Yao Wan, and Lichao Sun. MLLM-as-a-judge: Assessing multimodal LLM-as-a-judge with vision-language benchmark. In *International Conference on Machine Learning (ICML)*, 2024a. arXiv:2402.04788.

Xin-Sheng Chen, Jiayu Zhu, Pei lin Li, Hanzheng Wang, Shuojin Yang, and Meng-Hao Guo. Presentbench: A fine-grained rubric-based benchmark for slide generation. *arXiv preprint arXiv:2603.07244*, 2026.

Zhaorun Chen, Yichao Du, Zichen Wen, Yiyang Zhou, Chenhang Cui, Zhenzhen Weng, Haoqin Tu, Chaoqi Wang, Zhengwei Tong, Qinglan Huang, Canyu Chen, Qinghao Ye, et al. MJ-Bench: Is your multimodal reward model really a good judge for text-to-image generation? *arXiv preprint arXiv:2407.04842*, 2024b.

Wayne Chi, Yixiong Fang, Arnav Yayavaram, Siddharth Yayavaram, Seth Karten, Qiuhong Anna Wei, Runkun Chen, Alexander Wang, Valerie Chen, Ameet Talwalkar, and Chris Donahue. GameDevBench: Evaluating agentic capabilities through game development. *arXiv preprint arXiv:2602.11103*, 2026a. doi: 10.48550/arXiv.2602.11103. URL https://arxiv.org/abs/2602.11103.

Yizhe Chi, Deyao Hong, Dapeng Jiang, Tianwei Luo, Kaisen Yang, Boshi Zhang, Zhe Cao, Xiaoyan Fan, Bingxiang He, Han Hao, et al. Frontier-eng: Benchmarking self-evolving agents on real-world engineering tasks with generative optimization. *arXiv preprint arXiv:2604.12290*, 2026b.

Yizhe Chi, Wenyi Li, Deyao Hong, Xiaoqiu Wang, Mingju Gao, Kaisen Yang, Bingxiang He, Youjie Zheng, Calvin Xiao, and Qinhuai Na. Ai4ai-bench: Benchmarking llm agents in algorithmic design for recursive self-improvement. *arXiv preprint arXiv:2608.20318*, 2026c.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104.

Zhiyao Cui, Chenxu Wang, Shuyue Hu, Yiqun Zhang, Wenqi Shao, Qiaosheng Zhang, and Zhen Wang. Design first, code later: Aesthetically pleasing template-free slides generation. *arXiv preprint arXiv:2605.26451*, 2026.

Tsu-Jui Fu, William Yang Wang, Daniel McDuff, and Yale Song. DOC2PPT: Automatic presentation slides generation from scientific documents. In *AAAI Conference on Artificial Intelligence*, volume 36, pp. 634–642, 2022. doi: 10.1609/aaai.v36i1.19943.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and Trevor Darrell. AutoPresent: Designing structured visuals from scratch. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2501.00912.

Yiduo Guo, Zekai Zhang, Yaobo Liang, Dongyan Zhao, and Nan Duan. Pptc benchmark: Evaluating large language models for powerpoint task completion. *arXiv preprint arXiv:2311.01767*, 2023.

Ziyi Guo, Zhou Liu, and Wentao Zhang. Paper2SysArch: Structure-constrained system architecture generation from scientific papers. *arXiv preprint arXiv:2511.18036*, 2025.

Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S. Davis, Vijay Mahadevan, and Abhinav Shrivastava. LayoutTransformer: Layout generation and completion with self-attention. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1004–1014, 2021.

Deyao Hong, Yizhe Chi, Wenyi Li, Xiaoqiu Wang, Mingju Gao, Kaisen Yang, Bingxiang He, Youjie Zheng, Calvin Xiao, and Qinhuai Na. Swe refactor bench: Can coding agents complete a long-horizon, whole-repository stack migration? *arXiv preprint arXiv:2608.23564*, 2026.

Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A. Ross, Cordelia Schmid, and Alireza Fathi. SceneCraft: An LLM agent for synthesizing 3D scenes as Blender code. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19252–19282. PMLR, 2024. URL https://proceedings.mlr.press/v235/hu24g.html.

Siyuan Huang, Yifan Zhou, Yutong Gao, Zi Yin, Juyang Bai, Xinxin Liu, Rama Chellappa, Chun Pong Lau, Cheng Peng, Sayan Nag, and Shraman Pramanick. SciFig: Towards automating editable figure generation for scientific papers. *arXiv preprint arXiv:2601.04390*, 2026.

Zheng Huang, Xukai Liu, Tianyu Hu, Kai Zhang, and Ye Liu. PPTBench: Towards holistic evaluation of large language models for PowerPoint layout and design understanding. *arXiv preprint arXiv:2512.02624*, 2025. doi: 10.48550/arXiv.2512.02624. URL https://arxiv.org/abs/2512.02624.

Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. LayoutDM: Discrete diffusion model for controllable layout generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10167–10176, 2023.

Peidong Jia, Chenxuan Li, Yuhui Yuan, Zeyu Liu, Yichao Shen, Bohan Chen, Xingru Chen, Yinglin Zheng, Dong Chen, Ji Li, Xiaodong Xie, Shanghang Zhang, and Baining Guo. COLE: A hierarchical generation framework for multi-layered and editable graphic design. *arXiv preprint arXiv:2311.16974*, 2024.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. FigureQA: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*, 2018.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision (ECCV)*, pp. 235–251, 2016. doi: 10.1007/978-3-319-46493-0_15.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5376–5384, 2017.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.08491.

Hugo Laurençon, Léo Tronchon, and Victor Sanh. Unlocking the conversion of web screenshots into HTML code with the WebSight dataset. *arXiv preprint arXiv:2403.09029*, 2024.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning (ICML)*, pp. 18893–18912, 2023.

Seongyun Lee, Seungone Kim, Sue Park, Geewook Kim, and Minjoon Seo. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11286–11315, 2024. doi: 10.18653/v1/2024.findings-acl.672.

Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and Zhao-Xiang Zhang. SheetCopilot: Bringing software productivity to the next level through large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, Lingpeng Kong, and Qi Liu. VL-RewardBench: A challenging benchmark for vision-language generative reward models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a. arXiv:2411.17451.

Ryan Li, Yanzhe Zhang, and Diyi Yang. Sketch2Code: Evaluating vision-language models for interactive web design prototyping. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, pp. 3921–3955, 2025b.

Wenyi Li, Renkai Luo, Yue Yu, Huan-ang Gao, Mingju Gao, Li Yuan, Chaoyou Fu, and Hao Zhao. Benchmarking phd-level coding in 3d geometric computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30974–30985, June 2026.

Chumeng Liang and Jiaxuan You. DiagramEval: Evaluating LLM-generated diagrams via graphs. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. arXiv:2510.25761.

Jiawei Lin, Jiaqi Guo, Shizhao Sun, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. LayoutPrompter: Awaken the design ability of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2511–2522, 2023.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 6774–6786, 2021.

Zeyao Ma, Bohan Zhang, Jing Zhang, Jifan Yu, Xiaokang Zhang, Xiaohan Zhang, Sijia Luo, Xi Wang, and Jie Tang. SpreadsheetBench: Towards challenging real world spreadsheet manipulation. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, 2022.

Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. UniChart: A universal vision-language pretrained model for chart comprehension and reasoning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 14662–14684, 2023.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1697–1706, 2022.

Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. PlotQA: Reasoning over scientific plots. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1527–1536, 2020.

Huitong Pan, Qi Zhang, Cornelia Caragea, Eduard Dragut, and Longin Jan Latecki. FlowLearn: Evaluating large vision-language models on flowchart understanding. In *European Conference on Artificial Intelligence (ECAI)*, pp. 73–80, 2024. doi: 10.3233/FAIA240473.

Yiming Pan, Chengwei Hu, Xuancheng Huang, Can Huang, Mingming Zhao, Yuean Bi, Xiaohan Zhang, Aohan Zeng, and Linmei Hu. AeSlides: Incentivizing aesthetic layout in LLM-based slide generation via verifiable rewards. *arXiv preprint arXiv:2604.22840*, 2026.

Wei Pang, Kevin Qinghong Lin, Xiangru Jian, Xi He, and Philip Torr. Paper2Poster: Towards multimodal poster automation from scientific papers. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2025. arXiv:2505.21497.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Jonathan Roberts, Kai Han, Neil Houlsby, and Samuel Albanie. SciFIBench: Benchmarking large multimodal models for scientific figure interpretation. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, 2024a.

Josselin Somerville Roberts, Tony Lee, Chi Heem Wong, Michihiro Yasunaga, Yifan Mai, and Percy Liang. Image2Struct: Benchmarking structure extraction for vision-language models. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, pp. 115058–115097, 2024b.

Juan A. Rodriguez, Abhay Puri, Shubham Agarwal, Issam H. Laradji, Pau Rodriguez, Sai Rajeswar, David Vazquez, Christopher Pal, and Marco Pedersoli. StarVector: Generating scalable vector graphics code from images and text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16175–16186, 2025.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Chenglei Si, Yanzhe Zhang, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. Design2Code: How far are we from automating front-end engineering? *arXiv preprint arXiv:2403.03163*, 2024.

Shubhankar Singh, Purvi Chaurasia, Yerram Varun, Pranshu Pandya, Vatsal Gupta, Vivek Gupta, and Dan Roth. FlowVQA: Mapping multimodal logic in visual question answering with flowcharts. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1330–1350, 2024. doi: 10.18653/v1/2024.findings-acl.78.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Edward Sun, Yufang Hou, Dakuo Wang, Yunfeng Zhang, and Nancy X. R. Wang. D2S: Document-to-slide generation via query-based text summarization. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 1405–1418, 2021.

Lianshan Sun, Hanchao Du, and Tao Hou. FR-DETR: End-to-end flowchart recognition with precision and robustness. *IEEE Access*, 10:64292–64301, 2022. doi: 10.1109/ACCESS.2022.3183068.

Jiahao Tang, Henry Hengyuan Zhao, Lijian Wu, Zijian Zhang, Yifei Tao, Dongxing Mao, Yang Wan, Jingru Tan, Min Zeng, Min Li, and Alex Jinpeng Wang. From charts to code: A hierarchical benchmark for multimodal models. *arXiv preprint arXiv:2510.17932*, 2025a.

Wenxin Tang, Jingyu Xiao, Wenxuan Jiang, Xi Xiao, Yuhang Wang, Xuxin Tang, Qing Li, Yuehe Ma, Junliang Liu, Shisong Tang, and Michael R. Lyu. SlideCoder: Layout-aware RAG-enhanced hierarchical slide generation from design. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9015–9039, 2025b. arXiv:2506.07964.

Simon Tannert, Marcelo G. Feighelstein, Jasmina Bogojeska, Joseph Shtok, Assaf Arbelle, Peter W. J. Staar, Anika Schumann, Jonas Kuhn, and Leonid Karlinsky. FlowchartQA: The first large-scale benchmark for reasoning over flowcharts. In *Proceedings of the 1st Workshop on Linguistic Insights from and for Multimodal Language Processing (LIMO)*, pp. 34–46, 2023.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents. In *International Conference on Machine Learning (ICML)*, 2024a. arXiv:2402.01030.

Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. OfficeBench: Benchmarking language agents across multiple applications for office automation. *arXiv preprint arXiv:2407.19056*, 2024b.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, Alexis Chevalier, Sanjeev Arora, and Danqi Chen. CharXiv: Charting gaps in realistic chart understanding in multimodal LLMs. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, 2024c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Chengyue Wu, Yixiao Ge, Qiushan Guo, Jiahao Wang, Zhixuan Liang, Zeyu Lu, Ying Shan, and Ping Luo. Plot2code: A comprehensive benchmark for evaluating multi-modal large language models in code generation from scientific plots. *arXiv preprint arXiv:2405.07990*, 2024.

Ronghuan Wu, Wanchao Su, and Jing Liao. Chat2SVG: Vector graphics generation with large language models and image diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23690–23700, 2025.

Jingyu Xiao, Yuxuan Wan, Yintong Huo, Zixin Wang, Xinyi Xu, Wenxuan Wang, Zhiyao Xu, Yuhang Wang, and Michael R. Lyu. Interaction2Code: Benchmarking MLLM-based interactive webpage code generation from interactive prototyping. In *IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 241–253, 2025. arXiv:2411.03292.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, 2024.

Ximing Xing, Juncheng Hu, Guotao Liang, Jing Zhang, Dong Xu, and Qian Yu. Empowering LLMs to understand and generate complex vector graphics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19487–19497, 2025.

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, Gongye Liu, Xiaomei Nie, Deng Cai, and Yujiu Yang. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2406.09961.

Tao Yang, Yingmin Luo, Zhongang Qi, Yang Wu, Ying Shan, and Chang Wen Chen. PosterLLaVa: Constructing a unified multi-modal layout generator with LLM. *arXiv preprint arXiv:2406.02884*, 2024a.

Yunqiao Yang, Wenbo Li, Houxing Ren, Zimu Lu, Ke Wang, Zhiyuan Huang, Zhuofan Zong, Mingjie Zhan, and Hongsheng Li. Slidesgen-bench: Evaluating slides generation via computational and quantitative metrics. *arXiv preprint arXiv:2601.09487*, 2026.

Zhiyu Yang, Zihan Zhou, Shuo Wang, Xin Cong, Xu Han, Yukun Yan, Zhenghao Liu, Zhixing Tan, Pengyuan Liu, Dong Yu, Zhiyuan Liu, Xiaodong Shi, and Maosong Sun. MatPlotAgent: Method and evaluation for LLM-based agentic scientific data visualization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11789–11804, 2024b. doi: 10.18653/v1/2024.findings-acl.701.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Sukmin Yun, Haokun Lin, Rusiru Thushara, Mohammad Qazim Bhat, Yongxin Wang, Zutao Jiang, Mingkai Deng, Jinhong Wang, Tianhua Tao, Junbo Li, Haonan Li, Preslav Nakov, Timothy Baldwin, Zhengzhong Liu, Eric P. Xing, Xiaodan Liang, and Zhiqiang Shen. Web2Code: A large-scale webpage-to-code dataset and evaluation framework for multimodal LLMs. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, 2024.

Abhay Zala, Han Lin, Jaemin Cho, and Mohit Bansal. DiagrammerGPT: Generating open-domain, open-platform diagrams via LLM planning. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2310.12128.

Jiajun Zhang, Yuying Li, Zhixun Li, Xingyu Guo, Jingzhuo Wu, Leqi Zheng, Yiran Yang, Jianke Zhang, Qingbin Li, Shannan Yan, Zhetong Li, Changguo Jia, Junfei Wu, Zilei Wang, Qiang Liu, and Liang Wang. Realchart2code: Advancing chart-to-code generation with real data and multi-task evaluation. *arXiv preprint arXiv:2603.25804*, 2026a.

Ming-Liang Zhang, Fei Yin, Yi-Han Hao, and Cheng-Lin Liu. Plane geometry diagram parsing. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1636–1643, 2022. doi: 10.24963/ijcai.2022/228.

Tong Zhang, Honglin Lin, Zhou Liu, Chong Chen, and Wentao Zhang. SciFlow-Bench: Evaluating structure-aware scientific diagram generation via inverse parsing. *arXiv preprint arXiv:2602.09809*, 2026b.

Zekai Zhang, Yiduo Guo, Yaobo Liang, Dongyan Zhao, and Nan Duan. PPTC-R benchmark: Towards evaluating the robustness of large language models for PowerPoint task completion. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. arXiv:2403.03788.

Haozhe Zhao, Shuzheng Si, Zhenhailong Wang, Zheng Wang, Liang Chen, Xiaotong Li, Zhixiang Liang, Maosong Sun, and Minjia Zhang. Crafter: A multi-agent harness for editable scientific figure generation from diverse inputs. *arXiv preprint arXiv:2605.30611*, 2026.

Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Zhiyuan Liu, and Maosong Sun. Chartcoder: Advancing multimodal large language model for chart-to-code generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. arXiv:2501.06598.

Hao Zheng, Xinyan Guan, Hao Kong, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Ben He, Xianpei Han, and Le Sun. Pptagent: Generating and evaluating presentations beyond text-to-slides. *arXiv preprint arXiv:2501.03936*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*, 2023.

Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-judge: Evaluate agents with agents. *arXiv preprint arXiv:2410.10934*, 2024.

Bocheng Zou, Mu Cai, Jianrui Zhang, and Yong Jae Lee. VGBench: Evaluating large language models on vector graphics understanding and generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3647–3659, 2024.

### AI USE STATEMENT

We used generative AI tools to assist with language polishing and code generation. Model outputs also form the reconstruction candidates and structured judge findings in the evaluation protocol described in Section 2; deterministic rules convert those findings into scores. The authors reviewed the manuscript and remain responsible for its content and results.

### REPRODUCIBILITY STATEMENT

Task construction, the artifact contract, the four-stage judge and the multiplicative score are specified in Section 2. The evaluation protocol, the 36 configurations and the validity study against human labels are in Section 3. Appendix B lists domain counts, arXiv categories, and representative source papers. Execution and image preparation are described in Appendix C; validation, sensitivity analyses, and task resampling are detailed in Appendix G. Scored examples and the complete detail rubric appear in Appendices H and I.

## A RELATED BENCHMARK COMPARISONS

Table 5 locates PPTBench among visual reconstruction, diagram parsing, and presentation benchmarks. Native PPTX is an output constraint; topology gating is a scoring choice.

Table 5: Related evaluation settings, output representations, and evaluation targets.

| Setting | Representation | Evaluation target |
|---|---|---|
| Chart reconstruction (Yang et al., 2025; Wu et al., 2024; Tang et al., 2025a; Zhang et al., 2026a) | Plotting code | Reproduce chart appearance and numerical content. |
| Visual code reconstruction (Belouadi et al., 2024b; Rodriguez et al., 2025; Roberts et al., 2024b) | TikZ, SVG, or other source | Recover a rendered target in an executable representation. |
| Diagram parsing and evaluation (Kembhavi et al., 2016; Sun et al., 2022; Liang & You, 2025) | Diagram constituents and relations | Recover or compare semantic structure. |
| Slide generation and editing (Zheng et al., 2025; Yang et al., 2026; Chen et al., 2026; Guo et al., 2023) | Slide decks or PPTX edits | Generate an open-ended design or execute specified edits. |
| Reference-image slide reconstruction (Tang et al., 2025b) | Editable PPTX | Reconstruct a slide design; assess content, layout, execution, and visual fidelity. |
| PPTBench (Huang et al., 2025) | Answers or PowerPoint API sequences | Evaluate detection, understanding, modification, and generation with task-specific metrics. |
| PPTBench (ours) | Native PPTX slide | Reconstruct a fixed scientific diagram; check native-object validity, visible process fidelity, and visual defects. |

**Visual reconstruction.** Chart-to-code systems (Yang et al., 2025; Wu et al., 2024; Zhao et al., 2025; Tang et al., 2025a; Zhang et al., 2026a; Yang et al., 2024b), TikZ and SVG generation (Belouadi et al., 2024a;b; Rodriguez et al., 2025; Wu et al., 2025; Xing et al., 2025; Zou et al., 2024), and screenshot-to-markup methods (Beltramelli, 2018; Si et al., 2024; Laurençon et al., 2024; Yun et al., 2024; Li et al., 2025b; Xiao et al., 2025) recover executable representations. Image2Struct (Roberts et al., 2024b) evaluates the rendered reconstruction. PPTBench additionally requires native slide objects and evaluates process semantics.

**Presentation and document authoring.** Slide and poster generation (Fu et al., 2022; Sun et al., 2021; Zheng et al., 2025; Yang et al., 2026; Chen et al., 2026; Ge et al., 2025; Cui et al., 2026; Pan et al., 2026; Pang et al., 2025) and layout synthesis (Gupta et al., 2021; Inoue et al., 2023; Lin et al.,

2023; Jia et al., 2024; Yang et al., 2024a) evaluate open-ended designs. Office benchmarks specify operations on existing files (Guo et al., 2023; Zhang et al., 2024; Li et al., 2023; Ma et al., 2024; Wang et al., 2024b). Slide2Code (Tang et al., 2025b) fixes a reference image and evaluates editable-slide reconstruction. Its 900-example benchmark has three complexity tiers; the main experiment uses 300 examples. Evaluation compares content and position with the original PPTX, alongside execution, CLIP, and SSIM. PPTBench shares the image-to-editable-slide setting but collects 500 scientific flow diagrams without target PPTX files. Its staged records separate native-object validity, reference-specific process and connector requirements, and visual penalties.

**Distinction from the earlier PPTBench.** The earlier PPTBench (Huang et al., 2025) contains 4,439 instances in 11 subtasks drawn from 958 government presentations. Tasks on existing slides combine screenshots with structured JSON; modification and generation use 17 predefined PowerPoint APIs. Evaluation uses exact matching for detection and post-edit JSON, answer accuracy for understanding, and a six-point LLM rubric for generation. Its generation task includes screenshot inputs, overlapping with visual reconstruction. Our PPTBench fixes 500 scientific diagram images without target PPTX files or supplied object decompositions. Agents infer editable objects and connections from the reference using their choice of authoring library. The evaluation checks native-object validity and applies hard gates to process and connector fidelity and rendering quality, then converts localized findings into deterministic visual deductions. This design isolates failures in scientific process reconstruction from local visual defects.

**Diagram understanding and generation.** Figure and document QA (Kahou et al., 2018; Methani et al., 2020; Masry et al., 2022; Mathew et al., 2021; 2022; Lee et al., 2023; Masry et al., 2023; Wang et al., 2024c; Roberts et al., 2024a), structured parsing (Kembhavi et al., 2016; 2017; Sun et al., 2022; Zhang et al., 2022; Lu et al., 2021), and flowchart reasoning (Tannert et al., 2023; Pan et al., 2024; Singh et al., 2024) measure complementary forms of visual understanding. Diagram generation (Zala et al., 2024; Huang et al., 2026; Zhao et al., 2026; Guo et al., 2025) and structural evaluation (Liang & You, 2025; Zhang et al., 2026b) address semantic fidelity. PPTBench evaluates the process depicted by editable objects without requiring a unique reference OOXML tree.

**Judges and coding agents.** Model-based evaluation studies rubric adherence and judge biases (Zheng et al., 2023; Chi et al., 2026b; Liu et al., 2023; Kim et al., 2024; Chi et al., 2026c; Zhu et al., 2023; Panickssery et al., 2024; Hong et al., 2026; Chen et al., 2024a; Lee et al., 2024; Chen et al., 2024b; Li et al., 2025a). Agent-as-a-Judge introduces evaluator tool use (Zhuge et al., 2024). PPTBench combines this capability with deterministic scoring and human agreement checks (Cohen, 1960). Its tested agents use tool actions, code execution, and feedback-driven revision (Yao et al., 2023; Wang et al., 2024a; Shinn et al., 2023; Madaan et al., 2023); execution-based benchmarks (Jimenez et al., 2024; Xie et al., 2024) and reasoning scaling (Wei et al., 2022; Snell et al., 2024) provide related evaluation settings.

## B TASK CATALOG

Table 6 lists domain counts and representative source papers. The ten presentation domains aggregate 50 arXiv primary categories; source identifiers are retained for traceability and withheld from agents.

Table 6: Task counts, arXiv categories, and representative source papers for the ten benchmark domains.

| Domain | Tasks | arXiv classes | Representative source paper |
|---|---|---|---|
| Systems, architecture and SE | 91 | cs.SE (32); cs.DC (31); cs.AR (25); cs.CE (1); cs.ET (1); cs.OS (1) | TDFlow: Agentic Workflows for Test Driven Development (arXiv: 2510.23761) |
| AI and machine learning | 90 | cs.LG (54); cs.AI (19); cs.NE (15); stat.ML (2) | A Distributionally Robust Framework for Nuisance in Causal Effect Estimation (arXiv: 2505.17717) |
| Computer vision and imaging | 67 | cs.CV (51); eess.IV (15); cs.MM (1) | Building Optimal Neural Architectures using Interpretable Knowledge (arXiv: 2403.13293) |
| NLP, speech and audio | 54 | cs.CL (29); eess.AS (17); cs.SD (8) | Speed Always Wins: A Survey on Efficient Architectures for Large Language Models (arXiv: 2508.09834) |
| Quantum and fundamental physics | 43 | quant-ph (34); gr-qc (7); hep-ex (1); nucl-th (1) | Pinball: A Cryogenic Predecoder for Surface Code Decoding Under Circuit-Level Noise (arXiv: 2512.09807) |
| Astronomy and astrophysics | 37 | astro-ph.IM (30); astro-ph.EP (3); astro-ph.GA (2); astro-ph.HE (2) | Validation of the HERA Phase I Epoch of Reionization 21 cm Power Spectrum Software Pipeline (arXiv: 2104.09547) |
| Robotics and HCI | 29 | cs.RO (18); cs.HC (11) | Implementing a Robot Intrusion Prevention System (RIPS) for ROS 2 (arXiv: 2412.19272) |
| Networking, security and IR | 25 | cs.NI (10); cs.CR (5); cs.IT (5); cs.IR (4); cs.SI (1) | An OPC UA-based industrial Big Data architecture (arXiv: 2306.01418) |
| Biomedicine and life sciences | 19 | q-bio.QM (6); q-bio.NC (5); q-bio.BM (4); physics.med-ph (2); physics.bio-ph (1); q-bio.GN (1) | MegaFold: Efficient Training of Next-Generation 3D Attention Protein Models on Cross-Platform GPUs (arXiv: 2506.20686) |
| Others | 45 | physics.ins-det (2); math.DS (1); physics.comp-ph (1); physics.flu-dyn (1); stat.AP (1); cs.CY (5); econ.GN (3); physics.soc-ph (2); cond-mat.mtrl-sci (10); cond-mat.dis-nn (1); cond-mat.str-el (1); eess.SP (16); math.OC (1) | A Flexible Data Acquisition System Architecture for the Nab Experiment (arXiv: 2407.17606) |

# C Complete results and execution details

## C.1 Execution and cost accounting

The generation prompt asks the agent to inspect the reference image and reconstruct it as a single editable PowerPoint slide. It lists the approved raster resources for the task, requires each resource to appear unmodified in its intended location, and requires all other diagram content to use native PowerPoint objects. Agents choose their authoring method; python-pptx and Pillow are provided. Before submission, the agent must open or parse the deck to verify its validity.

Astra uses Codex CLI 0.154.0, with generation concurrency 10 and judge concurrency 4 per effort. Model-call, first-event, and stream-idle timeouts are disabled. Invalid or missing model outputs score zero. Rendering operates on the final generated deck. These runtime settings describe Astra; the other model–harness combinations are listed in Table 1.

We report the final generation-cost total for each configuration over the 500-task evaluation, excluding judging. Costs use recorded provider charges or token usage repriced at a frozen price snapshot, with reasoning tokens billed as output and cached input accounted for separately. For Astra, the estimate uses the generation associated with each final evaluated output, including artifact failures. Rates per million tokens are \$10 for uncached input, \$1 for cached input, \$12.50 for cache writes, and \$50 for output; long-context surcharges, service-tier adjustments, and tool fees are excluded. The other configurations use their recorded generation totals. Astra averages 156,884 tokens per task at high effort and 849,174 at max.

## C.2 Image preparation and evaluation

Evaluation applies Stage 0 artifact validation, renders the candidate with LibreOffice, runs the three judge stages, and aggregates three rounds with deterministic scoring rules.

Normalization uses the median border color as background, a content threshold of 10 intensity levels, and a 2% crop margin with a four-pixel minimum. Aspect-preserving crops are centered in $1200 \times 900$ white panels separated by four pixels. Full-resolution images accompany the pair. All 2,500 Astra runs exclude captions. Agents receive reference and resource images without source-paper provenance.

## C.3 Outcome proportions

Figure 6 averages outcome proportions equally across each model's tested effort settings. Across all 36 configurations, 5,714 cases pass all gates, including 461 with no recorded detail finding. These model-level averages complement Table 1.

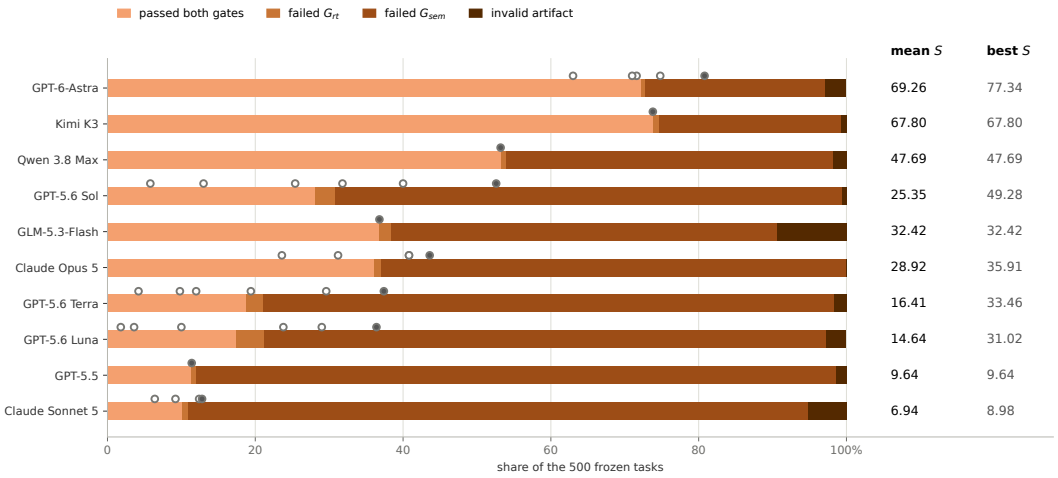
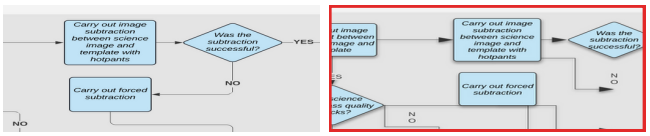



Figure 6: Outcome proportions by model, averaged across tested effort settings. Dots mark the gate-pass rates of individual configurations.

## D FAILURE CASES AND DETAIL STATISTICS

Figures 7 and 8 show process/connector and rendering failures. Task 0017 has a disconnected subtraction branch and rerouted selection path; task 0104 has an altered feedback connector below the Update block.

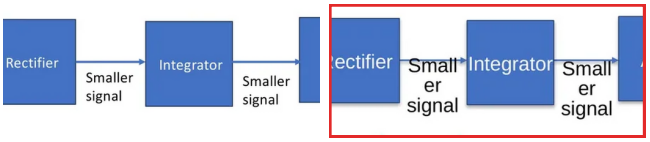
**a**




reference candidate

**Failed** $G_{\text{sem}}$
$S = 0$
The layout and palette are preserved, but the NO branch after the subtraction decision is left dangling and the template-selection link is reversed. The candidate therefore describes a different processing path.

**b**

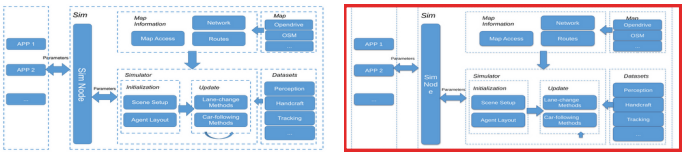



reference candidate

**Failed** $G_{\text{rend}}$
$S = 0$
The topology is preserved and legibility is not. Connector labels break mid-word, into Small / er / signal, across most of the figure, leaving a slide that states the process and cannot be read.

Figure 7: Examples of hard-gate failures: (a) incorrect process connections from Astra High on task 0017; (b) illegible connector labels from Luna Max. Both fail all three judge rounds.

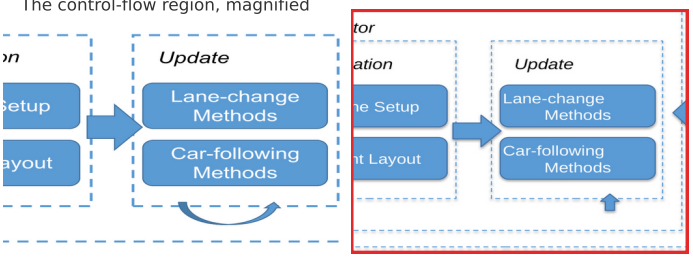
**a**



reference candidate

The pair as scored, at full size: the simulator layout, dataset blocks and main arrows are reproduced.

**b** The control-flow region, magnified




reference candidate

**Failed** $G_{\text{sem}}$
$S = 0$
The candidate replaces the U-shaped feedback path below Update with a short upward arrow. The slide remains visually convincing at full size, but the control flow now asserts a different process.

Figure 8: Semantic failure on task 0104 (Astra High): (a) full reference and reconstruction; (b) enlarged control-flow region showing the altered feedback connection.

### D.1 Issue frequencies and round support

Table 7 reports the fifteen largest penalties among the 5,714 gate-passing reconstructions. The Cases column counts model–task pairs with at least one occurrence; Category ratio averages the largest affected fraction across passing rounds for each case. Single-round support is the fraction of cases in which only one passing round records the issue. Across all issue types, text and local graphics account for 10,191 and 10,849 recorded occurrences, respectively.

Table 7: The fifteen largest aggregate detail penalties among 5,714 gate-passing reconstructions. Cases may contain multiple issue types.

| Issue type | Dimension | Severity | Cases | Category ratio | Total penalty | Single-round (%) |
|---|---|---|---|---|---|---|
| `unexpected_wrap` | Text | Medium | 1,864 | 0.577 | 5,723.9 | 6.8 |
| `outside_text_box` | Text | High | 658 | 0.319 | 3,441.8 | 35.7 |
| `node_aspect_ratio` | Graphics | Medium | 802 | 0.655 | 2,614.5 | 53.7 |
| `horizontal_alignment` | Text | Medium | 737 | 0.694 | 2,446.7 | 42.1 |
| `text_connector_overlap` | Text | High | 385 | 0.342 | 2,061.8 | 55.3 |
| `text_node_overlap` | Text | High | 379 | 0.306 | 1,929.8 | 43.5 |
| `substituted_icon` | Graphics | Medium | 542 | 0.794 | 1,877.0 | 36.5 |
| `node_size` | Graphics | Medium | 520 | 0.654 | 1,699.7 | 67.1 |
| `font_size` | Text | Low | 1,315 | 0.905 | 1,593.2 | 46.1 |
| `edge_touch` | Text | Medium | 574 | 0.308 | 1,473.1 | 62.7 |
| `node_fill` | Graphics | Low | 1,200 | 0.848 | 1,421.2 | 44.2 |
| `mid_word_break` | Text | Medium | 434 | 0.575 | 1,343.5 | 48.2 |
| `vertical_alignment` | Text | Medium | 382 | 0.798 | 1,325.8 | 76.2 |
| `text_color` | Text | Medium | 388 | 0.737 | 1,307.0 | 45.1 |
| `node_spacing` | Layout | Medium | 360 | 0.781 | 1,237.6 | 73.3 |

## E MATCHED REASONING AND CONFIGURATION WEIGHTING

**Common-pass comparisons.** For each effort pair, we retain tasks that pass under both settings and subtract the lower-effort detail score from the higher-effort score. Table 8 compares high with the highest available effort. Pass-rate differences use all 500 tasks. Intervals are percentile intervals from 10,000 paired task resamples, seed 20260925, with no multiplicity adjustment. Generated outputs and judge findings remain fixed.

Table 8: High versus highest effort on shared passing tasks. $N$ is common-pass count; $\Delta p$ uses all 500 tasks.

| Model | Higher effort | $N$ | $\Delta$ detail | 95% CI | $\Delta p$ (pp) |
|---|---|---|---|---|---|
| Opus 5 | max | 147 | 1.76 | [0.15, 3.43] | 0.0 |
| Sonnet 5 | xhigh | 35 | 0.58 | [-2.75, 3.87] | 0.4 |
| Luna | max | 85 | 4.39 | [1.88, 7.01] | 12.6 |
| Sol | max | 109 | 4.13 | [2.43, 5.91] | 20.8 |
| Terra | max | 66 | 2.83 | [0.22, 5.47] | 18.0 |
| Astra | max | 289 | 0.98 | [0.18, 1.75] | -9.8 |

For Astra High versus Max, 289 tasks pass at both settings. Max gains 66 passing tasks and loses 115, giving a net decrease of 49 out of 500 tasks, or 9.8 percentage points. On the shared passing tasks, detail increases by 0.98 points. This distinguishes changes in the passing set from changes in reconstruction quality (Section 3.3).

**Configuration weighting.** Equal-family weighting first averages available effort settings within each family, then averages the ten family scores. Highest-effort selection retains one configuration per family; available effort levels differ. Observed-best selection chooses each family's highest-scoring setting on these same 500 tasks. Table 9 quantifies these choices. Its confidence intervals use paired 500-task resamples.

Table 9: Aggregate score under alternative configuration menus. Median is over configuration or family means, as specified.

| Weighting / selection | Mean | 95% CI | Median |
|---|---|---|---|
| 36 configurations, equal weight | 28.18 | [26.72, 29.62] | 24.38 |
| 10 families, mean over available efforts | 31.91 | [30.40, 33.48] | 27.14 |
| 10 families, highest available effort | 38.26 | [36.39, 40.10] | 33.50 |
| 10 families, best observed effort | 39.36 | [37.49, 41.25] | 34.69 |

## F GENERATION TRAJECTORIES

A review turn renders or inspects the candidate; a revision turn edits the deck. The trajectory analysis covers 27 configurations in total: 26 across five effort sweeps and one GPT-5.5 configuration (Table 10). The covered families are Sol, Terra, Luna, Opus, Sonnet, and GPT-5.5; Sol Medium is excluded. Correlations summarize the observed effort sweeps, in which reasoning effort varies alongside inspection, revision, and score.

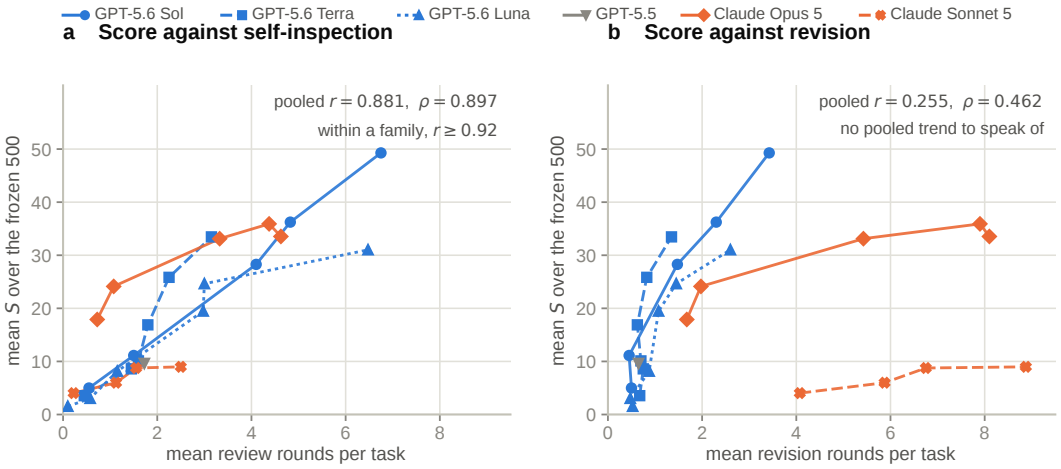



Figure 9: Mean score versus review turns (left) and revision turns (right) for 27 configurations. Connected points follow each model's reasoning-effort sweep.

Table 10: Correlations between mean score and generation behavior. A dash indicates an undefined within-model correlation for the single GPT-5.5 configuration.

| | | Review turns | | Revision turns | |
|---|---|---|---|---|---|
| Model | Configurations | Pearson $r$ | Spearman $\rho$ | Pearson $r$ | Spearman $\rho$ |
| All matched configurations | 27 | 0.881 | 0.897 | 0.255 | 0.462 |
| Claude Opus 5 | 5 | 0.947 | 0.900 | 0.927 | 0.900 |
| Claude Sonnet 5 | 4 | 0.922 | 1.000 | 0.913 | 1.000 |
| GPT-5.6 Luna | 6 | 0.946 | 1.000 | 0.920 | 0.943 |
| GPT-5.6 Sol | 5 | 0.998 | 1.000 | 0.979 | 0.900 |
| GPT-5.6 Terra | 6 | 0.955 | 1.000 | 0.798 | 0.600 |
| GPT-5.5 | 1 | – | – | – | – |

# G JUDGE RELIABILITY AND RANKING UNCERTAINTY

## G.1 VALIDATION DESIGN AND JUDGE REPEATABILITY

Protocol development uses 50 tasks. Judge selection uses a disjoint set of 50 tasks, with four reconstructions spanning quality per task, giving 200 validation cases. Three annotators independently label these cases, blinded to model identity, reasoning effort, judge outputs, and one another's labels. Both subsets belong to the 500-task evaluation set. Table 2 reports agreement on the judge-selection cases, and Figure 10 summarizes gate agreement and repeatability. The Wilson interval is computed over the 200 cases from the reported aggregate counts.

Prompts, taxonomy, severity bases, dimension caps, ratio staircase, and aggregation were frozen before the initial leaderboard recomputation and retained for subsequent configurations.

Human gate agreement and issue-set agreement measure different decisions. On the 52 cases passed by all three annotators, their issue sets match exactly in 9 cases (17.3%), with mean pairwise Jaccard similarity 0.392. Human validation therefore covers gates and issue sets; continuous detail scores are examined through the judge-variation and scoring-policy analyses below.

Table 3 compares independent runs and two Luna High triplets. Majorities from rounds 1–3 and independently collected rounds 4–6 agree on 192/200 cases; three are gated only by the first triplet and five only by the second. The additional triplet evaluates gates without Stage 3 enumeration.

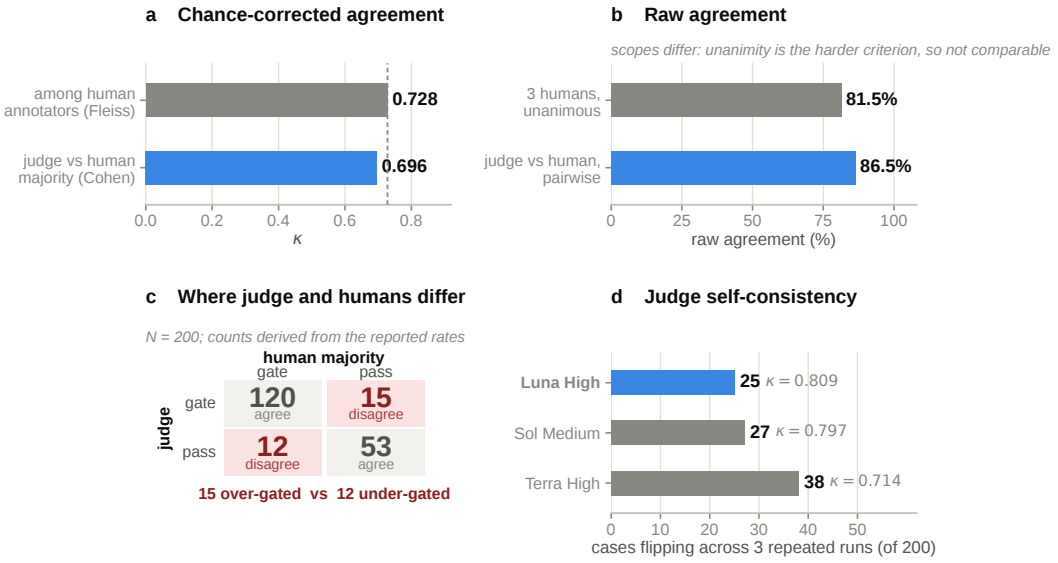



Figure 10: Gate validation on 200 cases: (a) chance-corrected agreement; (b) raw agreement; (c) confusion matrix; (d) single-round gate consistency across three repeats.

### G.2 Within-triplet detail variation

We analyze the three recorded judge rounds for each of Astra's 2,500 artifacts. Reapplying consensus aggregation reproduces all reported scores exactly. Table 11 uses the 1,645 artifacts whose three rounds all reach Stage 3. Each round is scored separately using its own findings and category ratios. Range is the maximum minus minimum of these three scores; pairwise difference averages the three absolute score differences per artifact.

Table 11: Detail-score variation within existing Astra judge triplets, conditional on all three rounds passing.

| Effort | Artifacts | Median range | 95th percentile | Mean pairwise difference |
|---|---|---|---|---|
| Low | 279 | 2.50 | 8.52 | 2.03 |
| Medium | 324 | 1.87 | 7.76 | 1.84 |
| High | 366 | 1.50 | 7.50 | 1.62 |
| XHigh | 354 | 1.06 | 7.50 | 1.30 |
| Max | 322 | 0.62 | 7.50 | 1.27 |
| All five | 1,645 | 1.25 | 7.50 | 1.60 |

These measurements describe variation within a recorded triplet, conditional on all three rounds passing. The independently repeated triplets in Table 3 instead measure the repeatability of aggregate gate decisions.

### G.3 Sensitivity to gate-vote threshold

We reaggregate Astra's 7,500 recorded rounds, requiring one, two, or three rounds to pass both judged gates. Each policy retains the original union/max detail aggregation over non-gated rounds; all 71 deterministic artifact failures remain zero. Majority reaggregation exactly reproduces all 2,500 recorded results. Gate votes disagree in 258 cases across 170 task IDs. Requiring one pass admits 97 majority failures; requiring unanimity rejects 161 majority passes.

Table 12 reports the resulting scores and passing counts.

Table 12: Astra scores under three gate-vote thresholds. Parentheses give passing cases out of 500.

| Effort | At least one pass | Majority (reported) | Unanimous pass |
|---|---|---|---|
| Low | 63.15 (335) | 59.42 (315) | 52.81 (279) |
| Medium | 72.60 (382) | 68.16 (358) | 61.78 (324) |
| High | 80.56 (421) | 77.34 (404) | 70.20 (366) |
| XHigh | 75.94 (392) | 72.60 (374) | 68.85 (354) |
| Max | 72.10 (373) | 68.76 (355) | 62.59 (322) |

The per-case unanimity score is the recorded majority score multiplied by an indicator for three passing rounds, so requiring unanimity can only preserve or decrease each score. One-pass voting can admit cases rejected by the majority.

### G.4 Sensitivity to detail scoring policy

We recompute 18,000 cases from the frozen merged findings, reproducing all original scores before changing one policy at a time. Hard-gate outcomes remain fixed. Two-round support discards issues observed in fewer than two non-gated rounds. Unit multiplier sets $m(\rho) = 1$ while retaining severity bases. Equal weights averages the three retained dimension fractions and rescales to 100. Category maximum retains the largest subtype penalty within each dimension and category, then applies the original cap. This variant measures sensitivity to accumulating multiple subtype deductions within one category.

Table 13 compares each variant with the original configuration ordering.

Table 13: Sensitivity of 36 configuration scores to detail policies. $\Delta$ measures score change from the original rubric.

| Policy | Kendall $\tau$ | Median $\Delta$ | Max. $\lvert\Delta\rvert$ | Reversed pairs |
|---|---|---|---|---|
| Two-round issue support | 0.975 | 1.77 | 4.53 | 8/630 |
| Unit affected-ratio multiplier | 0.971 | 1.70 | 4.15 | 9/630 |
| Equal dimension weights | 0.997 | 0.13 | 0.55 | 1/630 |
| Maximum per issue category | 0.981 | 1.21 | 2.74 | 6/630 |

Among the 5,714 passed cases, two-round support raises mean detail by 5.89 points and category maximum by 3.87. All four variants retain Astra High in first place and preserve the top-five set. These comparisons hold the generated artifacts and judge findings fixed.

### G.5 PAIRED TASK BOOTSTRAP

At each benchmark size, we draw 10,000 task samples with replacement and evaluate every configuration on the same sample. Table 4 gives rank correlation, exact top-$k$ membership, and score-interval widths; Figure 11 plots their dependence on sample size. At 500 tasks, 5 of 35 adjacent score differences have 95% intervals excluding zero. Astra High's lead over Astra XHigh is 4.74 points, with paired 95% CI $[-0.07, 9.53]$.

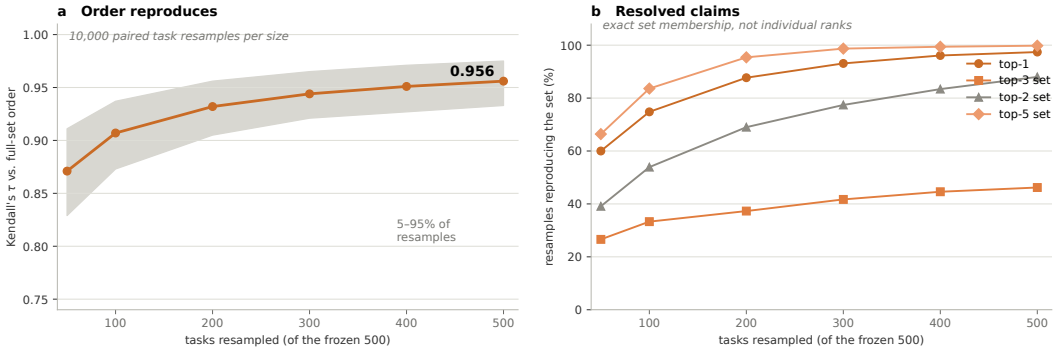



Figure 11: Ranking stability under paired task resampling: (a) Kendall's $\tau$, with a 5–95% band; (b) exact top-$k$ set recovery rates.

## H SCORED RECONSTRUCTION EXAMPLES

Table 14 decomposes four gate-passing reconstructions. Figure 12 shows the Luna Medium example; Figure 13 shows the other three.

Table 14: Detail scores for four gate-passing reconstructions in Figures 12 and 13. Bold indicates the highest score in each column.

| System | Layout /30 | Text /40 | Graphics /30 | $S$ /100 | Dominant defect |
|---|---|---|---|---|---|
| GPT-5.6 Terra XHigh | 28.00 | **40.00** | 27.50 | **95.50** | Panel gradient lost; node fills shifted across all 13 coloured nodes |
| GPT-5.6 Terra Medium | 27.75 | **40.00** | 25.00 | 92.75 | Icons substituted; 15 rounded nodes squared; one title divider merged |
| GPT-5.6 Luna Medium | **28.75** | 23.49 | **27.75** | 79.99 | One text frame overruns its shape in four ways; cylinder re-oriented |
| GPT-5.6 Sol Max | **28.75** | 27.50 | 20.50 | 76.75 | Systematic text misalignment and mid-word breaks; one shape outline missing |

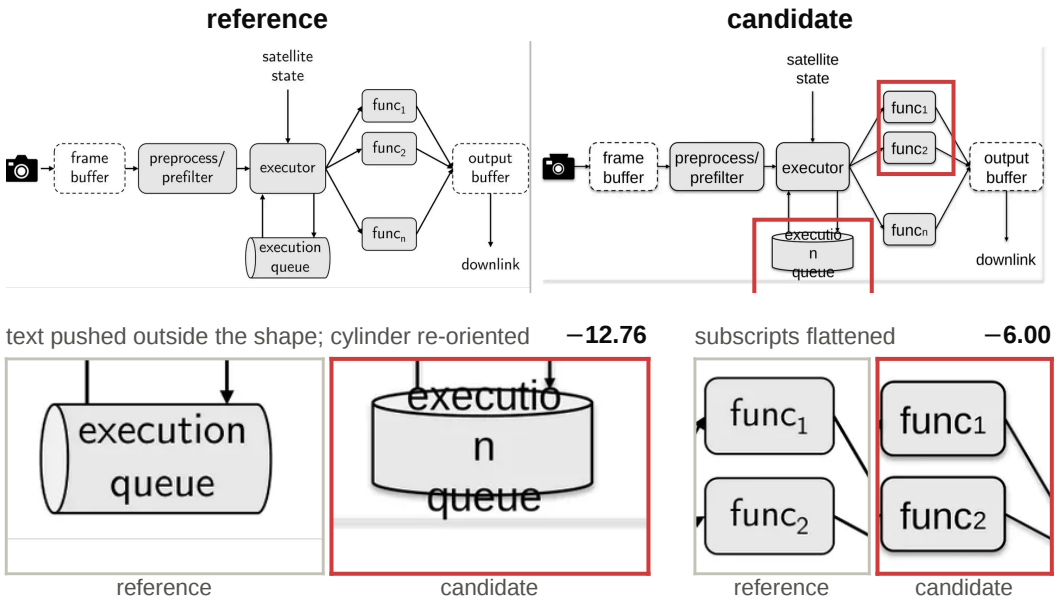
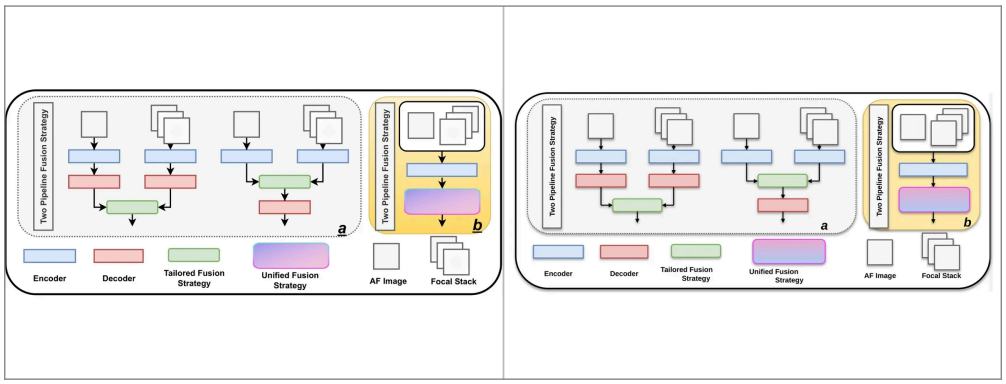



Figure 12: Luna Medium on task 0852 ($S = 79.99$). Enlarged regions show text overflow, flattened subscripts, and the reoriented cylinder.

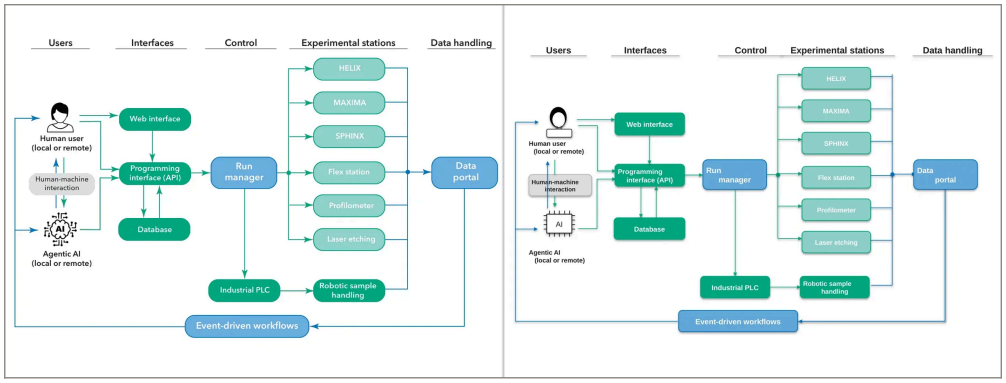
**GPT-5.6 Terra XHigh** $S = 95.50$

Panel gradient lost, node fills shifted across all 13 coloured nodes.




**GPT-5.6 Terra Medium** $S = 92.75$

Icons substituted, 15 rounded nodes squared, one divider merged.




**GPT-5.6 Sol Max** $S = 76.75$

Systematic text misalignment and mid-word breaks; one shape outline lost.

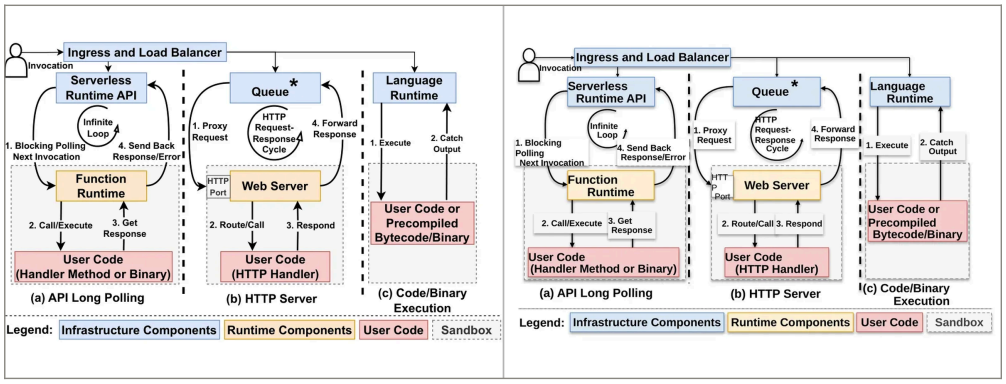



Figure 13: Three additional gate-passing reconstructions with different detail errors. Table 14 reports their dimension scores.

# I DETAIL SCORING AND TAXONOMY

**Structured findings and round aggregation.** Each Stage 3 finding records its dimension, category, issue subtype, location, description, and affected-instance count. The judge estimates one affected fraction per category, which is shared by the findings in that category. Affected counts are retained for inspection but do not directly enter the penalty. Only rounds that pass both judged gates contribute detail findings. If fewer than two rounds pass, the candidate receives zero, including when one round fails Stage 1 and another fails Stage 2. Otherwise, findings are merged by dimension, category, and issue subtype. A finding reported in any passing round enters the merged set, and repeated findings retain the largest affected fraction. This counts each issue type once within its category and dimension while preserving its widest reported extent. The contributing rounds and individual observations are retained for audit.

**Deductions for visual defects.** A finding's deduction depends on its severity and how much of the diagram it affects. Let $K_r$ contain findings from passing round $r$ and $K_d$ contain merged findings in dimension $d$. For finding $k$, $\rho_{rk} \in [0, 1]$ is the category-level affected fraction assigned to it in round $r$. Its merged fraction is

$$\rho_k = \max_{r:\, k \in K_r} \rho_{rk}. \tag{2}$$

The taxonomy assigns a severity base $b_k \in \{3.0, 1.5, 0.5, 0\}$ to high, medium, low, and diagnostic findings, respectively. The affected fraction scales this base by

$$m(\rho) = \begin{cases} 1.00, & 0 \le \rho \le 0.02, \\ 1.25, & 0.02 < \rho \le 0.10, \\ 1.50, & 0.10 < \rho \le 0.25, \\ 2.00, & 0.25 < \rho \le 0.50, \\ 2.50, & 0.50 < \rho \le 1. \end{cases} \tag{3}$$

Using fractions makes an issue affecting four of four nodes more costly than the same issue affecting four of forty. With dimension caps $(W_{\text{layout}}, W_{\text{text}}, W_{\text{graphics}}) = (30, 40, 30)$, the total penalty $P_d$ and retained score $S_d$ are

$$P_d = \min\left(W_d, \sum_{k \in K_d} \text{round}_2[b_k m(\rho_k)]\right), \qquad S_d = W_d - P_d, \tag{4}$$

where $\text{round}_2$ rounds each deduction to two decimals before summation. These three retained scores form the visual-quality term in Eq. (1). The deterministic calculation gives identical scores for identical merged findings and allows rubric weights to be changed without re-judging the images.

**Issue taxonomy.** Table 15 lists the issue subtypes and severity bases used in Eq. (4). Diagnostic entries are recorded without deductions.

Table 15: Stage 3 issue taxonomy and severity bases. Diagnostic findings carry zero penalty.

| Category | Severity base | Issue subtype |
|---|---|---|
| **Layout and composition** | | |
| Panel misalignment | High (3.0) | `missing_panel`, `extra_panel`, `duplicate_panel`, `substituted_panel`, `panel_order`, `panel_overlap`, `panel_clipping`, `conspicuous_blank_region` |
| Panel misalignment | Medium (1.5) | `canvas_crop`, `canvas_aspect_ratio`, `canvas_orientation`, `panel_position`, `panel_size`, `panel_proportion`, `panel_spacing`, `panel_alignment`, `separator_position`, `other_panel_layout` |
| Panel misalignment | Diagnostic (0.0) | `global_scale`, `global_centering`, `outer_whitespace` |
| Node layout deviation | High (3.0) | `missing_node`, `extra_node`, `duplicate_node`, `substituted_node`, `reading_order`, `grouping`, `nesting`, `hierarchy`, `node_overlap`, `node_occlusion` |

Table 15 (continued)

| Category | Severity base | Issue subtype |
|---|---|---|
| Node layout deviation | Medium (1.5) | node_position, node_spacing, node_alignment, node_distribution, grid_arrangement, padding, z_order, local_scale, local_mirroring, local_rotation, other_node_layout |
| Panel style error | Low (0.5) | panel_fill, panel_border_color, panel_border_width, panel_border_pattern, panel_opacity, panel_gradient, panel_texture, panel_clipping_mask, panel_corner_radius, panel_glow, panel_bevel, panel_outer_frame, panel_header_strip, panel_title_tab, panel_separator_style, other_panel_style |
| Panel style error | Diagnostic (0.0) | panel_shadow |
| **Text and typography** | | |
| Spelling and omission | High (3.0) | missing_text, extra_text, substituted_text, duplicated_text, reordered_text, incorrect_number, incorrect_identifier, math_symbol, greek_letter, subscript, superscript, special_glyph, wrong_label_assignment, other_text_content |
| Spelling and omission | Low (0.5) | capitalization, punctuation, abbreviation, accent, unit |
| Text overflow or edge touch | High (3.0) | text_clipping, text_truncation, outside_text_box, text_text_overlap, text_node_overlap, text_icon_overlap, text_connector_overlap, text_arrowhead_overlap, text_cell_overlap, cut_off_glyph, other_text_fit |
| Text overflow or edge touch | Medium (1.5) | edge_touch, insufficient_text_padding |
| Improper typography | Medium (1.5) | font_family, font_weight, text_color, text_contrast, line_height, baseline, math_typesetting, horizontal_alignment, vertical_alignment, text_anchoring, unexpected_wrap, mid_word_break, isolated_character_line, line_count, line_order, paragraph_width, text_rotation, text_orientation |
| Improper typography | Low (0.5) | font_size, font_style, text_opacity, letter_spacing, word_spacing, kerning, internal_padding, hyphenation, curved_text, text_blur, text_aliasing, other_typography |
| **Local graphics, nodes and connectors** | | |
| Node shape error | High (3.0) | missing_shape_primitive, extra_shape_primitive |
| Node shape error | Medium (1.5) | shape_type, node_aspect_ratio, node_size, node_rotation, node_orientation, node_skew, node_symmetry, notch_geometry, folded_corner, cutout, tab_geometry, port_geometry, handle_geometry, tail_geometry, pointer_geometry, other_node_geometry |
| Node shape error | Low (0.5) | node_corner_radius |
| Node style error | Low (0.5) | node_fill, node_stroke, node_border_width, node_border_pattern, node_color, node_opacity, node_transparency, node_gradient, node_texture, node_hatch, node_highlight, node_contrast, state_color, palette_mapping, three_dimensional_treatment, node_bevel, node_gloss, internal_z_order, style_consistency, other_node_style |
| Node shadow error | Diagnostic (0.0) | shadow_presence, shadow_offset, shadow_direction, shadow_blur, shadow_spread, shadow_color, shadow_opacity, glow_presence, glow_style, hard_vs_soft_shadow, other_shadow |

Table 15 (continued)

| Category | Severity base | Issue subtype |
|---|---|---|
| Vector and icon confusion | Medium (1.5) | missing_icon, extra_icon, substituted_icon, malformed_icon, mirrored_icon, rotated_icon, distorted_icon, icon_state, legend_symbol, legend_mapping, rasterized_vector, grid_dimensions, grid_count, primitive_count, primitive_order, mini_diagram_topology, chart_axes, chart_ticks, chart_legend, chart_series, chart_marker, raster_crop, raster_stretch, raster_replacement, other_local_graphic |
| Vector and icon confusion | Low (0.5) | vector_blur, vector_pixelation, jagged_vector, cell_size, cell_spacing, cell_border, cell_alignment, primitive_position, primitive_color, primitive_fill, primitive_pattern, raster_blur |
| Line and arrow details | High (3.0) | missing_connector, extra_connector, wrong_source_attachment, wrong_target_attachment, wrong_port, connector_direction, bidirectionality |
| Line and arrow details | Medium (1.5) | detached_endpoint, endpoint_side, arrowhead_endpoint, intermediate_arrowhead, arrowhead_count, arrowhead_shape, route_shape, bezier_curvature, bend_count, bend_location, bend_radius, loop_shape, bypass_shape, junction_dot, split_merge_marker, crossing_bridge, intersection_treatment, connector_overlap_text, connector_overlap_node, connector_clipping, border_confusion, other_connector |
| Line and arrow details | Low (0.5) | arrowhead_size, arrowhead_fill, arrowhead_outline, arrowhead_color, arrowhead_angle, parallel_spacing, line_pattern, dash_sequence, dash_gap, line_width, line_color, line_opacity, line_cap, line_join, line_blur, line_glow |
| Line and arrow details | Diagnostic (0.0) | line_shadow |