# EDITIKZ: SCIENTIFIC FIGURE EDITING FROM REVISION TRAJECTORIES

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

## ABSTRACT

Vision-language models (VLMs) have shown strong performance in generating scientific figures from text or images. However, publication-ready figures often require iterative refinement, making scientific figure editing an important yet largely unexplored step toward interactive figure creation. Existing approaches rely on costly proprietary agentic systems, focus primarily on evaluation, or construct training supervision from synthetically generated edits. Instead, we leverage naturally occurring scientific revision and development trajectories as a scalable source of supervision. To this end, we introduce DaEdiTikZ, the first large-scale dataset of revision-derived scientific figure edits, constructed by mining 391K plausible TikZ edit pairs from arXiv, GitHub, and TeX SE and inferring 781K directed edit instructions with a VLM conditioned on rendered figures and TikZ code. We further introduce DaEdiTikZ-Bench, a human-refined benchmark with 690 instances, and train two compact Qwen3.5-based EdiTikZ models (4B and 9B) by jointly learning image-to-TikZ reconstruction and instruction-conditioned editing, followed by reinforcement learning (RL) with complementary rewards for rendered fidelity and edit application. Automatic evaluation places our 9B model above all tested baselines, while human evaluation with 9 annotators and 4,320 ratings places it above GPT-5.6-Sol and on par with Gemini-3.1-Pro. Under severe out-of-distribution shifts, it remains competitive with GPT-5.6-Sol near its 2K training sequence-length regime. Models and datasets will be released.

## 1 INTRODUCTION

VLMs are increasingly capable of assisting researchers in multimodal tasks (Eger et al., 2026), including understanding and generating figures (Li et al., 2024b; Wang et al., 2024), tables (Moosavi et al., 2021), slides (Ge et al., 2025), and posters (Pang et al., 2025). These advances are driven by improvements in multimodal alignment (Liu et al., 2023), reasoning (Zhang et al., 2024; Huang et al., 2026), and agentic systems (Koh et al., 2024) that combine planning and tool use to tackle complex scientific workflows (Sun et al., 2026a). Despite this progress, publication-ready figures rarely emerge in a single generation and typically require revisions to their content, layout, and visual details, making scientific figure editing an important yet underexplored capability.

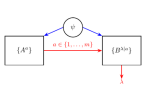
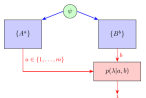
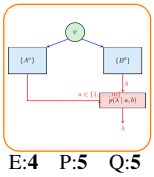
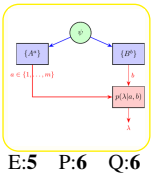
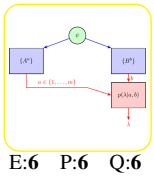
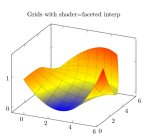
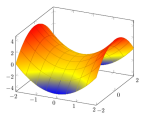
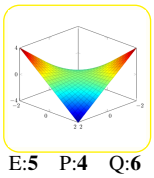
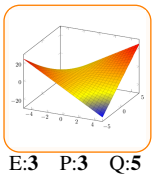
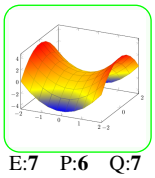
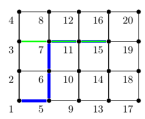
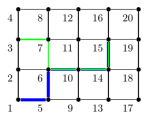
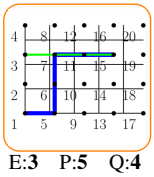
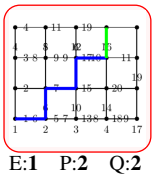
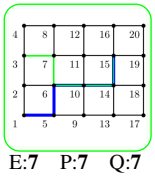
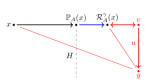
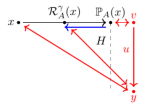
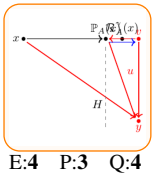
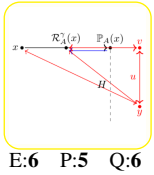
Graphics programming languages such as TikZ are the de facto standard in academia due to their precision, interpretability, and seamless integration into the LaTeX ecosystem. However, their diverse syntax and steep learning curve make them difficult for humans to master (Belouadi et al., 2024a). Prior work has focused on generating TikZ from text (Greisinger & Eger, 2026) or images (Belouadi et al., 2024b), whereas existing editing efforts rely on proprietary agentic systems (Lin et al., 2026b), target specialized domains such as charts (Zhao et al., 2025), or focus primarily on evaluation (Rahman et al., 2026; Bo et al., 2026). Large-scale training supervision remains limited and predominantly synthetic (Wang et al., 2026; Bo et al., 2026), motivating more realistic sources of editing supervision.

In this work, we take a different perspective. Scientific figures naturally evolve through iterative human revisions during research, paper writing, and community discussions. These revisions capture rich but previously overlooked expert decisions about how figures should change, yet remain unused



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

Table 1: Scientific figure edits by GPT-5.6-Sol and our EdiTikZ-9B models before and after RL. Models receive the source image and VLM-generated edit instruction. Human annotations score edit application (E), source preservation (P), and visual quality (Q). Overall quality: ■ very good, ■ good, ■ bad, ■ very bad. More examples are in Appendix A.5.4.

| Source | Edit Instruction | Ground Truth | GPT-5.6-Sol | EdiTikZ-9B | EdiTikZ-9B-RL |
|---|---|---|---|---|---|
|  | The circular node labeled $\psi$ at the top is changed from having a black outline to having a light green fill. The rectangular boxes labeled $A^a$ and $B\cdots$ are changed from having a white fill to having a light blue fill. The label inside the right-hand box is changed from $\{B^{\lambda\mid a}\}$ to $\{B^{b}\}$. A new rectangular box with a light red fill, labeled $p(\lambda\mid a,b)$, is added below the right-hand box. The horizontal red arrow labeled $a \in \{1,\ldots,m\}$ is rerouted. It now starts from the bottom of the left box, goes down, turns right, and points to the left side of the new red box. A new vertical red arrow labeled $b$ is added, connecting the bottom of the right box to the top of the new red box. The vertical red arrow labeled $\lambda$ is moved to originate from the bottom of the new red box instead of the right-hand box. |  |  E:4 P:5 Q:5 |  E:5 P:6 Q:6 |  E:6 P:6 Q:6 |
|  | The title 'Grids with shader=faceted interp' at the top of the plot is removed. The surface geometry is changed from a custom set of coordinates to a mathematical function (hyperbolic paraboloid), resulting in a saddle shape with a peak in the top-left middle and a valley in the bottom middle. The axis ranges are modified: the x-axis now spans from -2 to 2, the y-axis spans from -2 to 2, and the z-axis spans from -4 to 4. |  |  E:5 P:4 Q:6 |  E:3 P:3 Q:5 |  E:7 P:6 Q:7 |
|  | The thick blue path (path1) is rerouted. In the first image, it goes from node 6 up to node 7, then right to node 11, and finally to node 15. In the second image, the path from node 6 goes right to node 10, then right to node 14, and finally up to node 15. The thin green path (path2) is rerouted. In the first image, it goes from node 7 right to node 11, then right to node 15. In the second image, the path from node 7 goes down to node 6, then right to node 10, then right to node 14, and finally up to node 15. |  |  E:3 P:5 Q:4 |  E:1 P:2 Q:2 |  E:7 P:7 Q:7 |
|  | The point labeled $\mathcal{R}^{\gamma}_{A}(x)$ is moved from the right of $\mathbb{P}_A(x)$ to the left of it (between $x$ and $\mathbb{P}_A(x)$). The black arrow moves through the point $\mathcal{R}^{\gamma}_{A}(x)$ to point to $\mathbb{P}_A(x)$ while the blue arrow from $\mathbb{P}_A(x)$ to $\mathcal{R}^{\gamma}_{A}(x)$ is reversed and positioned slightly below the black arrow. The red point $v$ and the red point $y$ are moved horizontally to the left, closer to the vertical dashed line $H$. The red double-headed arrow now points from $\mathbb{P}_A(x)$ to $v$. The two red lines connecting the top points ($x$ and $\mathcal{R}^{\gamma}_{A}(x)$) to the bottom point $y$ are changed to double-headed arrows. |  |  E:6 P:5 Q:5 |  E:4 P:3 Q:4 |  E:6 P:5 Q:6 |

as supervision for multimodal models. Inspired by how early instruction-tuning methods leverage naturally occurring software revisions (Muennighoff et al., 2024; Wei et al., 2024; Li et al., 2024a), we introduce a scalable framework that recovers plausible scientific figure revision pairs from real-world repositories. Applied to TikZ figures from arXiv, GitHub, and TeX SE, this yields DaEdiTikZ, the first large-scale dataset of revision-derived scientific figure edits, containing 391K edit pairs. Since figures and their programs already exist, we synthesize only the missing edit instruction using a VLM conditioned on rendered figures and TikZ code, yielding 781K directed editing instances. We also introduce DaEdiTikZ-Bench, a human-refined benchmark with 690 editing instances.

Building on DaEdiTikZ, we train two small Qwen3.5-based EdiTikZ models that jointly learn to reconstruct figures as TikZ and edit them from instructions, followed by RL with complementary rewards for rendered fidelity and edit application. Across three human-evaluation criteria on DaEdiTikZ-Bench, our 9B model performs above GPT-5.6-Sol and on par with Gemini-3.1-Pro. Post-training gains transfer even beyond the 2K-token training horizon to substantially more complex out-of-distribution figures from SPIQA and CharXiv. Table 1 shows representative editing results. Our key contributions are as follows:

- **Revision-Derived Supervision:** We introduce a scalable framework for recovering plausible edit pairs from naturally occurring collections of related scientific figures.
- **Dataset and Benchmark:** We release DaEdiTikZ with 391K plausible edit pairs, yielding 781K directed editing instances, and DaEdiTikZ-Bench with 690 human-refined instances.
- **Editing-Specific Post-Training:** We jointly train reconstruction and editing during SFT followed by GDPO with complementary rewards for rendered fidelity and edit application.
- **EdiTikZ Models:** We release 4B and 9B EdiTikZ models. EdiTikZ-9B-RL outperforms all tested baselines in automatic evaluation and surpasses GPT-5.6-Sol in human evaluation.

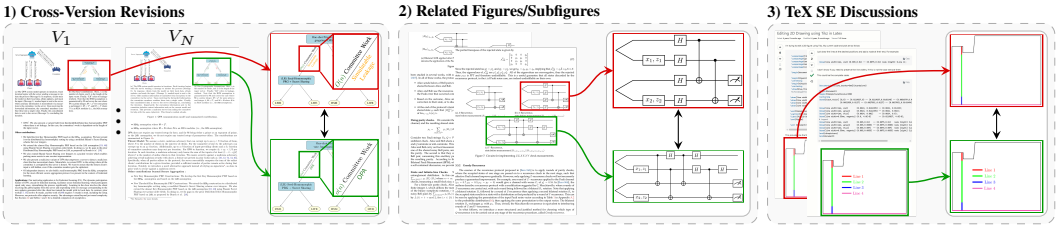



Figure 1: Sources of figure editing supervision. We recover plausible edit pairs from cross-version revisions (left), related figures and subfigures within shared scientific contexts (middle), and iterative refinements in TeX SE discussions (right). Red and green highlight paired variants in their sources.

## 2 Related Work

**Generating Scientific Figures with Graphics Programs** For TikZ, prior work generates code from text (Belouadi et al., 2024a; 2025; Greisinger & Eger, 2026), or reconstructs it from images (Belouadi et al., 2024b; ZENG et al., 2026; Lin et al., 2026a). Other work targets SVG (Rodriguez et al., 2025b; Wu et al., 2024; Zou et al., 2024), Python (Ni et al., 2025; Yang et al., 2024), multiple visualization languages (Zhang et al., 2025; Ni et al., 2026), or generates diagrams from documents (Zhu et al., 2026; Guan et al., 2026; Mondal et al., 2024). However, these methods generate figures from scratch instead of modifying them.

**Scientific Figure Editing** Prior work studies editing of charts (Zhao et al., 2025; Li et al., 2026a), SVGs (Kuchař et al., 2025; Lin et al., 2026b), TikZ (Wei et al., 2025), and rasters (Zhao et al., 2026) using agentic systems. Concurrent work includes SciDiagramEdit (Sun et al., 2026b), which mines 364 arXiv revision pairs for benchmark construction and agentic SVG editing, S1-Omni-Image (Li et al., 2026b), which unifies scientific-image understanding, generation, and editing, and DisciplineGen-1M (Wang et al., 2026), which constructs OCR-based synthetic editing supervision. VisEditBench (Rahman et al., 2026) evaluates Matplotlib/Vega-Lite editing, while DiagramMMU (Bo et al., 2026) evaluates image-conditioned TikZ editing with template-based modifications. In contrast, we construct large-scale training supervision from plausible pairs of human-authored scientific figures and synthesize only the missing edit instruction.

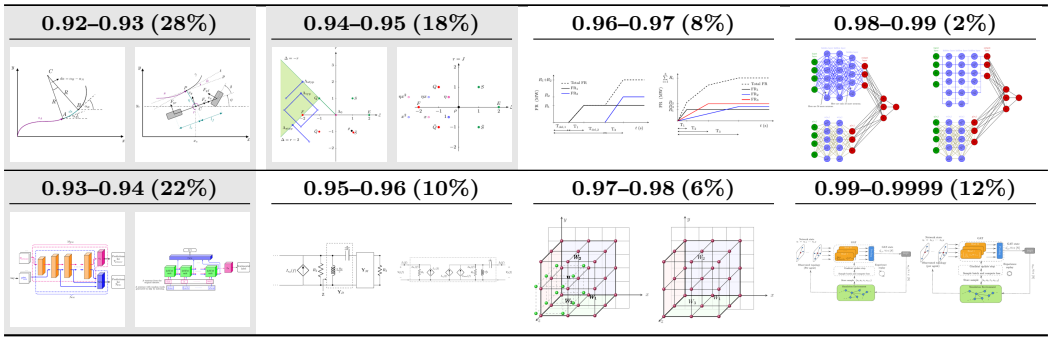
**RL from Rendering Feedback** Rendered-feedback RL has been applied to SVG (Rodriguez et al., 2025a; ZENG et al., 2026; Rodriguez et al., 2026) and TikZ generation (Greisinger & Eger, 2026; Lin et al., 2026a), using perceptual, domain-specific, code-based, and self-consistency rewards. Recent methods use VLM feedback to compare charts (Tang et al., 2026) or answer instance-specific visual questions (Yang et al., 2026). Scientific figure editing instead requires preserving source content while applying localized changes. We therefore combine global rendered similarity with a source-conditioned, target-reference-free VLM verifier for individual requested edits.

## 3 Dataset and Benchmark

**Revision-Derived Editing Supervision** Our key observation is that plausible scientific figure edits naturally arise throughout scientific revision and development processes, including (i) figures modified across arXiv or GitHub versions, (ii) related (sub-)figures in the same paper or repository, (iii) alternative TikZ programs retained in source files but not rendered in the document, and (iv) iterative refinements in TeX SE discussions (Figure 1). Exact figure lineage is difficult to recover reliably as figures may be added, removed, renamed, reordered, or moved across files, while surrounding anchors such as captions, references, and related text can also change. We therefore identify semantically similar pairs within shared scientific contexts and retain plausible editing transformations.

**Collecting Scientific Revision Traces** We extend DaTikZ-V4 (Greisinger & Eger, 2026) by recovering TikZ from all historical versions of arXiv submissions containing `tikzpicture`, `circuitikz`, or `tikzcd`. We apply the TikZilla preprocessing pipeline, including document expansion, subfigure extraction, code standardization, dynamic package inclusion, filtering, rendering, and deduplication on the standardized TikZ body. Across 91K arXiv submissions, 38K contain

Table 2: Examples of scientific-figure edit pairs across semantic-similarity intervals and the percentage of implausible editing transformations in each interval. Gray cells denote excluded intervals.



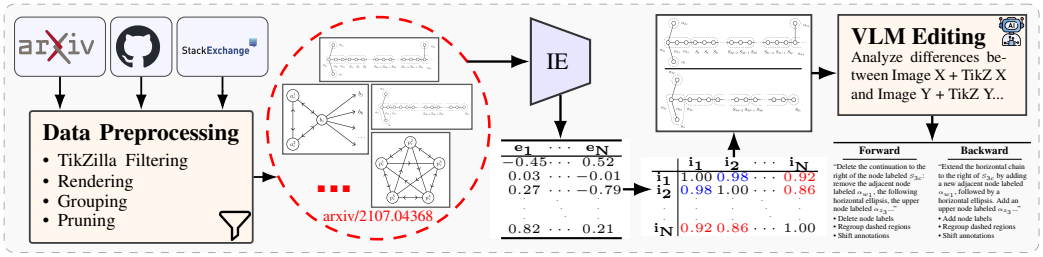




Figure 2: Construction pipeline for DaEdiTikZ. Scientific figures are collected and standardized, grouped by their scientific context, embedded with a scientific image encoder, paired according to cosine similarity, and passed to a VLM to produce bidirectional editing instructions.

at least two versions with modified TikZ code. Historical versions contribute 0.91M additional figures, increasing the unique arXiv corpus from 1.47M to 2.38M. Combined with the existing GitHub and TeX SE portions, this yields a candidate corpus of 2.91M unique TikZ figures.

**Recovering Plausible Edit Pairs** We group figures by arXiv submission across versions, GitHub repository, and TeX SE discussion thread, yielding 222K groups, of which 123K contain at least two unique figures. We prune groups above the 90th size percentile (>10) and compute within-group cosine similarities using `DeTikZify-V2`'s image encoder. To determine the filtering threshold, we manually evaluate 50 pairs in each of eight similarity intervals (0.92–0.9999, width 0.01) and retain intervals containing fewer than 15% implausible transformations (Table 2). This produces 430,442 candidate pairs from 87,051 contributing groups, connecting 589,986 unique figures.

**Inferring Edit Instructions** Because both endpoint figures are human-authored, we synthesize only the missing edit instruction using Qwen3.6-27B conditioned jointly on their renders and TikZ code. For each of the 430,442 candidate pairs, we infer both directions ($A \to B$ and $B \to A$), producing 860,884 candidate directional editing instances. The VLM classifies each direction as `ok`, `invalid`, or `identical`. For accepted transformations, it decomposes the transformation into atomic edits with an intent (`add`, `remove`, or `modify`), operation (`text`, `annotation`, `geometry`, `data`, `style`, or `structure`), and natural-language description. Requiring both directions to be accepted yields DaEdiTikZ with 390,516 figure pairs and 781,032 directional editing instances. Each instance contains 4.2 atomic edits on average, with descriptions averaging 22.3 words per atomic edit. Detailed analysis of DaEdiTikZ is in the Appendix A.2.

**Dataset Quality Analysis** To validate instruction inference, two annotators evaluate 125 revision pairs, including 35 overlapping samples for agreement (Figure 3). They identify edit plausibility, omissions, hallucinations, and attribute, numeric, or spatial misinterpretations ($\kappa = 0.82$), and rate overall quality on a 1–5 Likert scale (weighted $\kappa = 0.79$). Overall, 98% of retained transformations are plausible and 82.9% of instructions are rated good (4) or very good (5). While 50% contain at least one error, 34% contain omissions and 33% misinterpretations, whereas only 10% contain

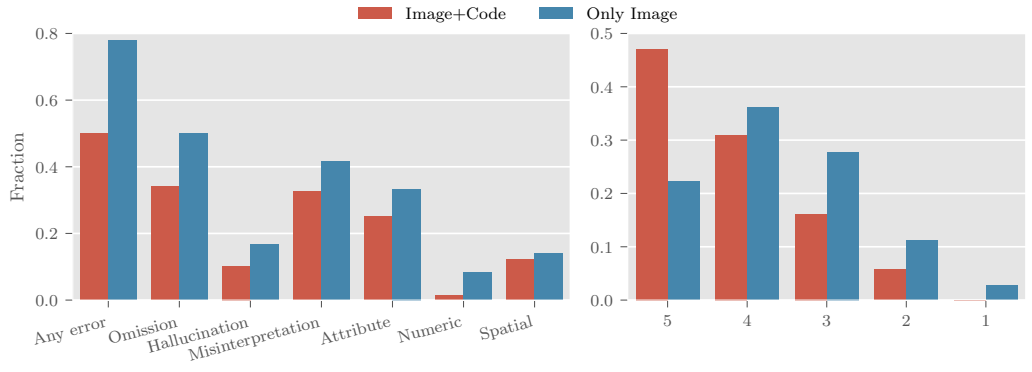



Figure 3: Human evaluation of inferred edit instructions. Left: error rates with and without TikZ-code grounding, decomposed into omissions, hallucinations, and attribute, numeric, and spatial misinterpretations. Right: overall instruction quality rated on a 1–5 Likert scale.

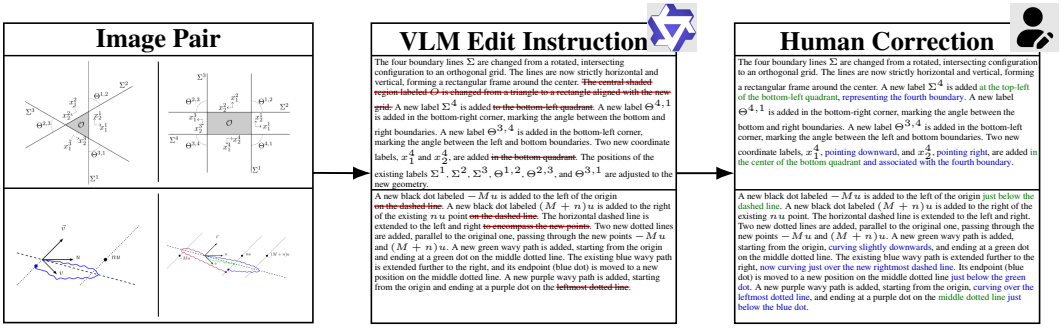



Figure 4: Examples of human-refined benchmark instructions. ~~Red strikethrough~~ marks removed errors, green marks corrections, and blue marks added omissions.

hallucinations. To quantify the benefit of code grounding, we repeat the analysis without TikZ code on 90 annotations. The error rate increases from 50% to 80%, with omissions increasing by 16 percentage points and numeric misinterpretations from 1% to 8.5%, indicating that code provides complementary grounding.

**DaEdiTikZ-Bench** We construct DaEdiTikZ-Bench as a human-refined evaluation set from arXiv submissions published between March and June 2026. To ensure diversity, we retain one pair per submission, with 100 pairs from each of five similarity intervals (0.95–0.96, ..., 0.99–0.9999) and 50 pairs per group size from one to ten. We manually inspect all 500 candidates and remove quality issues, trivial edits, and rendering artifacts, leaving 345 revision pairs and 690 editing instances. Six annotators correct every VLM-generated instruction by removing hallucinations, correcting misinterpretations, and adding omissions (Figure 4).

## 4 Editing-Specific Post-Training

**Joint Reconstruction and Editing SFT** DaEdiTikZ provides source figure–instruction–TikZ target triplets $(I_s, u, y)$, where $y = (y_1, \ldots, y_T)$ and $I_t$ is the rendered target figure. We minimize:

$$\mathcal{L}_{\text{edit}}(\theta) = \mathbb{E}_{(I_s,u,y)\sim\mathcal{D}_{\text{edit}}}\left[-\sum_{t=1}^{T}\log p_\theta(y_t \mid y_{<t}, I_s, u)\right] \tag{1}$$

Since editing requires reconstructing the source figure while selectively modifying it, we jointly train with an equal number of image-to-TikZ reconstruction samples from DaTikZ-V4. Reconstruction uses the same objective over $(I_t, y) \sim \mathcal{D}_{\text{rec}}$, strengthening the shared image-to-TikZ mapping while exposing the model to a broader distribution of scientific figures and TikZ programs.

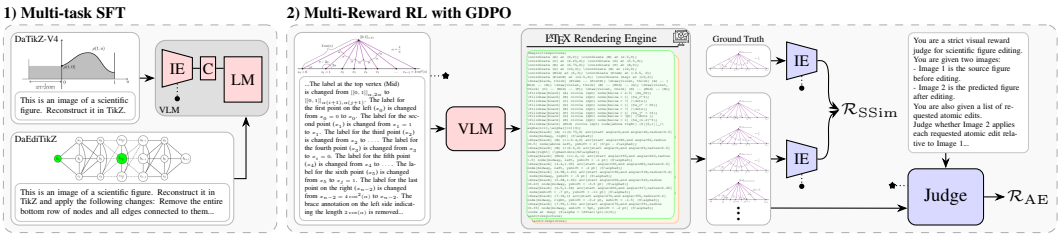



Figure 5: Two-stage training pipeline. Left: Multi-task SFT jointly trains on equal amounts of editing (DaEdiTikZ) and reconstruction (DaTikZ-V4) data. Right: GDPO optimizes on a disjoint DaEdiTikZ subset using SelfSim from the frozen SFT vision encoder and instruction-following from a VLM judge.

**Editing-Specific Rewards** We further optimize the resulting SFT model using rewards computed from sampled TikZ rollouts $\hat{y}$ and their renderings $\hat{I}$. Unlike TikZilla, which trains a separate scientific image encoder (Greisinger & Eger, 2026), we reuse a frozen copy of the SFT model's vision encoder. SFT already adapts this encoder to scientific figures through joint editing and reconstruction. We freeze it during RL to prevent reward hacking. Given patch embeddings $\mathbf{x} = \{x_i\}_{i=1}^{N}$ and $\mathbf{z} = \{z_j\}_{j=1}^{M}$ of $I_t$ and $\hat{I}$, respectively, we compute:

$$D_{ij} = 1 - \cos(x_i, z_j), \qquad d_{\text{EMD}}(\mathbf{x}, \mathbf{z}) = \min_{F \geq 0} \sum_{i=1}^{N} \sum_{j=1}^{M} F_{ij} D_{ij} \tag{2}$$

subject to uniform marginals $\sum_j F_{ij} = 1/N$ and $\sum_i F_{ij} = 1/M$. The SelfSim reward is:

$$\mathcal{R}_{\text{SSim}} = \text{clip}\left(1 + 2\tanh[-d_{\text{EMD}}(\mathbf{x}, \mathbf{z})], 0, 1\right) \tag{3}$$

However, target similarity alone is insufficient for editing. First, DaEdiTikZ contains similar source–target pairs, allowing high $\mathcal{R}_{\text{SSim}}$ from preserving unchanged content without applying the requested edits. Second, VLM-inferred instructions may contain omissions or inaccuracies, such that the target may not exactly realize the instruction and can penalize valid instruction-following outputs. We therefore introduce a complementary reference-free instruction-following reward $\mathcal{R}_{\text{IF}}$. A Qwen3.6-27B judge receives $(I_s, u, \hat{I})$ and scores each of the $K$ atomic edits in $u$ with $v_k \in \{0, 1\}$. We set $\mathcal{R}_{\text{IF}} = \frac{1}{K}\sum_k v_k(I_s, u, \hat{I})$, giving proportional credit for partially applied instructions. Finally, we define compilation and format validity as $\mathcal{R}_{\text{Comp}} = \mathbb{1}[\text{compile}(\hat{y})]$ and $\mathcal{R}_{\text{Fmt}} = \mathbb{1}[\text{valid_format}(\hat{y})]$, where the latter requires the expected standalone TikZ structure (`\documentclass[tikz]{standalone}`, ..., `\begin{document}`, ..., `\end{document}`). Compilation and format validity gate both rewards: $r_m = \mathcal{R}_{\text{Comp}}\mathcal{R}_{\text{Fmt}}\mathcal{R}_m$ for $m \in \mathcal{M} = \{\text{SSim}, \text{IF}\}$, assigning failed rollouts zero reward. Figure 5 summarizes the two-stage pipeline.

**Multi-Reward Optimization with GDPO** $\mathcal{R}_{\text{SSim}}$ provides dense target-similarity feedback, whereas $\mathcal{R}_{\text{IF}}$ measures discrete atomic edit application. Since standard multi-reward GRPO aggregates rewards before group normalization, rewards with larger variance can dominate the learning signal. We instead use Group reward-Decoupled Normalization Policy Optimization (GDPO) (Liu et al., 2026), which normalizes each reward independently before aggregation. For $G$ rollouts, GDPO computes:

$$A_m^{(i,j)} = \frac{r_m^{(i,j)} - \text{mean}_{j'}[r_m^{(i,j')}]}{\text{std}_{j'}[r_m^{(i,j')}] + \varepsilon}, \qquad A_{\text{sum}}^{(i,j)} = \sum_{m \in \mathcal{M}} w_m A_m^{(i,j)} \tag{4}$$

Following GDPO, we batch-normalize $A_{\text{sum}}$ to obtain $\widehat{A}_{\text{sum}}$ and optimize the clipped policy objective:

$$\mathcal{J}_{\text{GDPO}}(\theta) = \mathbb{E}_{q_i \sim \mathcal{D}_{\text{edit}}}\left[\frac{1}{G}\sum_{j=1}^{G}\frac{1}{L}\sum_{t=1}^{|o_{i,j}|}\min\left(\frac{\pi_\theta\left(\hat{y}_{i,j,t} \mid x_i, \hat{y}_{i,j}^{<t}\right)}{\pi_{\theta_{\text{old}}}\left(\hat{y}_{i,j,t} \mid x_i, \hat{y}_{i,j}^{<t}\right)}\widehat{A}_{\text{sum}}^{(i,j)},\right.\right.$$
$$\left.\left.\text{clip}\left(\frac{\pi_\theta\left(\hat{y}_{i,j,t} \mid x_i, \hat{y}_{i,j}^{<t}\right)}{\pi_{\theta_{\text{old}}}\left(\hat{y}_{i,j,t} \mid x_i, \hat{y}_{i,j}^{<t}\right)}, 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}}\right)\widehat{A}_{\text{sum}}^{(i,j)}\right) - \beta\, D_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\theta_{\text{SFT}}}\right)\right]$$

Implementation details are provided in the Appendix A.3.

# 5 EXPERIMENTS

**Setup** We use disjoint group-level splits, reserving 27K DaEdiTikZ trajectories for RL and using the remaining 754K editing trajectories together with 754K DaTikZ-V4 reconstruction samples for SFT (1.51M instances total). Thus, figures from the same group never occur across training stages. SFT updates all parameters, whereas RL updates only the language model while freezing the vision encoder and embeddings. Unless stated otherwise, evaluation uses the 690 human-refined DaEdiTikZ-Bench instances, which are disjoint from all training groups.

**Models** We evaluate six proprietary VLMs—GPT-5.6-Sol, GPT-5.5, GPT-5.4, Gemini-3.1-Pro, Gemini-3.6-Flash, and Gemini-3.5-Flash—and eight open-source VLMs: Qwen3.6-27B[1], Qwen3.5 (27B, 9B, and 4B) (Qwen Team, 2026), Qwen3-VL (8B and 4B) (Bai et al., 2025a), and Qwen2.5-VL (7B and 3B) (Bai et al., 2025b). We apply SFT to all models up to 9B parameters except Qwen2.5-VL-7B, yielding our EdiTikZ family. Subscripts distinguish earlier Qwen generations. RL is applied to EdiTikZ-4B and EdiTikZ-9B, denoted EdiTikZ-4B-RL and EdiTikZ-9B-RL.

**Metrics** We evaluate code similarity with TeX Edit Distance (TED) and perceptual similarity with DreamSim (DSim) (Fu et al., 2023). Following VLM-based evaluation (Ku et al., 2024), GPT-5.5 scores three editing-specific criteria: *(i) Edit Application (EA)*, measuring correct application of requested edits; *(ii) Source Preservation (SP)*, measuring preservation of unaffected content; and *(iii) Visual Quality (VQ)*, measuring legibility and publication readiness. Scores are produced on a 0–10 scale and normalized to $[0, 1]$. We also report compilation rate (CR) and average output tokens (AT). The aggregate score (Avg) averages $1 - \mathrm{TED}$, DSim, EA, SP, and VQ.

# 6 RESULTS

**Automatic Evaluation** Across all architectures, SFT improves Avg by 0.186–0.363 and compilation rate by 19.0–39.3 percentage points. RL further improves EdiTikZ-4B/9B to 0.674/0.726 Avg. EdiTikZ-4B-RL reaches proprietary-level performance, while EdiTikZ-9B-RL achieves the highest overall score (Table 3). Additional results are in Appendix A.5.

MODEL RANKINGS REVERSE AFTER SFT Qwen3.5-4B/9B initially underperform Qwen3-VL-4B/8B (0.249/0.345 vs. 0.314/0.354 Avg), but surpass them after SFT (0.612/0.643 vs. 0.538/0.540), showing that zero-shot editing performance is a poor proxy for task-specific adaptation potential.

VISUAL CORRECTNESS VS. CODE SIMILARITY Unlike prior TikZ-generation RL, where TED improves after RL (Greisinger & Eger, 2026; ZENG et al., 2026), ours worsens despite consistent gains across rendered metrics. This suggests that TED is a poor primary metric for editing, where visually correct outputs may differ substantially at the code level.

**Human Evaluation** We conduct a human evaluation with 9 qualified annotators, who rate predictions from eight models on EA, SP, and VQ using a 1–7 Likert scale (Figure 6). Each annotator evaluates 20 randomized figure groups with 10% overlap, yielding 4,320 ratings. Quadratic-weighted agreement is high ($\kappa_{\mathrm{EA}} = 0.801$, $\kappa_{\mathrm{SP}} = 0.786$, $\kappa_{\mathrm{VQ}} = 0.832$).

HUMAN EVALUATION CONFIRMS POST-TRAINING GAINS SFT raises the combined score of Qwen3.5-4B/9B from 6.40/8.78 to 14.71/15.18, with RL further improving it to 16.14/17.43, with gains across all three criteria. EdiTikZ-9B-RL nearly matches Gemini-3.1-Pro (17.43 vs. 17.72) and performs above GPT-5.6-Sol (16.75). SFT narrows the 4B–9B gap from 2.38 to 0.47 points, whereas RL widens it to 1.29 points.

AUTOMATIC METRICS ALIGN WITH HUMANS Our aggregate metric correlates strongly with combined human judgments ($\rho = 0.823$). While TED correlates poorly ($\rho = 0.374$), DSim and the criterion-specific EA, SP, and VQ metrics each reach $\rho \approx 0.77$. $\mathcal{R}_{\mathrm{SSim}}$ correlates more strongly with human SP/VQ, whereas adding $\mathcal{R}_{\mathrm{IF}}$ improves EA correlation by 0.045 and raises overall correlation from 0.812 to 0.827, showing the intended complementarity.

[1]GPT-5.6-Sol, GPT-5.5, GPT-5.4, Gemini 3.1 Pro, Gemini 3.6 Flash, Gemini 3.5 Flash, Qwen3.6-27B

Table 3: Results on DaEdiTikZ-Bench. **Bold** is best while underline is second-best.

| Model | TED↓ | DSim↑ | EA↑ | SP↑ | VQ↑ | Avg↑ | CR↑ | AT↓ |
|---|---|---|---|---|---|---|---|---|
| GPT-5.6-Sol | 0.764 | 0.796 | 0.735 | 0.775 | 0.823 | 0.673 | 88.3% | 485 |
| GPT-5.5 | 0.765 | 0.829 | 0.744 | 0.790 | 0.849 | 0.689 | 92.2% | 488 |
| GPT-5.4 | 0.763 | 0.741 | 0.674 | 0.706 | 0.762 | 0.624 | 84.6% | 487 |
| Gemini-3.1-Pro | 0.716 | 0.795 | **0.761** | 0.798 | 0.828 | 0.693 | 86.5% | **384** |
| Gemini-3.6-Flash | 0.740 | 0.665 | 0.654 | 0.677 | 0.698 | 0.591 | 72.2% | 399 |
| Gemini-3.5-Flash | 0.737 | 0.676 | 0.656 | 0.678 | 0.718 | 0.598 | 74.0% | 427 |
| Qwen3.6-27B | 0.768 | 0.675 | 0.470 | 0.521 | 0.635 | 0.507 | 79.0% | 547 |
| Qwen3.5-27B | 0.757 | 0.677 | 0.482 | 0.524 | 0.636 | 0.512 | 79.0% | 490 |
| Qwen2.5-VL-7B | 0.797 | 0.388 | 0.139 | 0.158 | 0.296 | 0.237 | 50.6% | 689 |
| Qwen2.5-VL-3B | 0.810 | 0.329 | 0.062 | 0.062 | 0.213 | 0.171 | 45.9% | 747 |
| EdiTikZ-3B | 0.707 | 0.700 | 0.309 | 0.332 | 0.525 | 0.432 | 82.2% | 623 |
| Qwen3-VL-4B | 0.788 | 0.494 | 0.232 | 0.245 | 0.386 | 0.314 | 61.9% | 651 |
| EdiTikZ-4B$_{\text{Qwen3}}$ | 0.644 | 0.790 | 0.445 | 0.474 | 0.627 | 0.538 | 89.3% | 567 |
| Qwen3-VL-8B | 0.772 | 0.543 | 0.281 | 0.293 | 0.427 | 0.354 | 67.2% | 509 |
| EdiTikZ-8B | 0.676 | 0.765 | 0.462 | 0.509 | 0.639 | 0.540 | 86.2% | 579 |
| Qwen3.5-4B | 0.806 | 0.411 | 0.146 | 0.188 | 0.304 | 0.249 | 51.4% | 726 |
| EdiTikZ-4B | 0.629 | 0.813 | 0.552 | 0.609 | 0.714 | 0.612 | 90.7% | 542 |
| EdiTikZ-4B-RL | 0.642 | 0.871 | 0.633 | 0.706 | 0.803 | 0.674 | 95.2% | 494 |
| Qwen3.5-9B | 0.781 | 0.523 | 0.241 | 0.311 | 0.430 | 0.345 | 64.1% | 599 |
| EdiTikZ-9B | **0.628** | 0.834 | 0.598 | 0.658 | 0.753 | 0.643 | 92.0% | 545 |
| EdiTikZ-9B-RL | 0.676 | **0.893** | 0.734 | **0.815** | **0.865** | **0.726** | **96.8%** | 488 |

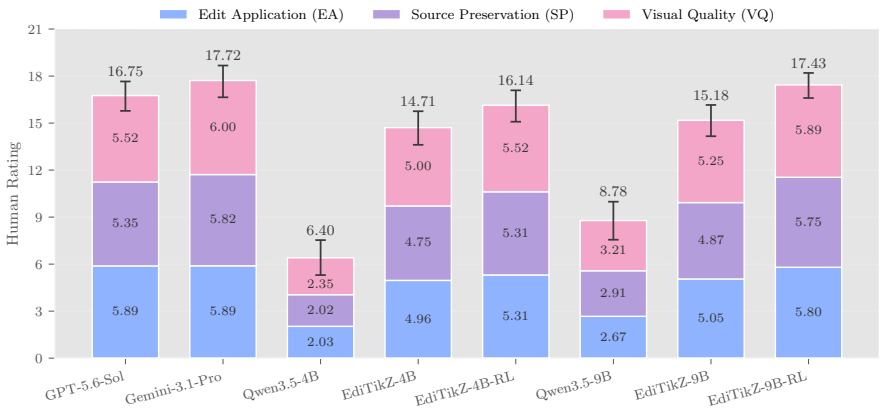



Figure 6: Likert-scale (1-7) across three evaluation criteria (EA, SP, and VQ) with 95% confidence intervals for eight VLMs (4 baseline, 4 fine-tuned).

**Ablations: Data Mixtures** Table 4 compares reconstruction and editing mixtures on two VLMs. Joint training performs best for 3B (0.432 Avg vs. 0.392 editing-only) and matches sequential training for 8B (0.539/0.540), with both exceeding editing-only (0.531). Thus, reconstruction consistently improves editing, while joint training additionally retains both capabilities.

**Ablations: Rewards and GDPO** Table 5 ablates our rewards and optimization algorithm. $\mathcal{R}_{\text{IF}}$ outperforms $\mathcal{R}_{\text{SSim}}$, by +0.031 EA. Combining both with GRPO adds +0.009 Avg, with GDPO adding another +0.038, supporting independent normalization of the complementary rewards.

**Generalization under Severe Distribution Shift** We stress-test EdiTikZ on SPIQA Pramanick et al. (2024) and CharXiv Wang et al. (2024), which contain complex architectural diagrams, multi-panel plots, tables, schematics, and charts across diverse scientific domains, often produced with tools such as Matplotlib, MATLAB, ggplot, and Plotly rather than TikZ. We sample 300 SPIQA and 600 CharXiv figures and manually remove examples requiring external data, leaving 190 and 497 instances, respectively. Since neither dataset provides editing pairs, we use GPT-5.6-Sol to generate synthetic edit instructions and evaluate predictions reference-free using EA, SP, VQ, CR, and AT. OOD generations require 3–5× more tokens than DaEdiTikZ-Bench and frequently exceed the 2K-

Table 4: Ablation on DaEdiTikZ-Bench for data-mixing strategies on two VLMs.

| VLM | Variant | TED↓ | DSim↑ | EA↑ | SP↑ | VQ↑ | Avg↑ | CR↑ | AT↓ |
|---|---|---|---|---|---|---|---|---|---|
| Qwen2.5-VL-3B | Base | 0.810 | 0.329 | 0.062 | 0.062 | 0.213 | 0.171 | 45.9% | 747 |
| | Only Recon | 0.751 | 0.615 | 0.016 | 0.277 | 0.426 | 0.317 | 68.4% | 800 |
| | Only Edit | 0.741 | 0.626 | 0.296 | 0.309 | 0.472 | 0.392 | 73.9% | 723 |
| | Recon→Edit | 0.720 | 0.647 | 0.272 | 0.297 | 0.480 | 0.395 | 76.2% | 686 |
| | Recon+Edit | **0.707** | **0.700** | **0.309** | **0.332** | **0.525** | **0.432** | **82.2%** | **623** |
| Qwen3-VL-8B | Base | 0.772 | 0.543 | 0.281 | 0.293 | 0.427 | 0.354 | 67.2% | **509** |
| | Only Recon | 0.709 | 0.737 | 0.157 | 0.495 | 0.592 | 0.454 | 85.1% | 646 |
| | Only Edit | 0.713 | 0.785 | 0.451 | 0.493 | 0.638 | 0.531 | **90.6%** | 557 |
| | Recon→Edit | 0.685 | **0.769** | 0.459 | **0.512** | **0.640** | 0.539 | 87.1% | 593 |
| | Recon+Edit | **0.676** | 0.765 | **0.462** | 0.509 | 0.639 | **0.540** | 86.2% | 579 |

Table 5: Ablation on DaEdiTikZ-Bench for reward functions and algorithms using EdiTikZ-9B.

| Variant | TED↓ | DSim↑ | EA↑ | SP↑ | VQ↑ | Avg↑ | CR↑ | AT↓ |
|---|---|---|---|---|---|---|---|---|
| Post-SFT | **0.628** | 0.834 | 0.598 | 0.658 | 0.753 | 0.643 | 92.0% | 545 |
| $\mathcal{R}_{\text{SSim}}$ | 0.667 | 0.872 | 0.633 | 0.706 | 0.805 | 0.670 | 96.1% | 489 |
| $\mathcal{R}_{\text{IF}}$ | 0.671 | 0.866 | 0.664 | 0.721 | 0.814 | 0.679 | 95.2% | 488 |
| Both (w. GRPO) | 0.662 | 0.879 | 0.671 | 0.728 | 0.822 | 0.688 | 96.4% | **476** |
| Both (w. GDPO) | 0.676 | **0.893** | **0.734** | **0.815** | **0.865** | **0.726** | **96.8%** | 488 |

token completion limit used during post-training. We stratify examples by mean generation length across all four models (Figure 7), ensuring identical examples within each bin.

RL GAINS INCREASE WITH GENERATION LENGTH For short generations (<1K), SFT provides most of the gain over Qwen3.5-9B. With increasing length, SFT gains diminish while the additional benefit from RL grows, dominating from 1.5–4K tokens. RL also maintains >80% compilation through 3–4K, consistently exceeding GPT-5.6-Sol, whereas SFT compilation degrades steadily.

COMPETITIVE WITHIN THE TRAINING HORIZON Within the trained ≤2K regime, EdiTikZ-9B-RL remains within 0.1 Judge Avg of GPT-5.6-Sol across all bins. Beyond 2K, EdiTikZ degrades faster and the gap widens, although SFT and RL gains persist throughout the 2–8K regime.

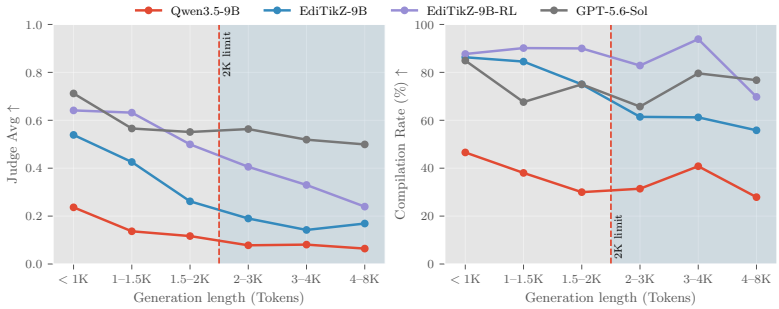



Figure 7: OOD performance on SPIQA and CharXiv combined by generation length. Judge Avg is $(\text{EA} + \text{SP} + \text{VQ})/3$. The red dashed line marks the 2K-token post-training limit.

## 7 CONCLUSION, LIMITATIONS, AND FUTURE WORK

We introduced a scalable framework for recovering naturally occurring scientific-figure revisions from arXiv, GitHub, and TeX SE, instantiated as DaEdiTikZ, a large-scale real-world TikZ editing dataset. We further introduced the human-refined DaEdiTikZ-Bench and EdiTikZ, a family of 4–9B models trained with multi-task SFT and multi-reward RL. EdiTikZ-9B-RL leads automatic evaluation and reaches comparable human ratings to the strongest proprietary system. Post-training gains transfer to substantially more complex SPIQA and CharXiv figures, even beyond the 2K-token training horizon. Overall, naturally occurring revision trajectories provide effective supervision for small, open scientific-figure editing models competitive with much larger proprietary systems.

DaEdiTikZ inherits instruction noise, including omissions and misinterpretations, and its mined pairs are plausible edits rather than guaranteed chronological revisions. Performance also degrades for long OOD generations, motivating post-training on more complex figures. Evaluation in this regime is itself limited by synthetic instructions and potentially less reliable reference-free judging. Beyond these limitations, our visualization-language-agnostic revision-mining framework may extend to Matplotlib or LaTeX tables, while access to source programs would enable localized editing without full reconstruction. Revision trajectories could also support comparative VQA, retrieval, and representation learning, helping unify generation and editing within general-purpose scientific visualization models.

## AI USE STATEMENT

We did not use AI to design or refine the dataset and methodological approaches, to search for relevant literature, to provide feedback on experiments, or to analyze data or interpret results. Generative AI tools were used to create editing instructions for the construction of the dataset, the benchmark, and the out-of-distribution dataset, as explicitly described in Section 3. In addition, we used generative AI tools to improve the grammar, wording, and readability of the manuscript (including figures, tables, and equations) as well as to assist with debugging and editing the code written by the authors. All AI-assisted work was reviewed and verified by the authors. We therefore take responsibility for the final content of this work, including text, statements, code, and artifacts created with the help of generative AI.

## REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Jonas Belouadi, Anne Lauscher, and Steffen Eger. AutomaTikZ: Text-guided synthesis of scientific vector graphics with TikZ. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=v3K5TVP8kZ.

Jonas Belouadi, Simone Paolo Ponzetto, and Steffen Eger. DeTikZify: Synthesizing graphics programs for scientific figures and sketches with TikZ. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=bcVLFQCOjc.

Jonas Belouadi, Eddy Ilg, Margret Keuper, Hideki Tanaka, Masao Utiyama, Raj Dabre, Steffen Eger, and Simone Ponzetto. TikZero: Zero-shot text-guided graphics program synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17793–17806, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Belouadi_TikZero_Zero-Shot_Text-Guided_Graphics_Program_Synthesis_ICCV_2025_paper.html.

Weihao Bo, Shan Zhang, Yanpeng Sun, Jie Liu, Yongke Yao, Jinhao Du, Wei He, Kai Zou, Zechao Li, and Jingdong Wang. Diagram-mmu: A multi-modal benchmark for scientific diagrams, 2026. URL https://arxiv.org/abs/2608.12262.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402, 2022. URL https://api.semanticscholar.org/CorpusID:253581213.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Guanzhou Chen, Erfei Cui, Changyao Tian, Danni Yang, Ganlin Yang, Yu Qiao, Hongsheng Li, Gen Luo, and Hongjie Zhang. Scaleedit-12m: Scaling open-source image editing data generation via multi-agent framework, 2026. URL https://arxiv.org/abs/2603.20644.

Steffen Eger, Yong Cao, Jennifer D'Souza, Andreas Geiger, Christian Greisinger, Stephanie Gross, Yufang Hou, Brigitte Krenn, Anne Lauscher, Yizhi Li, Chenghua Lin, Nafise Sadat Moosavi, Wei Zhao, and Tristan Miller. Transforming science with large language models: A survey on ai-assisted scientific discovery, experimentation, content generation, and evaluation, 2026. URL https://arxiv.org/abs/2502.05151.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 50742–50768. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9f09f316a3eaf59d9ced5ffaefe97e0f-Paper-Conference.pdf.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2902–2911, June 2025.

Christian Greisinger and Steffen Eger. Tikzilla: Scaling text-to-tikz with high-quality data and reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=rJv2byEWA3.

Yaohan Guan, Pristina Wang, Najim Dehak, Alan L. Yuille, Jieneng Chen, and Daniel Khashabi. Genfig1: Visual summaries of scholarly work as a challenge for vision-language models. *ArXiv*, abs/2604.04172, 2026. URL https://api.semanticscholar.org/CorpusID:287201464.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Xu Tang, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models, 2026. URL https://arxiv.org/abs/2503.06749.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.50. URL https://aclanthology.org/2024.acl-long.50/.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. VIEScore: Towards explainable metrics for conditional image synthesis evaluation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12268–12290, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.663. URL https://aclanthology.org/2024.acl-long.663/.

Josef Kuchař, Marek Kadlčík, Michal Spiegel, and Michal Štefánik. Vectoredits: A dataset and benchmark for instruction-based editing of vector graphics, 2025. URL https://arxiv.org/abs/2506.15903.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From Word Embeddings to Document Distances. In *International Conference on Machine Learning*, volume 37, pp. 957–966, 2015. URL https://mlanthology.org/icml/2015/kusner2015icml-word/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Kaixin Li, Qisheng Hu, James Xu Zhao, Hui Chen, Yuxi Xie, Tiedong Liu, Michael Shieh, and Junxian He. InstructCoder: Instruction tuning large language models for code editing. In Xiyan Fu and Eve Fleisig (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 473–493, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. ISBN 979-8-89176-097-4. URL https://aclanthology.org/2024.acl-srw.52/.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal ArXiv: A dataset for improving scientific comprehension of large vision-language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14369–14387, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.775. URL https://aclanthology.org/2024.acl-long.775/.

Li Li, Ryan A. Rossi, Sungchul Kim, Sunav Choudhary, Franck Dernoncourt, Puneet Mathur, Zhengzhong Tu, and Yue Zhao. Charts are not images: On the challenges of scientific chart editing. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=259xBeNyDV.

Qingxiao Li, Zikai Wang, Qingli Wang, and Nan Xu. S1-omni-image: A unified model for scientific image understanding, generation, and editing, 2026b. URL https://arxiv.org/abs/2606.24441.

Juekai Lin, Yun Zhu, Honglin Lin, Sijing Li, Tianwei Lin, Zheng Liu, Xiaoyang Wang, Wenqiao Zhang, and Lijun Wu. Scientific graphics program synthesis via dual self-consistency reinforcement learning, 2026a. URL https://arxiv.org/abs/2604.06079.

Zhen Lin, Qiujie Xie, Minjun Zhu, Shichen Li, Qiyao Sun, Enhao Gu, Yiran Ding, Ke Sun, Fang Guo, Panzhong Lu, Zhiyuan Ning, Yixuan Weng, and Yue Zhang. Autofigure-edit: Generating editable scientific illustration, 2026b. URL https://arxiv.org/abs/2603.06674.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization, 2026. URL https://arxiv.org/abs/2601.05242.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022.

Ishani Mondal, Zongxia Li, Yufang Hou, Anandhavelu Natarajan, Aparna Garimella, and Jordan Boyd-Graber. SciDoc2Diagrammer-MAF: Towards generation of scientific diagrams from documents guided by multi-aspect feedback refinement. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13342–13375, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.780. URL https://aclanthology.org/2024.findings-emnlp.780/.

Nafise Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. Scigen: a dataset for reasoning-aware text generation from scientific tables. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/149e9677a5989fd342ae44213df68868-Paper-round2.pdf.

Niklas Muennighoff, Qian Liu, Armel Randy Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro Von Werra, and Shayne Longpre. Octopack: Instruction tuning code large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mw1PWNSWZP.

Yuansheng Ni, Ping Nie, Kai Zou, Xiang Yue, and Wenhu Chen. VisCoder: Fine-tuning LLMs for executable python visualization code generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 2956–2983, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.160. URL https://aclanthology.org/2025.findings-emnlp.160/.

Yuansheng Ni, Songcheng Cai, Xiangchao Chen, Jiarong Liang, Zhiheng Lyu, Jiaqi Deng, Kai Zou, Ping Nie, Fei Yuan, Xiang Yue, and Wenhu Chen. Viscoder2: Building multi-language visualization coding agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4zoMnmZzh4.

Wei Pang, Kevin Qinghong Lin, Xiangru Jian, Xi He, and Philip Torr. Paper2poster: Towards multimodal poster automation from scientific papers. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/17337b1d5eeac8b59c80e025a552fa7a-Paper-Datasets_and_Benchmarks_Track.pdf.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. Spiqa: A dataset for multimodal question answering on scientific papers. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 118807–118833. Curran Associates, Inc., 2024. doi: 10.52202/079017-3773. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/d74033a247989e8f6f3bf9e0c9629fb5-Paper-Datasets_and_Benchmarks_Track.pdf.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Mizanur Rahman, Arshia Azimlu, Shadikur Rahman, Md Tahmid Rahman Laskar, Amran Bhuiyan, Shafiq Joty, and Enamul Hoque Prince. Viseditbench: Can vision-language models edit visualization code from multimodal feedback?, 2026. URL https://arxiv.org/abs/2608.10408.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pp. 1–16. IEEE, 2020.

Juan Rodriguez, Haotian Zhang, Abhay Puri, Rishav Pramanik, Aarash Feizi, Pascal Wichmann, Arnab Mondal, Mohammad R. Samsami, Rabiul Awal, Perouz Taslakian, Spandana Gella, Sai Rajeswar Mudumba, David Vazquez, Chris Pal, and Marco Pedersoli. Rendering-aware reinforcement learning for vector graphics generation. In D. Belgrave, C. Zhang,

H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 60496–60534. Curran Associates, Inc., 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/57126328c3b40cf618a34f1c5df24d8a-Paper-Conference.pdf.

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 *Conference on Computer Vision and Pattern Recognition*, pp. 16175–16186, 2025b. doi: 10.1109/CVPR52734.2025.01508. URL https://mlanthology.org/cvpr/2025/rodriguez2025cvpr-starvector/.

Juan A. Rodriguez, Haotian Zhang, Abhay Puri, Aly Shariff, Meng lin, Xiaoqing Xie, Tianyang Zhang, Rishav Pramanik, Sai Rajeswar, Perouz Taslakian, Spandana Gella, David Vazquez, Christopher Pal, and Marco Pedersoli. Vectorgym: A multi-task benchmark for SVG code generation and manipulation, 2026. URL https://openreview.net/forum?id=DBFbNT65xO.

Qiushi Sun, Zhoumianze Liu, Chang Ma, Zichen Ding, Fangzhi Xu, Zhangyue Yin, Haiteng Zhao, Zhenyu Wu, Kanzhi Cheng, Zhaoyang Liu, Jianing Wang, Qintong Li, Xiangru Tang, Tianbao Xie, Xiachong Feng, Xiang Li, Ben Kao, Wenhai Wang, Biqing Qi, Lingpeng Kong, and Zhiyong Wu. Scienceboard: Evaluating multimodal autonomous agents in realistic scientific workflows, 2026a. URL https://arxiv.org/abs/2505.19897.

Yasheng Sun, Zezi Zeng, Yifan Yang, Chong Luo, Wenyi Wang, Ziwei Liu, and Jürgen Schmidhuber. Scidiagramedit: Learning to edit scientific diagrams from paper revisions, 2026b. URL https://arxiv.org/abs/2607.15272.

Zitian Tang, Xu Zhang, Jianbo Yuan, Yang Zou, Varad Gunjal, Songyao Jiang, and Davide Modolo. Mm-recoder: Advancing chart-to-code generation with reinforcement learning and self-correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22164–22173, June 2026.

Peng Wang, Yichun Shi, Xiaochen Lian, Zhonghua Zhai, Xin Xia, Xuefeng Xiao, Weilin Huang, and Jianchao Yang. Seededit 3.0: Fast and high-quality generative image editing, 2025. URL https://arxiv.org/abs/2506.05083.

Zhaokai Wang, Mingxin Liu, Zirun Zhu, Ziqian Fan, Yiguo He, Mohan Zhang, Leyao Gu, Xiangyu Zhao, Ning Liao, Shaofeng Zhang, Xuanhe Zhou, Zhihang Zhong, Junchi Yan, and Xue Yang. Disciplinegen-1m: A large-scale dataset for multidisciplinary visual generation and editing, 2026. URL https://arxiv.org/abs/2607.02290.

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 A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 113569–113697. Curran Associates, Inc., 2024. doi: 10.52202/079017-3609. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/cdf6f8e9fd9aeaf79b6024caec24f15b-Paper-Datasets_and_Benchmarks_Track.pdf.

Jiayi Wei, Greg Durrett, and Isil Dillig. Coeditor: Leveraging repo-level diffs for code auto-editing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ALVwQjZRS8.

Jingxuan Wei, Cheng Tan, Qi Chen, Gaowei Wu, Siyuan Li, Zhangyang Gao, Linzhuang Sun, Bihui Yu, and Ruifeng Guo. From words to structured visuals: A benchmark and framework for text-to-diagram generation and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13315–13325, June 2025.

Rong Wu, Wanchao Su, and Jing Liao. Chat2svg: Vector graphics generation with large language models and image diffusion models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23690–23700, 2024. URL https://api.semanticscholar.org/CorpusID:274280554.

Haoyue Yang, Xuanle Zhao, Xuexin Liu, Feibing Jiang, and Yao Zhu. OmniDiagram: Advancing unified diagram code generation via visual interrogation reward. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 16430–16452, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.809. URL https://aclanthology.org/2026.findings-acl.809/.

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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11789–11804, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.701. URL https://aclanthology.org/2024.findings-acl.701/.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26125–26135, 2024. URL https://api.semanticscholar.org/CorpusID:274233770.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Xingchen ZENG, Zhewei Su, Hengming Zhang, Juyong Jiang, Jiazhi Xia, and Wei Zeng. Davinci: Reinforcing visual-structural syntax in MLLMs for generalized scientific diagram parsing. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=OAXECnLxuk.

Leixin Zhang, Steffen Eger, Yinjie Cheng, WEIHE ZHAI, Jonas Belouadi, Fahimeh Moafian, and Zhixue Zhao. Scimage: How good are multimodal large language models at scientific text-to-image generation? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ugyqNEOjoU.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models, 2024. URL https://arxiv.org/abs/2302.00923.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale, 2024. URL https://arxiv.org/abs/2407.05282.

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, 2026. URL https://arxiv.org/abs/2605.30611.

Xuanle Zhao, Xuexin Liu, Yang Haoyue, Xianzhen Luo, Fanhu Zeng, Jianling Li, Qi Shi, and Chi Chen. ChartEdit: How far are MLLMs from automating chart analysis? evaluating MLLMs' capability via chart editing. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3616–3630, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.185. URL https://aclanthology.org/2025.findings-acl.185/.

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

Bocheng Zou, Mu Cai, Jianrui Zhang, and Yong Jae Lee. VGBench: Evaluating large language models on vector graphics understanding and generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3647–3659, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.213. URL https://aclanthology.org/2024.emnlp-main.213/.

# A Appendix

## A.1 Related Work

### A.1.1 Image Editing

Most image-editing progress has been driven by diffusion models (Ho et al., 2020). Early methods preserve source structure through attention manipulation (Mokady et al., 2022), while InstructPix2Pix (Brooks et al., 2022) generates editing supervision by using GPT-3 for instructions and Stable Diffusion with Prompt-to-Prompt for paired before/after images. UltraEdit (Zhao et al., 2024) instead anchors automatically generated edits in real photographs and artworks and adds region-level supervision. AnyEdit (Yu et al., 2024) expands this setting to over 20 edit types with adaptive editing and automatic result selection. SeedEdit (Wang et al., 2025) combines synthetic pairs with professional editing workflows, traditional operators, and video-derived image pairs, while ScaleEdit-12M (Chen et al., 2026) scales data generation through an open-source multi-agent pipeline with adaptive synthesis and task-aware verification. These methods primarily target natural images rather than structurally constrained scientific graphics.

## A.2 Dataset and Benchmark

### A.2.1 Inferring Edit Instructions

For large-scale edit instruction inference, we use Qwen3.6-27B (non-thinking) conditioned jointly on image pair and TikZ code. Temperature is 0.1, top_p is 1.0, and output tokens are set to 1024. On 4× NVIDIA H100 (94 GB) GPUs with the vLLM (Kwon et al., 2023) framework, this took 11 days. The prompt is in Figure 8.

**VLM Instruction Generation**

```
Your task is to analyze the differences between two scientific figures given as:
- Image 1 + TikZ Code 1
- Image 2 + TikZ Code 2
Provide detailed descriptions of the edits needed to transform the first scientific
figure into the second.

Requirements:

0) Pair quality: Before listing any edits, decide whether Image 2 is a plausible
edited version of Image 1.
- "ok": same underlying figure or scene with a plausible edit path.
- "identical": no visible differences and no code changes implying visible
differences.
- "invalid": no plausible edit path or no shared figure identity (e.g., different
figure type, different number of panels, different main subject, or no shared
anchors).

1) Code-first, render-grounded: Use the TikZ code to discover candidate differences
between the figures. Only keep differences that produce a perceptible change in the
rendered images. Describe edits using human-visible anchors from the images (rendered
text, mathematics, relative location, appearance, connectivity), not code-only
identifiers or absolute coordinates.

2) Atomic and reconstructable: Each edit must describe exactly one logical change and
include enough concrete before→after detail that a human, given Image 1 alone, could
plausibly recreate the corresponding part of Image 2. If many elements change in the
same way, group them into one global edit, but enumerate the specific visible changes.

Output requirements:
- Report ONLY differences (no full image or code descriptions).
```

```
- Output ONLY valid JSON (no extra text and no trailing commas).

Return exactly the following JSON format:
{
  "pair_quality": "ok|identical|invalid",
  "edits": [
    {
      "intent": "add|remove|modify",
      "operation": "text|annotation|geometry|style|data|structure|other",
      "detailed_change": "specific human-visible description"
    }
  ]
}

If pair_quality is "identical", return exactly:
{ "pair_quality": "identical", "edits": [] }
If pair_quality is "invalid", return exactly:
{ "pair_quality": "invalid", "edits": [] }

Operation guidance:
- text: visible strings or mathematics (labels, titles, tick labels, axis names,
legend entries).
- annotation: explanatory or highlighting elements (arrows, callouts, braces,
highlight boxes, emphasis marks).
- geometry: position, shape, size, orientation, alignment, or spacing of visible
elements.
- style: appearance changes that do not alter the encoded meaning (color, line style,
thickness, opacity, font).
- data: changes to plotted or encoded values, including new points, curves, colormap
normalization, or category-to-color mappings.
- structure: high-level organization while preserving the same underlying figure
identity (e.g., added or removed panels, plot type changes of the same data, topology
changes in the same diagram).
- other: visible changes not covered above (e.g., clipping, layer ordering, global
transforms).

TikZ Code 1:
{code_1}

TikZ Code 2:
{code_2}

Your output JSON:
```

Table 6: Summary statistics for DaEdiTikZ. A retained pair has both directions with `pair_quality=ok`. Each valid direction forms a separate editing trajectory.

| Statistic | Value |
|---|---|
| Total groups | 87,051 |
| Unique figures | 589,986 |
| Candidate figure pairs / trajectories | 430,442 / 860,884 |
| Bidirectional valid pairs / trajectories | 390,516 / 781,032 |
| Total atomic edits | 3,282,215 |
| Mean atomic edits per trajectory (median; P95 / P99) | 4.20 (4; 9 / 13) |
| Mean atomic edit length | 22.3 words (127.0 characters) |
| Mean trajectory length | 93.6 words (533.9 characters) |

Table 7: Directional response quality and pair-level retention of all 430,442 candidate pairs.

| Level | Outcome | Count | Percentage |
|---|---|---|---|
| Forward direction | OK | 401,357 | 93.24% |
| | Invalid | 17,770 | 4.13% |
| | Identical | 4,151 | 0.96% |
| | Missing | 7,164 | 1.66% |
| Backward direction | OK | 400,892 | 93.13% |
| | Invalid | 18,191 | 4.23% |
| | Identical | 4,243 | 0.99% |
| | Missing | 7,116 | 1.65% |
| Candidate pair | Both directions valid | 390,516 | 90.72% |
| | Excluded | 39,926 | 9.28% |

DaEdiTikZ connects 590K unique figures through 430K candidate pairs from 87K context groups. Pair validation retains 90.7% of candidates, yielding 781K directional trajectories and 3.28M atomic edits. Figure reuse is limited. 69.3% of figures occur in only one pair and 97.3% in at most three (Table 6).

Quality is nearly symmetric across directions, with 93.2% of both forward and backward responses accepted. 390.5K pairs support supervision in both directions (Table 7).

Table 8: Frequency and description length of atomic edits by intent and operation.

| Dimension | Category | Atomic edits | Share | Words/edit | Characters/edit |
|---|---|---|---|---|---|
| Intent | Add | 418,547 | 12.75% | 19.31 | 107.47 |
| | Modify | 2,515,332 | 76.64% | 23.53 | 134.14 |
| | Remove | 348,167 | 10.61% | 16.65 | 99.19 |
| Operation | Annotation | 372,974 | 11.36% | 20.24 | 117.69 |
| | Data | 322,790 | 9.83% | 32.33 | 174.08 |
| | Geometry | 657,043 | 20.02% | 25.90 | 147.39 |
| | Structure | 246,587 | 7.51% | 28.38 | 164.97 |
| | Style | 302,560 | 9.22% | 21.31 | 119.78 |
| | Text | 1,380,250 | 42.05% | 17.84 | 103.68 |

Table 9: Intent distribution conditioned on operation.

| Operation | Add | Modify | Remove |
|---|---|---|---|
| Annotation | 41.21% | 27.94% | 30.85% |
| Data | 9.47% | 81.18% | 9.35% |
| Geometry | 10.05% | 81.08% | 8.88% |
| Structure | 23.52% | 50.19% | 26.28% |
| Style | 0.70% | 98.49% | 0.81% |
| Text | 7.84% | 86.56% | 5.61% |

Most inferred atomic edits modify existing content (76.6%), while additions and removals jointly account for 23.4% (Table 8). Text is the most frequent operation (42.1%), followed by geometry (20.0%), annotation (11.4%), data (9.8%), style (9.2%), and structure (7.5%). Description length also varies systematically with edit semantics. Data and structural changes require the longest descriptions, averaging 32.3 and 28.4 words per edit, whereas text edits average 17.8 words. Modifications are longer than additions and removals, consistent with the need to specify both an existing and a desired state. Manual inspection suggests that modifications are somewhat overrepresented, as the VLM occasionally describes additions or removals as replacements (e.g., replacing a label with nothing).

Intent depends strongly on operation (Table 9). Text, geometry, data, and especially style edits predominantly modify existing content (>80%). In contrast, 50.2% of structure and 27.9% of annotation edits modify an element. Structure is balanced across additions and removals (23.5% vs. 26.3%), whereas annotations include more additions (41.2% vs. 30.9%).

Table 10: Intent and Operation distribution across semantic similarity intervals.

| Similarity | Intent | | | Operation | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Add | Modify | Remove | Annotation | Data | Geometry | Structure | Style | Text |
| $[0.95, 0.96)$ | 16.41 | 70.69 | 12.89 | 13.84 | 7.84 | 18.21 | 9.13 | 8.53 | 42.45 |
| $[0.96, 0.97)$ | 14.97 | 72.94 | 12.09 | 12.79 | 8.60 | 19.06 | 8.82 | 8.55 | 42.17 |
| $[0.97, 0.98)$ | 13.08 | 75.81 | 11.10 | 11.49 | 9.76 | 20.21 | 7.90 | 8.73 | 41.91 |
| $[0.98, 0.99)$ | 10.04 | 80.90 | 9.06 | 9.37 | 11.75 | 21.82 | 6.19 | 9.29 | 41.58 |
| $[0.99, 0.9999]$ | 5.16 | 89.77 | 5.07 | 6.70 | 13.27 | 22.35 | 3.39 | 12.34 | 41.95 |

Table 11: Candidate distribution, quality, and edit magnitude across semantic similarity intervals.

| Similarity | Pairs | Share | Retention | Invalid | Identical | Edits/traj. | Words/edit |
|---|---|---|---|---|---|---|---|
| $[0.95, 0.96)$ | 95,641 | 22.22% | 81.91% | 12.09% | 0.04% | 5.29 | 20.86 |
| $[0.96, 0.97)$ | 83,942 | 19.50% | 90.05% | 5.24% | 0.08% | 4.96 | 21.39 |
| $[0.97, 0.98)$ | 79,764 | 18.53% | 94.47% | 1.81% | 0.12% | 4.48 | 22.11 |
| $[0.98, 0.99)$ | 80,295 | 18.65% | 96.55% | 0.47% | 0.28% | 3.82 | 23.23 |
| $[0.99, 0.9999]$ | 90,098 | 20.93% | 92.91% | 0.22% | 4.19% | 2.61 | 25.35 |

Lower-similarity pairs contain more additions and removals, whereas higher similarity pairs contain more modifications. Moreover, highly similar pairs primarily modify existing geometry, data, and style while annotation and structural changes correspond to lower similarity. Text remains stable at approximately 42% across all intervals (Table 10).

The five intervals each contain between 18.5% and 22.2% of candidates in Table 11. Mean edit count decreases monotonically from 5.29 to 2.61 as similarity increases, while description length rises from 20.9 to 25.4 words per edit. Bidirectional retention peaks at 96.6% in $[0.98, 0.99)$. Lower similarities increasingly produce non-plausible edit pairs, whereas the highest interval contains more identical pairs that differ only at the code level (e.g., through refactoring).

Table 12: Source-specific characteristics.

| Source | Pairs | Share | Groups | Retention | Edits/traj. | Words/edit | Intent | Operation |
|---|---|---|---|---|---|---|---|---|
| arXiv | 402,411 | 93.49% | 79,246 | 90.90% | 4.23 | 22.26 | Modify 76.66%, add 12.74%, remove 10.59% | Text 42.34%, geometry 20.02%, annotation 11.29%, data 9.78%, style 9.09%, structure 7.47% |
| GitHub | 21,891 | 5.09% | 3,032 | 87.98% | 4.12 | 21.69 | Modify 76.14%, add 13.09%, remove 10.77% | Text 40.38%, geometry 17.51%, annotation 12.64%, data 11.31%, style 9.49%, structure 8.66% |
| TeX SE | 6,068 | 1.41% | 4,761 | 89.26% | 2.29 | 26.52 | Modify 77.11%, add 11.49%, remove 11.40% | Geometry 36.04%, style 23.26%, text 16.73%, annotation 11.76%, data 6.98%, structure 5.23% |

Table 13: Guidelines for annotating errors in VLM-generated edit instructions.

| Category | Guideline |
|---|---|
| Error | The instruction contains at least one of omission, hallucination, or misinterpretation. |
| Omission | A change is absent or only partially captured by the instruction. |
| Hallucination | A change is specified for which no corresponding source-to-target change exists. |
| Misinterpretation | A change is identified but describes one or more of its properties incorrectly. |
| Attribute | An edited object's identity, appearance, style, text, shape, or other non-numeric property is described incorrectly. |
| Numeric | A numerical value or quantitative change is described incorrectly (e.g., values, counts, dimensions, or magnitudes). |
| Spatial | The spatial relation, position, orientation, direction, or arrangement of edited elements is described incorrectly. |

ArXiv supplies most trajectories and is primarily text-centered, whereas GitHub contains more annotation and data edits. TeX SE provides a distinct form of supervision. Its trajectories contain fewer atomic edits (2.3 versus 4.2) but require the most detailed instructions (26.5 words per edit versus 22). It predominantly involves geometric and stylistic over textual refinements (Table 12).

### A.2.2 Dataset Quality Analysis

Our dataset quality analysis involved one master's student and one PhD student. Both annotators completed the evaluation sheet in Figure 9. The guidelines for completing our evaluation form are

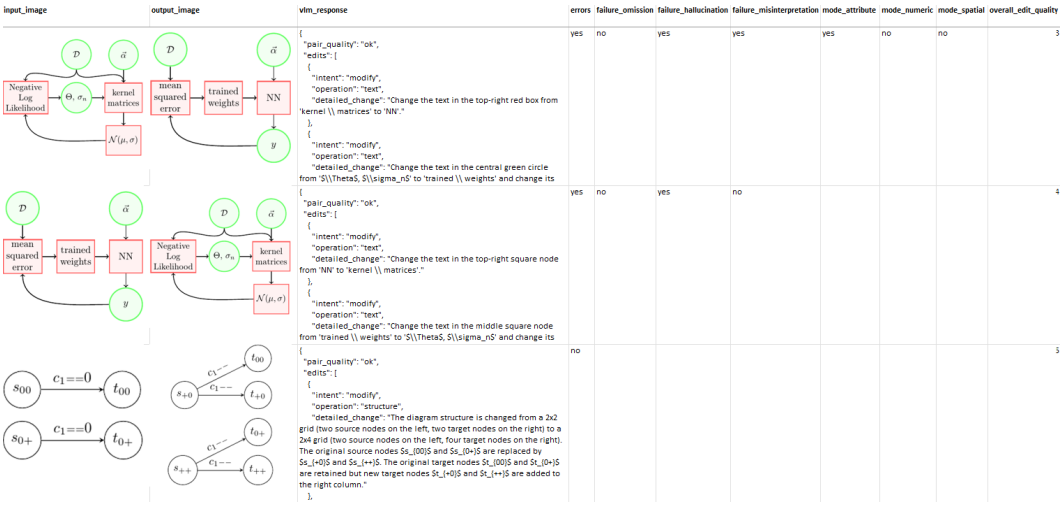



Figure 9: Screenshot of our excel sheet for evaluating the directional VLM-generated edit instructions.

summarized in Table 13.

### A.2.3 DaEdiTikZ-Bench

Six annotators (four master's students, one PhD student, one assistant professor) manually correct all 690 VLM-generated instructions from our benchmark. Similar to the dataset quality anal-

ysis, they were provided with the source image, target image, and the raw VLM response in JSON objects/entries. For omissions, they append another part of the JSON object (with `intent`, `operation`, and `detailed_change`), where the missed change is described. For hallucination, the corresponding part of the JSON object is removed and misinterpretation keeps it but corrects the error. The correction sheet is in Figure 10.

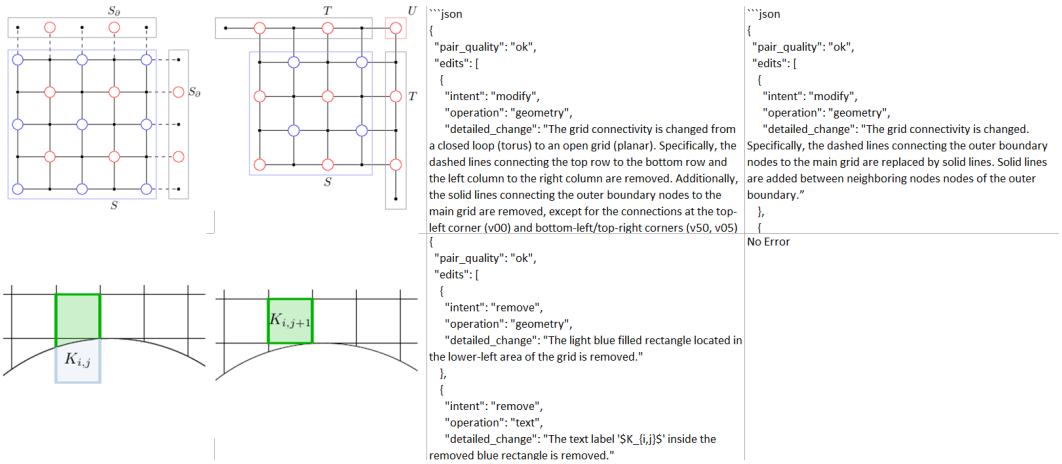



Figure 10: Screenshot of our excel sheet for correcting the directional VLM-generated edit instructions.

## A.3 Method

### A.3.1 Joint Reconstruction and Editing SFT

Figure 11 and 12 present the prompts for joint editing and reconstruction SFT. The editing prompt is used across all training stages and evaluation of all models.

**Editing Prompt**

```
This is an image of a scientific figure.  Reconstruct it in TikZ and apply the
following changes:
{edit_instruction}
Wrap your code using \documentclass[tikz]{standalone}, and include
\begin{document}...\end{document}.  Only output valid LaTeX code with no extra text.
```

**Reconstruction Prompt**

```
This is an image of a scientific figure.  Reconstruct it in TikZ.
Wrap your code using \documentclass[tikz]{standalone}, and include
\begin{document}...\end{document}.  Only output valid LaTeX code with no extra text.
```

### A.3.2 Editing-Specific Rewards

The prompt template for our instruction-following reward $\mathcal{R}_{\text{IF}}$ is shown in Figure 13. As our VLM-as-a-judge backbone, we use Qwen3.6-27B (thinking disabled). It uses greedy decoding (`temperature=0.0` and `top_p=1.0`) and 128 output tokens. Judging is done with vLLM on 1 x Nvidia H100 (94 GB).

**VLM-as-a-Judge ($\mathcal{R}_{\text{IF}}$)**

```
You are a strict visual reward judge for scientific figure editing.
You are given TWO images:
- Image 1 is the source figure before editing.
- Image 2 is the predicted figure after editing.
```

```
You are also given a list of requested atomic edits.
Judge whether Image 2 applies each requested atomic edit relative to Image 1.
For each atomic edit, decide whether the rendered image visibly and fully applies that
edit.
Binary rating rubric:
1 = APPLIED. All essential visible requirements are satisfied in the predicted
figure, the correct target was modified, requested text or mathematics is legible,
and conflicting old content is absent.
0 = NOT APPLIED. The edit is absent, incomplete, incorrect, applied to the wrong
target, contradicted by old content, insufficiently legible, or cannot be reliably
verified.
Important judging rules:
- Score the predicted figure, not the source figure.
- Use the source figure only to identify original objects, positions, labels, shapes,
connections, and content that should be added, removed, or modified.
- Judge only visible instruction faithfulness, not general similarity or visual
beauty.
- Do not give credit for incomplete or ambiguous attempts.
- For an edit with multiple required parts, return 1 only if all essential parts are
visibly satisfied.
- For text and mathematical edits, return 1 only if the requested content is legible.
- If an edit changes X to Y, return 1 only if Y is visible at the correct location and
conflicting X is absent there.
- For removals, return 1 only if the specified original content is absent from its
original location.
- Evaluate every edit independently and preserve the exact order.
- Return exactly one integer per atomic edit.
Return ONLY compact valid JSON in exactly this format:
{
  "ratings": [0, 0, ..., 0]
}
The array length must exactly match the number of atomic edits.
Atomic edits:
{atomic_edits}
Your output JSON:
```

### A.3.3 Multi-Reward Optimization with GDPO

GDPO independently normalizes the advantages induced by $\mathcal{R}_{\text{SSim}}$ and $\mathcal{R}_{\text{IF}}$ across its rollout group before combining them with equal weights. For the policy loss, we adopt the constant-length normalization proposed by Dr.GRPO (Liu et al., 2025) where the summed token-level loss of each rollout is normalized by the fixed maximum completion length $L$ which avoids introducing a response-length-dependent optimization bias for TikZ programs. We further adopt DAPO's Clip-Higher strategy (Yu et al., 2025), using asymmetric clipping with $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$. The relaxed upper bound allows larger probability increases for low-probability exploratory tokens while the lower bound remains unchanged. Rollouts are sampled with `temperature=1.0` and `top_p=0.99`, with a maximum completion length of 2048 tokens. Completions truncated at this limit are excluded from the policy loss. We disable KL regularization ($\beta = 0$).

## A.4 Experiments

### A.4.1 Models

We use separate hyperparameter configurations for models in the 3–4B (Small) and 8–9B (Large) parameter ranges during both SFT and RL. The configurations are summarized in Table 14. Input images are resized to $448 \times 448$. We exclude samples whose TikZ code exceeds 4,000 characters or whose instruction exceeds 2,000 characters. Optimization uses AdamW (Loshchilov & Hutter, 2019). We train with Deepspeed ZeRO-2 (Rajbhandari et al., 2020). All open-source baselines and trained models are evaluated on DaEdiTikZ-Bench with a maximum of 2,048 output tokens, temperature $0.2$, top-$p$ $0.9$, and top-$k$ $50$. Proprietary GPT and Gemini models use their default reasoning settings and a maximum output budget of 10K tokens. For the out-of-distribution evaluation on CharXiv and SPIQA, we use temperature $0.1$ and a 10K-token output budget for both open-source and proprietary models to test extrapolation beyond the open models' training output regime.

Table 14: Training hyperparameters for the small (3–4B) and large (8–9B) models.

| Hyperparameter | SFT | | RL | |
|---|---|---|---|---|
| | Small | Large | Small | Large |
| Training duration (days) | 6 | 13 | 8 | 10 |
| GPUs | 4 x H100 | 4 x H100 | 3 x H100 | 3 x H100 |
| Epochs | 2 | 2 | 1 | 1 |
| Per-device batch size | 10 | 6 | 10 | 6 |
| Gradient accumulation steps | 4 | 7 | 6 | 10 |
| Learning rate | $1 \times 10^{-4}$ | $2 \times 10^{-5}$ | $2 \times 10^{-6}$ | $1 \times 10^{-6}$ |
| Learning-rate scheduler | `cosine` | `cosine` | `constant` | `constant` |
| Weight decay | 0.0 | 0.0 | 0.01 | 0.01 |
| Generations per prompt | – | – | 8 | 8 |

### A.4.2 METRICS

TeX Edit Distance (TED) uses Extended Edit Distance (Kusner et al., 2015) with `TexLexer`. DreamSim (DSim) uses an ensemble of CLIP (Radford et al., 2021), DINO (Caron et al., 2021), and OpenCLIP (`ViT-B/16`). Average tokens (AT) are measured with the `o200k_base` tokenizer. Figure 14 presents our task specific VLM-as-a-Judge metric for Edit Application (EA), Source preservation (SP), and Visual Quality (VQ).

**VLM-as-a-Judge (EA, SP, and VQ)**

```
You are evaluating whether a predicted edited scientific figure correctly follows an
edit instruction.

You are given:
1.  The edit instruction
2.  The original source figure image before editing
3.  The predicted edited figure image
4.  A reference target figure image showing one intended edited result

Important:
- The edit instruction is the primary specification.
- The reference target image is a helpful guide for the intended result, but it may
contain minor artifacts, imperfect alignment, or details not fully described in the
instruction.
- Do not require the prediction to copy harmless imperfections from the reference
target.
- Reward predictions that correctly apply the instruction, preserve unrelated source
content/avoid extra changes, and remain visually clean.

Score the prediction on three criteria from 0 to 10.

Criterion 1:  edit_application_score
How completely and correctly are the requested edits applied?
0 = no requested edits are applied or the prediction is unrelated/unusable
1-2 = almost all requested edits are missing or wrong
3-4 = a few requested edits are attempted, but most are missing/wrong
5-6 = some requested edits are correct, but important edits are missing or inaccurate
7-8 = most requested edits are correct, with minor omissions or inaccuracies
9 = essentially all requested edits are correct, with only tiny issues
10 = all requested edits are applied correctly and completely

Criterion 2:  source_preservation_score
How well does the prediction preserve all source content not required to change, and
avoid adding/removing unrelated elements?
0 = unchanged source content is completely lost, corrupted, replaced, or dominated by
unrelated additions
1-2 = most unchanged content is badly altered, removed, or many unrelated elements are
added
3-4 = many unchanged elements are altered, missing, misplaced, or extra unrelated
elements are present
5-6 = major unchanged structure is preserved, but several details change unnecessarily
or some unrelated elements appear
7-8 = most unchanged content is preserved, with only minor/moderate unrelated changes
9 = nearly all unchanged content is preserved, with only tiny unrelated differences
10 = all unchanged source content is preserved very well and no unrelated elements are
introduced
```

Table 15: Reconstruction performance on the 690 endpoint figures of DaEdiTikZ-Bench.

| Model | TED↓ | DSim↑ | $SP_R$↑ | VQ↑ | Avg↑ | CR↑ | AT↓ |
|---|---|---|---|---|---|---|---|
| GPT-5.6-Sol | 0.798 | 0.777 | **0.802** | 0.828 | 0.652 | 84.0% | 568 |
| GPT-5.5 | 0.791 | 0.628 | 0.642 | 0.684 | 0.541 | 70.0% | 533 |
| Gemini-3.1-Pro | 0.730 | 0.773 | 0.770 | 0.808 | 0.655 | 82.0% | 459 |
| Gemini-3.6-Flash | 0.742 | 0.573 | 0.598 | 0.614 | 0.511 | 62.0% | **340** |
| Qwen3.6-27B | 0.782 | 0.581 | 0.470 | 0.590 | 0.465 | 66.4% | 583 |
| Qwen3.5-27B | 0.769 | 0.673 | 0.561 | 0.700 | 0.541 | 76.8% | 525 |
| Qwen2.5-VL-7B | 0.778 | 0.482 | 0.237 | 0.495 | 0.359 | 60.7% | 516 |
| Qwen2.5-VL-3B | 0.810 | 0.354 | 0.122 | 0.363 | 0.257 | 48.1% | 748 |
| DeTikZify-3B | 0.681 | 0.674 | 0.380 | 0.634 | 0.502 | 76.1% | 661 |
| EdiTikZ-3B | 0.718 | 0.697 | 0.348 | 0.669 | 0.499 | 80.3% | 624 |
| Qwen3-VL-4B | 0.801 | 0.480 | 0.302 | 0.493 | 0.369 | 58.6% | 749 |
| $\text{EdiTikZ-4B}_{\text{Qwen3}}$ | 0.651 | 0.810 | 0.538 | 0.782 | 0.620 | 89.3% | 535 |
| Qwen3-VL-8B | 0.784 | 0.555 | 0.360 | 0.559 | 0.423 | 65.9% | 625 |
| DeTikZify-8B | 0.640 | 0.843 | 0.661 | 0.822 | 0.672 | 91.9% | 510 |
| EdiTikZ-8B | 0.690 | 0.795 | 0.609 | 0.780 | 0.624 | 87.1% | 545 |
| Qwen3.5-4B | 0.826 | 0.352 | 0.220 | 0.338 | 0.271 | 42.6% | 773 |
| EdiTikZ-4B | 0.618 | 0.850 | 0.727 | 0.843 | 0.701 | 91.6% | 512 |
| Qwen3.5-9B | 0.810 | 0.480 | 0.344 | 0.481 | 0.374 | 56.2% | 750 |
| EdiTikZ-9B | **0.590** | **0.894** | 0.795 | **0.892** | **0.748** | **94.5%** | 503 |

```
Criterion 3:  visual_quality_score
How visually clean, legible, and publication-ready is the predicted figure?
0 = unusable rendering, blank image, or severe corruption
1-2 = severe layout/rendering problems, mostly unreadable
3-4 = many visual problems such as clipping, overlap, or unreadable labels
5-6 = usable but visibly flawed or messy
7-8 = mostly clean and legible, with minor/moderate visual issues
9 = very clean, with only tiny visual issues
10 = clean, legible, well-aligned, and publication-quality

Return valid JSON only in exactly this format:
{
   "edit_application_score":  0,
   "edit_application_reasoning":  "",
   "source_preservation_score":  0,
   "source_preservation_reasoning":  "",
   "visual_quality_score":  0,
   "visual_quality_reasoning":  ""
}

Use integer scores from 0 to 10.  Keep each reasoning field to one concise sentence.

Edit instruction:
{edit_instruction}

Your output JSON:
```

## A.5 Results

### A.5.1 Automatic Evaluation

DaEdiTikZ-Bench enables evaluating inverse graphics by treating source and target figures of each editing pair as independent reconstruction examples. We evaluate all 690 figures using the same metrics where applicable, excluding EA and adapting SP (Table 15). Our EdiTikZ-4B and 9B models achieve 0.701 and 0.748 Avg, outperforming all evaluated baselines including GPT-5.6-Sol (0.652), Gemini-3.1-Pro (0.655), and improving substantially over their base models (+0.374/+0.430). Moreover, EdiTikZ-8B performs worse than DeTikZify-8B (0.624 vs. 0.672) suggesting that editing supervision does not improve reconstruction, whereas reconstruction supervision improves editing. We hypothesize that editing requires preserving large parts of the source figure while applying localized changes, so that additional reconstruction examples strengthen the capability needed for editing. Conversely, reconstruction does not require instruction following, and mixing its image-to-TikZ supervision with potentially noisy edit instructions may dilute its objective.

### A.5.2 HUMAN EVALUATION

Five master's students, three PhD students, and one faculty member (5 male, 4 female) participate in the human evaluation. Each annotator receives detailed guidelines and an Excel sheet containing 20 benchmark examples, yielding 180 example-level annotations and 4,320 individual criterion ratings. Each row presents the source figure, edit instruction, and randomly ordered, anonymized outputs from Gemini-3.1-Pro, GPT-5.6-Sol, Qwen3.5-4B, EdiTikZ-4B, EdiTikZ-4B-RL, Qwen3.5-9B, EdiTikZ-9B, and EdiTikZ-9B-RL. Successfully compiled outputs are rated on 1–7 Likert scales for Edit Application (EA), Source Preservation (SP), and Visual Quality (VQ). Non-compilable outputs receive a score of 0. The complete rating criteria are provided below, and representative examples are shown in Figure 15, 16, and 17. Likert scale definitions are shown below:

- **Edit Application (EA)**: 7) All requested edits are applied correctly and completely. 6) Essentially all requested edits are correct, with only tiny issues. 5) Most requested edits are correct, with minor omissions or inaccuracies. 4) Some requested edits are correct, but important edits are missing or inaccurate. 3) A few requested edits are attempted, but most are missing or wrong. 2) Almost all requested edits are missing or wrong. 1) No requested edits are applied, or the prediction is unrelated or unusable.
- **Source Preservation (SP)**: 7) All unchanged source content is preserved very well, and no unrelated elements are introduced. 6) Nearly all unchanged content is preserved, with only tiny unrelated differences. 5) Most unchanged content is preserved, with only minor or moderate unrelated changes. 4) The major unchanged structure is preserved, but several details change unnecessarily or some unrelated elements appear. 3) Many unchanged elements are altered, missing, misplaced, or accompanied by extra unrelated elements. 2) Most unchanged content is badly altered or removed, or many unrelated elements are added. 1) Unchanged source content is completely lost, corrupted, replaced, or dominated by unrelated additions.
- **Visual Quality (VQ)**: 7) Clean, legible, well-aligned, and publication-quality. 6) Very clean, with only tiny visual issues. 5) Mostly clean and legible, with minor or moderate visual issues. 4) Usable but visibly flawed or messy. 3) Many visual problems, such as clipping, overlap, or unreadable labels. 2) Severe layout or rendering problems; mostly unreadable. 1) Unusable rendering, blank image, or severe corruption.

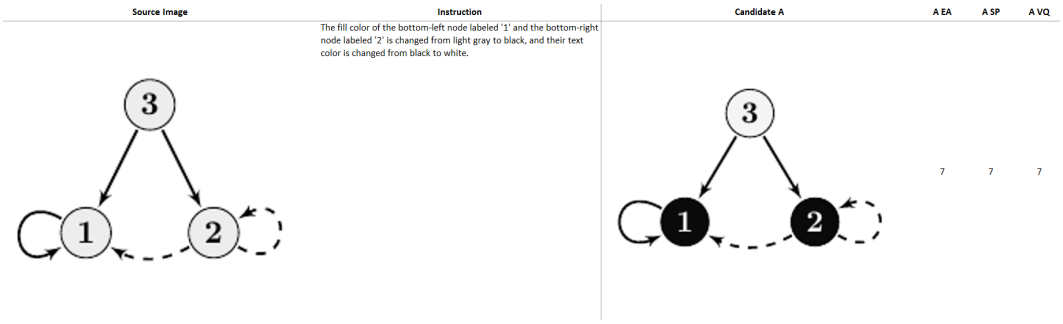



Figure 15: Example with perfect scores for EA, SP, and VQ.

### A.5.3 GENERALIZATION UNDER SEVERE DISTRIBUTION SHIFT

During pilot generation, synthetic instructions frequently collapsed to repetitive edit types, specified only one or two shallow changes, or referred to elements that were not visibly grounded in the input figure. We therefore condition GPT-5.6-Sol on the desired number of atomic edits and the exact numbers of `modify`, `add`, and `remove` intents. The prompt additionally specifies admissible operation types, atomicity and visual-grounding constraints, and nine diverse human-written examples of plausible scientific-figure edits. To avoid a fixed synthetic edit profile, we sample the requested number of edits and intent composition for each figure from the empirical DaEdiTikZ distribution. The complete prompt for generating synthetic edit instructions for the SPIQA and CharXiv analyses is provided in Figure 18.

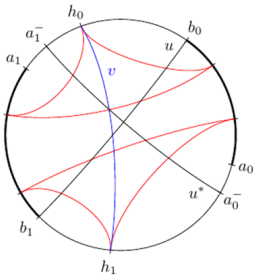
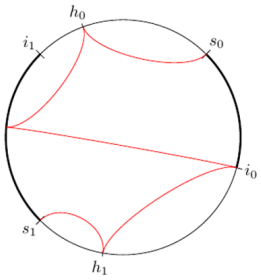
| Source Image | Instruction | Candidate D | D EA | D SP | D VQ |
|---|---|---|---|---|---|
|  | Remove the two black straight lines (chords) that cross the interior of the circle, connecting the top-left to bottom-right and top-right to bottom-left.<br>Remove the blue curved line connecting the top point ($h_0$) to the bottom point ($h_1$).<br>Remove the text labels '$v$', '$u$', and '$u^{\ast}$' located inside the circle.<br>Remove the tick mark and label '$a^-_0$' located on the bottom-right side of the circle.<br>Remove the tick mark and label '$a^-_1$' located on the top-left side of the circle.<br>Change the label '$b_0$' (top-right) to '$s_0$'.<br>Change the label '$a_0$' (bottom-right) to '$i_0$'.<br>Change the label '$b_1$' (bottom-left) to '$s_1$'.<br>Change the label '$a_1$' (top-left) to '$i_1$'. |  | 7 | 4 | 7 |

Figure 16: Example with lower source preservation but high edit application and visual quality.

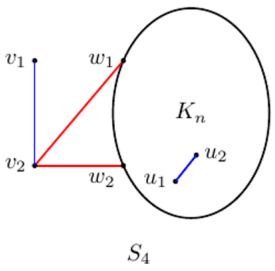
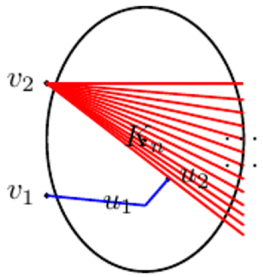
| Source Image | Instruction | Candidate H | H EA | H SP | H VQ |
|---|---|---|---|---|---|
|  | The two red lines originating from the bottom-left vertex (labeled $v_2$ in Image 1) are removed. They are replaced by a fan of multiple red lines originating from the top-left vertex (labeled $v_2$ in Image 2) and another fan of red lines originating from the bottom-left vertex (labeled $v_1$ in Image 2), both pointing towards the right.<br>The labels for the vertices on the left are swapped: the top vertex is now labeled $v_2$ and the bottom vertex is now labeled $v_1$.<br>The labels $w_1$ and $w_2$ and their corresponding points on the ellipse boundary are removed.<br>The label $S_4$ at the bottom of the figure is removed.<br>Two ellipsis symbols ($\dots$) are added to the right of the red line fans to indicate continuation. |  | 3 | 4 | 2 |

Figure 17: Example with lower scores for all three.

**OOD Edit-Instruction Generation**

```
You are creating image-editing instructions for a scientific-figure editing evaluation
benchmark.

Inspect the supplied scientific figure carefully.

Generate exactly {num_edits} atomic edit instructions.

The instructions must contain exactly:
- {num_modify} MODIFY edits
- {num_add} ADD edits
- {num_remove} REMOVE edits

INTENT DEFINITIONS

MODIFY:
Modify an existing visible element while preserving or replacing its role, appearance,
value, position, structure, or content.

ADD:
Introduce a new element that is not currently present in the figure.

REMOVE:
Delete an existing visible element without replacing it as part of the same atomic
intent.

ATOMICITY
Each edit object must represent exactly one atomic edit.
Do not combine multiple independent edits into one instruction merely to satisfy the
requested intent counts.
A replacement of one visible element with another may be treated as a MODIFY intent
when it represents a single coherent replacement.

VALIDITY REQUIREMENTS
- Every MODIFY or REMOVE instruction must refer to an element that is clearly visible
and unambiguously identifiable in the supplied image.
- Do not invent existing labels, values, curves, nodes, arrows, boxes, legends,
colors, equations, annotations, or other elements.
- Never refer to an existing element unless you can visually verify that it is
```

present.
- For ADD instructions, describe the new element's position or connection relative to clearly visible existing elements.
- All requested edits must be visually executable from the supplied image alone.
- Do not require information from the source paper, underlying numerical data, hidden metadata, or outside knowledge.
- The edits must be mutually compatible.
- Do not ask to modify, move, recolor, or relabel an element that another requested edit removes.
- Do not produce redundant edits that accomplish essentially the same modification twice.

DIVERSITY REQUIREMENTS
Prefer meaningful textual, annotation-level, geometric, structural, stylistic, semantic, or data-level modifications.

Possible edit targets include, but are not limited to:
- labels, equations, symbols, and numerical values
- axes, ticks, legends, titles, and plotted data
- curves, bars, markers, arrows, and paths
- nodes, blocks, connections, and graph topology
- scientific diagrams, architectures, circuits, and geometry
- annotations, boxes, braces, loops, regions, and grids
- positions, dimensions, orientations, and spatial relationships
- colors, line styles, marker styles, and fills when visually meaningful

Do not force any particular edit type if it does not naturally apply to the shown figure.

When multiple edits are requested, prefer edits that affect different meaningful aspects of the figure rather than repeatedly modifying nearly identical elements.

SPECIFICITY
Refer to visible elements using enough identifying information to make the edit unambiguous.

Good references include:
- the upper-right node
- the dashed rectangle around the encoder
- the blue curve labeled 'Method A'
- the y-axis tick at 0.5
- the arrow connecting the first and second blocks

Avoid vague references such as 'the line', 'the box', 'the node', or 'the label' when multiple such elements exist.

STYLE EXAMPLES FROM REAL FIGURE-EDIT REQUESTS
These examples demonstrate the desired specificity and variety only.
Do not copy their content, entities, values, or sentence structures unless they naturally apply to the supplied figure.
1. Change the input label from '$\hat{\boldsymbol{u}}^{\mathrm{sym}}$' to '$\boldsymbol{u}$'.
2. Change the fill color of the 'Policy' block from white to light red.
3. Add a new block labeled '$\mathbb{R}2\mathbb{C}$' at the end of the chain, after the Policy block.
4. Replace the rectangular block labeled 'CLK' and 'MEM' with a D-shaped AND gate symbol.
5. Adjust the y-axis tick marks to 0, 5, and 10.
6. Replace the current 3D surface with a Rosenbrock-style surface containing a long curved valley.
7. Move the rectangular path from the right side of the y-axis to the left side, with its vertical segment at $x = -1$ and its horizontal segments at $y = 1$ and $y = -1$.
8. Add a blue curly brace annotation to the right of the legend spanning the top two entries, accompanied by the label 'Grouped'.
9. Remove the Greek letter labels '$\alpha$', '$\beta$', '$\gamma$', and '$\delta$' located above the first four nodes of the top horizontal line.

OUTPUT FORMAT
Return ONLY valid JSON.
Do not use Markdown code fences.
Do not include commentary before or after the JSON.
Do not explain your reasoning.

The JSON must have exactly this structure:

```
{
  "edits": [
    {
      "intent": "modify|add|remove",
      "instruction": "..."
    }
```

Table 16: Performance of our EdiTikZ models on SPIQA and CharXiv against baselines.

| | SPIQA | | | | | CharXiv | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **EA↑** | **SP↑** | **VQ↑** | **CR↑** | **AT↓** | **EA↑** | **SP↑** | **VQ↑** | **CR↑** | **AT↓** |
| GPT-5.6-Sol | **0.634** | **0.598** | **0.659** | 80.6% | **1352** | **0.524** | **0.488** | 0.552 | 68.3% | **1594** |
| Qwen3.5-27B | 0.234 | 0.191 | 0.277 | 57.9% | 1787 | 0.206 | 0.164 | 0.247 | 42.6% | 2351 |
| Qwen3.5-4B | 0.063 | 0.037 | 0.083 | 24.2% | 2872 | 0.076 | 0.051 | 0.127 | 26.8% | 3164 |
| EdiTikZ-4B | 0.158 | 0.112 | 0.309 | 63.2% | 2547 | 0.154 | 0.099 | 0.364 | 65.8% | 2550 |
| EdiTikZ-4B-RL | 0.243 | 0.174 | 0.445 | 85.8% | 1683 | 0.210 | 0.145 | 0.439 | 81.1% | 1917 |
| Qwen3.5-9B | 0.088 | 0.066 | 0.152 | 34.2% | 1888 | 0.140 | 0.092 | 0.196 | 37.9% | 2428 |
| EdiTikZ-9B | 0.319 | 0.238 | 0.428 | 74.2% | 2166 | 0.254 | 0.185 | 0.421 | 71.1% | 2883 |
| EdiTikZ-9B-RL | 0.520 | 0.466 | 0.622 | **87.6%** | 1981 | 0.399 | 0.323 | **0.559** | **85.0%** | 2541 |

```
    ]
}

Requirements:
- "intent" must be exactly one of:  "modify", "add", "remove".
- "instruction" must contain exactly one complete atomic edit instruction.
- Every instruction must be one sentence.
- The "edits" array must contain exactly {num_edits} objects.
- There must be exactly {num_modify} objects with intent "modify".
- There must be exactly {num_add} objects with intent "add".
- There must be exactly {num_remove} objects with intent "remove".

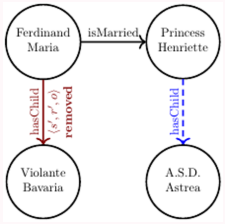
Now generate the edit instructions for the supplied scientific figure.
```

For the OOD evaluation, we adapt the prompt in Figure 14 to a reference free setting by omitting the target figure and all target-dependent instructions, and explicitly instructing the judge to evaluate the prediction from the source figure and edit instruction alone. The GPT-5.5 judge and decoding configuration remain unchanged. Table 16 provides the full results of our stress-tests on SPIQA and CharXiv. EdiTikZ generations are approximately $4$–$5\times$ longer than on DaEdiTikZ-Bench and exhibit substantially lower scores and compilation rates. GPT-5.6-Sol achieves the strongest overall performance on both datasets, while EdiTikZ-9B-RL remains competitive with higher compilation rates and slightly higher VQ on CharXiv. The improvement from SFT to RL is substantially larger across the OOD metrics than on DaEdiTikZ-Bench. Despite RL using only a small in-domain subset of DaEdiTikZ, its benefits transfer strongly to substantially more complex figures outside the training distribution. Across models, SP degrades most strongly, indicating that preserving unchanged content becomes particularly challenging as figure complexity increases.

### A.5.4 Examples

## Example with TikZ Code

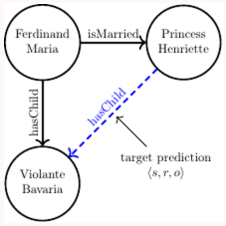
**Source**




**Edit Instruction**

The node labeled 'A.S.D. Astrea' and the blue dashed arrow connecting 'Princess Henriette' to it are removed from the diagram. The arrow connecting 'Ferdinand Maria' to 'Violante Bavaria' is changed from a solid red line to a solid black line. The text annotation '$\langle s', r', o\rangle$ removed' located below the arrow between 'Ferdinand Maria' and 'Violante Bavaria' is removed. A new blue dashed arrow labeled 'hasChild' is added, connecting 'Princess Henriette' to 'Violante Bavaria'. A text block reading 'target prediction $\langle s, r, o\rangle$' is added to the bottom right, with an arrow pointing from it to the new blue dashed arrow.

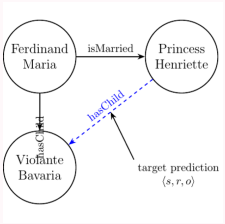
**Ground Truth**




```
\begin{tikzpicture}[scale=0.6, every node/.style={scale=0.6}]
\node[draw,circle,thick,inner sep=0,minimum size=2.1cm] (r_fm)
        at (12,-5) {\begin{tabular}{c} Ferdinand \\ Maria \end{
        tabular}};
\node[draw,circle,thick,inner sep=0,minimum size=2.1cm] (r_ph)
        at (16,-5) {\begin{tabular}{c} Princess\\ Henriette\end
        {tabular}};
\node[draw,circle,thick,inner sep=0,minimum size=2.1cm] (r_vb)
        at (12,-9) {\begin{tabular}{c} Violante\\ Bavaria\end{
        tabular}};
\node[draw,circle,thick,inner sep=0,minimum size=2.1cm] (r_ny)
        at (16,-9) {\begin{tabular}{c} A.S.D.\\Astrea\end{
        tabular}};
\draw[->,thick] (r_fm) -- (r_ph) node[pos=0.5, sloped,align=
        center, above]{isMarried};
\draw[->,thick,red!50!black] (r_fm) -- (r_vb) node[pos=0.5,
        sloped,align=center, above]{hasChild} node[pos=0.5,
        sloped,align=center, below]{$\langle s',r',o\rangle$\\\
        textbf{removed}};
\draw[->,thick,blue,densely dashed] (r_ph) -- (r_ny) node[pos
        =0.5, sloped,align=center, above] {hasChild};
\end{tikzpicture}
```

```
\begin{tikzpicture}[scale=0.6, every node/.style={scale=0.6}]
\node[draw,circle,thick,inner sep=0,minimum size=2.1cm] (o_fm)
        at (0,0) {\begin{tabular}{c} Ferdinand \\ Maria \end{
        tabular}};
\node[draw,circle,thick,inner sep=0,minimum size=2.1cm] (o_ph)
        at (4,0) {\begin{tabular}{c} Princess\\ Henriette\end{
        tabular}};
\node[draw,circle,thick,inner sep=0,minimum size=2.1cm] (o_vb)
        at (0,-4) {\begin{tabular}{c} Violante\\ Bavaria\end{
        tabular}};
\draw[->,thick] (o_fm) -- (o_ph) node[pos=0.5, sloped,align=
        center, above]{isMarried};
\draw[->,thick] (o_fm) -- (o_vb) node[pos=0.5, sloped,align=
        center,rotate=180, above]{hasChild};
\draw[->,thick,blue,densely dashed] (o_ph) -- (o_vb) node[pos
        =0.5, sloped,align=center, above] {hasChild};
\node (a) at (3.5,-3.5) {\begin{tabular}{c} target prediction\\
        $\langle s,r,o\rangle$\end{tabular} };
\draw[->] (a) -- (2.1,-2.1);
\end{tikzpicture}
```

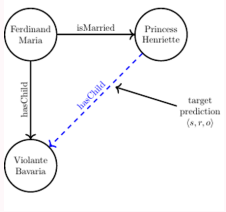
**GPT-5.6-Sol**




| | |
|---|---|
| **TED** | 0.716 |
| **DSim** | 0.922 |
| **EA** | 1.000 |
| **SP** | 1.000 |
| **VQ** | 0.800 |
| **AT** | 334 |

```
\begin{tikzpicture}[>=Stealth,person/.style={circle,draw,thick,minimum
        size=2.3cm,align=center,font=\large}]
\node[person] (ferdinand) at (0,3.5) {Ferdinand\\Maria};
\node[person] (princess) at (4.5,3.5) {Princess\\Henriette};
\node[person] (violante) at (0,0) {Violante\\Bavaria};
\draw[->,thick](ferdinand.east) -- node[above]{isMarried} (princess.
        west);
\draw[->,thick](ferdinand.south) -- node[left,rotate=90]{hasChild} (
        violante.north);
\draw[->,thick,blue,dashed](princess.218) -- node[midway,above,sloped,
        text=blue]{hasChild}coordinate[midway](prediction)(violante
        .38);
\node[align=center] (target) at (4.4,-0.25){target prediction\\$\
        langle s,r,o\rangle$};
\draw[->,thick] (target.north west) -- (prediction);
\end{tikzpicture}
```

**EdiTikZ-9B-RL**



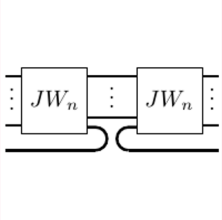

| | |
|---|---|
| **TED** | 0.690 |
| **DSim** | 0.918 |
| **EA** | 1.000 |
| **SP** | 1.000 |
| **VQ** | 1.000 |
| **AT** | 335 |

```
\begin{tikzpicture}[scale=0.6, transform shape]
\node[circle, draw, thick, minimum size=2cm, align=center] (A) at
        (0,0) {Ferdinand\\Maria};
\node[circle, draw, thick, minimum size=2cm, align=center] (B) at
        (5,0) {Princess\\Henriette};
\node[circle, draw, thick, minimum size=2cm, align=center] (C) at
        (0,-5) {Violante\\Bavaria};
\draw[->, thick] (A) -- (B) node[midway, above] {isMarried};
\draw[->, thick] (A) -- (C) node[midway, sloped, above] {hasChild};
\draw[->, thick, blue, dashed] (B) -- (C) node[midway, sloped, above]
        {hasChild};
\node[align=center] (D) at (6.5,-3) {target\\prediction\\$\langle s,r,
        o\rangle$};
\draw[->, thick] (D) -- (3.25,-2);
\end{tikzpicture}
```

Figure 19: TikZ programs and rendered figures for a representative scientific-figure edit. Models receive the source figure and edit instruction and generate the edited TikZ program. Per-example TED, DSim, EA, SP, VQ, and AT are shown beside each prediction.

Example with TikZ Code

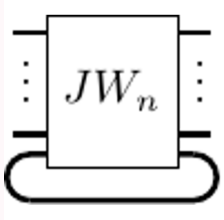
**Source**




**Edit Instruction**

Remove the right-hand $JW_n$ box, its external input/output lines, and the far-right vertical ellipsis. Remove the two horizontal lines. Move the loop from the gap between the boxes to a position beneath the remaining box, where it connects around the lower-left and lower-right sides. Increase the height of the remaining box upward from 2.0 to 2.3 units; the loop remains below the box rather than being contained within its vertical span. Reposition the two left input lines and the left and right vertical ellipses to suit the taller single-box layout.

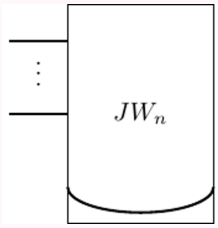
**Ground Truth**




```
\begin{tikzpicture}[scale=0.5]
\draw (-2.75,0) rectangle (-0.75,2);
\node at (-1.75, 1) {$JW_n$};
\draw (0.75,0) rectangle (2.75,2);
\node at (1.75, 1) {$JW_n$};
\draw[very thick] (-3.25, -0.5) -- (-0.5, -0.5);
\draw[very thick] (3.25, -0.5) -- (0.5, -0.5);
\draw[very thick] (-3.25, 0.25) -- (-2.75, 0.25);
\draw[very thick] (-3.25, 1.75) -- (-2.75, 1.75);
\draw[very thick] (3.25, 0.25) -- (2.75, 0.25);
\draw[very thick] (3.25, 1.75) -- (2.75, 1.75);
\draw[very thick] (-0.75, 0.5) -- (0.75, 0.5);
\draw[very thick] (-0.75, 1.75) -- (0.75, 1.75);
\node at (-3.05, 1.25) {$\vdots$};
\node at (0, 1.3) {$\vdots$};
\node at (3.05, 1.25) {$\vdots$};
\draw [very thick] (-0.5, -0.5) arc (-90:90:0.35) -- (-0.75,
      0.2);
\draw [very thick] (0.5, -0.5) arc (270:90:0.35) -- (0.75, 0.2)
      ;
\end{tikzpicture}
```

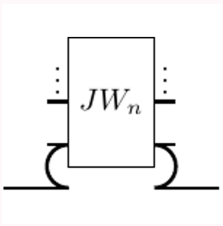
```
\begin{tikzpicture}[scale=0.5]
\draw[very thick] (-3.25, 0.5) -- (-0.25, 0.5);
\draw[very thick] (-3.25, 2.05) -- (-0.25, 2.05);
\node at (-3.05, 1.5) {$\vdots$};
\node at (-0.4, 1.5) {$\vdots$};
\draw [very thick] (-3., -0.5) arc (270:90:0.35) -- (-0.5, 0.2)
      ;
\draw [very thick] (-3., -0.5) -- (-0.5, -0.5) arc
      (-90:90:0.35);
\draw[fill=white] (-2.75,0) rectangle (-0.75,2.3);
\node at (-1.75, 1.15) {$JW_n$};
\end{tikzpicture}
```

**Gemini-3.1-Pro**




| | |
|---|---|
| **TED** | 0.680 |
| **DSim** | 0.758 |
| **EA** | 0.500 |
| **SP** | 0.600 |
| **VQ** | 0.700 |
| **AT** | 165 |

**EdiTikZ-9B-RL**




| | |
|---|---|
| **TED** | 0.659 |
| **DSim** | 0.898 |
| **EA** | 0.500 |
| **SP** | 0.900 |
| **VQ** | 0.800 |
| **AT** | 323 |

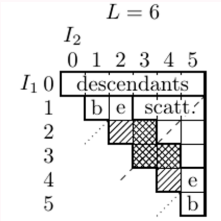
```
\begin{tikzpicture}
\draw (0,0) rectangle (2,3);
\node at (1,1.5) {$JW_n$};
\draw[line width=1pt] (-0.8, 2.5) -- (0, 2.5);
\draw[line width=1pt] (-0.8, 1.5) -- (0, 1.5);
\node at (-0.4, 2.15) {$\vdots$};
\draw[line width=1pt] (0, 0.5) arc (180:360:1 and 0.35);
\end{tikzpicture}
```

```
\begin{tikzpicture}[scale=0.5]
\draw (0,0) rectangle (2,3);
\node at (1,1.5) {$JW_n$};
\draw[very thick] (2,0.5) -- (2.5,0.5);
\draw[very thick] (2,1.5) -- (2.5,1.5);
\draw[very thick] (-0.5,0.5) -- (0,0.5);
\draw[very thick] (-0.5,1.5) -- (0,1.5);
\node at (-0.25,2.2) {$\vdots$};
\node at (2.25,2.2) {$\vdots$};
\draw[very thick] (2,0.5) arc (90:-90:0.5);
\draw[very thick] (0,0.5) arc (90:270:0.5);
\draw[very thick] (2,-.5) -- (3.5,-.5);
\draw[very thick] (0,-.5) -- (-1.5,-.5);
\end{tikzpicture}
```

Figure 20: TikZ programs and rendered figures for a representative scientific-figure edit. Models receive the source figure and edit instruction and generate the edited TikZ program. Per-example TED, DSim, EA, SP, VQ, and AT are shown beside each prediction.

## Example with TikZ Code

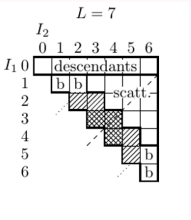
**Source**




**Edit Instruction**

The grid structure is expanded from a 6x6 matrix to a 7x7 matrix, adding a new column (index 6) and a new row (index 6) to the axes and the grid lines. The text label 'e' located at grid position (2, 1) is changed to 'b'." The fill pattern of the cell at grid position (3, 2) is changed from crosshatch to diagonal lines (north east lines). The fill pattern of the cell at grid position (4, 4) is changed from diagonal lines (north east lines) to crosshatch. A new filled cell with diagonal lines (north east lines) pattern is added at grid position (5, 4). A new filled cell with diagonal lines (north east lines) pattern is added at grid position (5, 5). The text label 'e' located at grid position (5, 4) is changed to 'b'. A new text label 'b' is added at grid position (6, 5). The dashed line indicating the boundary of the 'scatt.' region is extended to accommodate the larger grid size, now ending at the bottom right corner of the 7x7 grid.

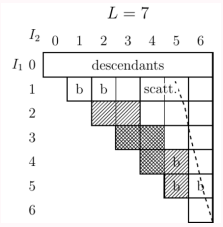
**Ground Truth**




```
\begin{tikzpicture}[scale=.39]
\node at (3,2.5) {$L=6$};
\node at (.5,1.5) {$I_2$}; \foreach \x in {0,...,5} \node at (\
        x+.5,.5) {$\x$};
\node at (-1.3,-.5) {$I_1$}; \foreach \y in {0,...,5} \node at
        (-.5,-\y-.5) {$\y$};
\draw[very thick](0,0) -| (6,-1) -| cycle (1,-1) |- (2,-2) |-
        (3,-3) |- (4,-4) |- (5,-5) |- (6,-6) -- (6,-1) (3,-1)
        |- (4,-2) |- (5,-3) |- (6,-4);
\foreach \x in {1,...,5} \draw[ultra thin] (\x,0) |- (6,-\x);
\draw[dashed] (2.5,-4.5) -- (6,-1) -- (5.5,-1.5);
\foreach \x in {1,4} \draw[dotted] (\x,-\x-2) -- (\x+1,-\x-1);
\node at (3,-.5) [inner sep=.5pt,fill=white] {descendants};
\node at (4.5,-1.5) [inner sep=.5pt,fill=white] {scatt.$\!$};
\node at (1.5,-1.5) {b};
\node at (2.5,-1.5) {e\vphantom{t}};
\fill[pattern=north east lines] (2,-2) rectangle (3,-3);
\fill[pattern=crosshatch] (3,-2) rectangle (4,-3);
\fill[pattern=crosshatch] (3,-3) rectangle (4,-4);
\fill[pattern=crosshatch] (4,-3) rectangle (5,-4);
\fill[pattern=north east lines] (4,-4) rectangle (5,-5);
\node at (5.5,-4.5) {e\vphantom{t}};
\node at (5.5,-5.5) {b};
\node at (0,-9) {};
\end{tikzpicture}
```

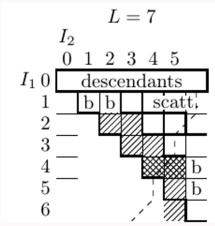
```
\begin{tikzpicture}[scale=.39]
\node at (3.5,2.5) {$L=7$};
\node at (.5,1.5) {$I_2$}; \foreach \x in {0,...,6} \node at (\
        x+.5,.5) {$\x$};
\node at (-1.3,-.5) {$I_1$}; \foreach \y in {0,...,6} \node at
        (-.5,-\y-.5) {$\y$};
\draw[very thick](0,0) -| (7,-1) -| cycle (1,-1) |- (2,-2) |-
        (3,-3) |- (4,-4) |- (5,-5) |- (6,-6) |- (7,-7) --
        (7,-1) (3,-1) |- (4,-2) |- (5,-3) |- (6,-4) |- (7,-5);
\foreach \x in {1,...,6} \draw[ultra thin] (\x,0) |- (7,-\x);
\draw[dashed] (3,-5) -- (7,-1);
\draw[dotted] (1.25,-3.25) -- (2,-2.5) (4.75,-6.75) -- (5.5,-6)
        ;
\node at (3.5,-.5) [inner sep=.5pt,fill=white] {descendants};
\node at (5.5,-2) [inner sep=.5pt,fill=white] {scatt.$\!$};
\node at (1.5,-1.5) {b};
\node at (2.5,-1.5) {b};
\fill[pattern=north east lines] (2,-2) rectangle (3,-3);
\fill[pattern=north east lines] (3,-2) rectangle (4,-3);
\fill[pattern=crosshatch] (3,-3) rectangle (4,-4);
\fill[pattern=crosshatch] (4,-3) rectangle (5,-4);
\fill[pattern=crosshatch] (4,-4) rectangle (5,-5);
\fill[pattern=north east lines] (5,-4) rectangle (6,-5);
\fill[pattern=north east lines] (5,-5) rectangle (6,-6);
\node at (6.5,-5.5) {b};
\node at (6.5,-6.5) {b};
\node at (0,-9) {};
\end{tikzpicture}
```

**GPT-5.6-Sol**




| | |
|---|---|
| **TED** | 0.647 |
| **DSim** | 0.867 |
| **EA** | 0.500 |
| **SP** | 0.600 |
| **VQ** | 0.900 |
| **AT** | 671 |

**EdiTikZ-4B-RL**




| | |
|---|---|
| **TED** | 0.804 |
| **DSim** | 0.938 |
| **EA** | 0.200 |
| **SP** | 0.200 |
| **VQ** | 0.400 |
| **AT** | 1209 |

```
\begin{tikzpicture}[x=0.72cm,y=-0.72cm,font=\\large]
\\node[font=\\Large] at (3.5,-1.65) {$L=7$};\n\\node at (-0.35,-0.72)
        {$I_2$};\n\\node at (-1.05,0.5) {$I_1$};\n\n\\foreach \\x in
        {0,...,6}\n \\node at (\\x+0.5,-0.48) (\\x);\n\\foreach \\y in
         {0,...,6}\n \\node at (-0.45,\\y+0.5) (\\y);\n\n\\fill[
        pattern-north east lines] (2,2) rectangle (3,3);\n\\fill[
        pattern-north east lines] (3,2) rectangle (4,3);\n\n\\fill[
        pattern-north east lines] (3,3) rectangle (4,4);\n\\fill[
        pattern-north west lines] (3,3) rectangle (4,4);\n\n\\fill[
        pattern-north east lines] (4,3) rectangle (5,4);\n\\fill[
        pattern-north west lines] (4,3) rectangle (5,4);\n\n\\fill[
        pattern-north east lines] (4,4) rectangle (5,5);\n\\fill[
        pattern-north west lines] (4,4) rectangle (5,5);\n\n\\fill[
        pattern-north east lines] (5,4) rectangle (6,5);\n\\fill[
        pattern-north east lines] (5,5) rectangle (6,6);\n\n\\draw[
        line width=1pt] (0,0) rectangle (7,1);\n\\foreach \\y in
        {1,...,6}\n \\draw[line width=1pt] (\\y,\\y) rectangle (7,\\y
        +1);\n\n\\foreach \\y in {1,...,6}{\n \\foreach \\x in {\\y
        ,...,6}{\n \\draw[line width=0.75pt] (\\x,\\y)--(\\x,\\y+1);\n
         }\n}\n\n\\draw[dashed,line width=0.9pt]\n (5.45,1.18) --
        (5.85,2.25) -- (6.05,3.35)\n -- (6.35,4.55) -- (6.62,5.75) --
        (7,7);\n\n\\node at (3.5,0.5) {descendants};\n\\node at
        (1.5,1.5) {b};\n\\node at (2.5,1.5) {b};\n\\node[fill=white,
        inner sep=1pt] at (4.85,1.5) {scatt.};\n\n\\node at (5.5,4.5)
        {b};\n\\node at (5.5,5.5) {b};\n\\node at (6.5,5.5) {b};
\end{tikzpicture}
```

```
\begin{tikzpicture}[scale=0.4]
\\node at (3.5,9) {$L=7$};\n\\draw[very thick] (0,6) rectangle (7,7);\
        n\\node at (3.5,6.5) {descendants};\n\\draw[thick] (0,6)
        rectangle (7,7);\n\\draw (1,5) rectangle (2,6);\n\\draw (2,5)
        rectangle (3,6);\n\\draw (3,5) rectangle (4,6);\n\\draw (4,5)
        rectangle (5,6);\n\\draw (5,5) rectangle (6,6);\n\\draw (6,5)
        rectangle (7,6);\n\\draw (2,4) rectangle (3,5);\n\\draw (3,4)
        rectangle (4,5);\n\\draw (4,4) rectangle (5,5);\n\\draw (5,4)
        rectangle (6,5);\n\\draw (6,4) rectangle (7,5);\n\\draw (3,3)
        rectangle (4,4);\n\\draw (4,3) rectangle (5,4);\n\\draw (5,3)
        rectangle (6,4);\n\\draw (6,3) rectangle (7,4);\n\\draw (4,2)
        rectangle (5,3);\n\\draw (5,2) rectangle (6,3);\n\\draw (6,2)
        rectangle (7,3);\n\\draw (5,1) rectangle (6,2);\n\\draw (6,1)
        rectangle (7,2);\n\\draw (6,0) rectangle (7,1);\n\\node at
        (0.5,7.5) {$I_2$};\n\\node at (-1,6.5) {$I_1$};\n\\node at
        (0.5,7.5) {$0$};\n\\node at (1.5,7.5) {$1$};\n\\node at
        (2.5,7.5) {$2$};\n\\node at (3.5,7.5) {$3$};\n\\node at
        (4.5,7.5) {$4$};\n\\node at (5.5,7.5) {$5$};\n\\node at
        (6.5,7.5) {$6$};\n\\node at (-0.5,6.5) {$0$};\n\\node at
        (-0.5,5.5) {$1$};\n\\node at (-0.5,4.5) {$2$};\n\\node at
        (-0.5,3.5) {$3$};\n\\node at (-0.5,2.5) {$4$};\n\\node at
        (-0.5,1.5) {$5$};\n\\node at (-0.5,0.5) {$6$};\n\\draw[dashed]
         (1.5,6) -- (1.5,5);\n\\draw[dashed] (2.5,6) -- (2.5,4);\n\\
        draw[dashed] (3.5,6) -- (3.5,3);\n\\draw[dashed] (4.5,6) --
        (4.5,2);\n\\draw[dashed] (5.5,6) -- (5.5,1);\n\\draw[dashed]
        (6.5,6) -- (6.5,0);\n\\draw[dotted] (1.5,5) -- (0.5,4);\n\\
        draw[dotted] (6.5,0) -- (5.5,-1);\n\\draw[dashed] (6.5,5) --
        (2.5,1);\n\\node at (1.5,5.5) {b};\n\\node at (2.5,5.5) {b};\n
        \\node at (4.5,5.5) {scatt.};\n\\node at (3.5,4.5) {\\tiny$\\
        nwarrow$};\n\\node at (5.5,4.5) {};\n\\node at (4.5,3.5) {};\n
        \\node at (6.5,3.5) {};\n\\node at (5.5,2.5) {b};\n\\node at
        (6.5,1.5) {b};\n\\draw[very thick, pattern-north east lines]
        (2,4) rectangle (3,5);\n\\draw[very thick, pattern-north east
        lines] (3,3) rectangle (4,4);\n\\draw[very thick, pattern=
        north east lines] (5,4) rectangle (6,5);\n\\draw[very thick,
        pattern=north east lines] (5,2) rectangle (6,3);\n\\draw[very
        thick, pattern=north east lines] (5,1) rectangle (6,2);\n\\
        draw[very thick, pattern=north east lines] (6,1) rectangle
        (7,2);\n\\draw[very thick, pattern=north east lines] (6,0)
        rectangle (7,1);\n\\draw[very thick, pattern=crosshatch] (4,3)
         rectangle (5,4);\n\\draw[very thick, pattern=crosshatch]
         (4,2) rectangle (5,3);
\end{tikzpicture}
```

Figure 21: TikZ programs and rendered figures for a representative scientific-figure edit. Models receive the source figure and edit instruction and generate the edited TikZ program. Per-example TED, DSim, EA, SP, VQ, and AT are shown beside each prediction.

Table 17: Scientific figure edits by GPT-5.6-Sol, Qwen3.5-9B, and EdiTikZ-9B-RL. Models receive the source image and VLM-generated edit instruction. Human annotations score edit application (E), source preservation (P), and visual quality (Q). Overall quality: ■ very good, ■ good, ■ bad, ■ very bad.

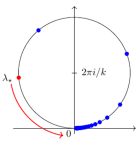
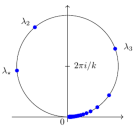
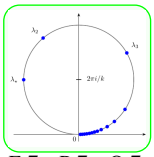
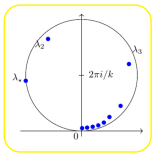
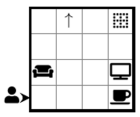
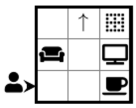
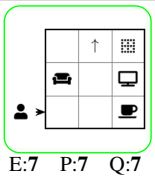
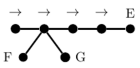
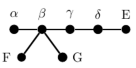
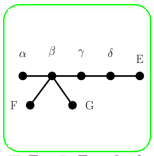
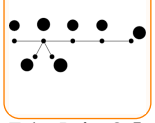
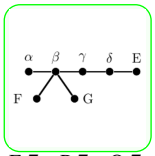
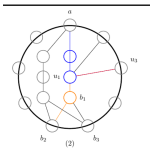
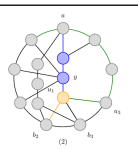
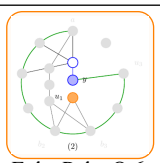
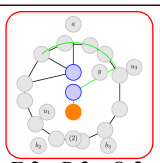
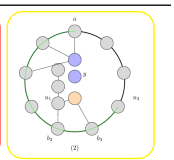
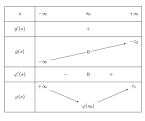
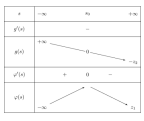
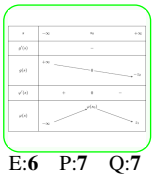
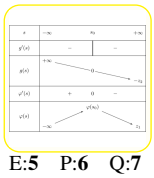
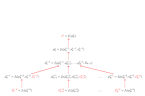
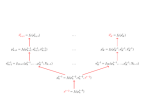
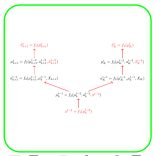
| Source | Edit Instruction | Ground Truth | GPT-5.6-Sol | Qwen3.5-9B | EdiTikZ-9B-RL |
|---|---|---|---|---|---|
|  | The red curved arrow originating from the left side of the circle and pointing towards the origin is removed. The single red data point located on the left side of the circle (near the label $\lambda_*$) is changed to blue. A label $\lambda_2$ is added in the upper-left quadrant, positioned near the corresponding blue data point. A label $\lambda_3$ is added in the upper-right quadrant, positioned near the corresponding blue data point. |  |  E:7 P:7 Q:7 | |  E:7 P:5 Q:5 |
|  | The grid structure is reduced from a 4x4 layout to a 3x3 layout. This involves removing the second row and the third column of cells, effectively shrinking the main container area. |  | | |  E:7 P:7 Q:7 |
|  | The four right-pointing arrow symbols ($\rightarrow$) located above the first four nodes on the top row are replaced with the Greek letters $\alpha$, $\beta$, $\gamma$, and $\delta$ respectively, from left to right. |  |  E:7 P:7 Q:6 |  E:1 P:4 Q:5 |  E:7 P:7 Q:7 |
|  | The text in the bottom-left cylinder node (originally 'Predictive world model') is changed to 'Specifically incorrect predictive world model'. The color of the bottom-left cylinder node is changed from light blue to green. The text in the bottom-right cylinder node (originally 'Predictive world model') is changed to 'Correct predictive world model'. A new rectangular node labeled 'Model Editing' is added to the right of the 'Correct predictive world model' node. A new cylindrical node labeled 'Desired incorrect model elements' is added above the 'Model Editing' node. The color of the new 'Model Editing' node and the 'Desired incorrect model elements' node is set to green. A new arrow is added connecting the 'Correct predictive world model' node to the 'Model Editing' node. A new arrow is added connecting the 'Desired incorrect model elements' node to the 'Model Editing' node. The feedback loop arrow at the bottom right is modified to originate from the 'Model Editing' node and point back to the 'Specifically incorrect predictive world model' node. |  |  E:6 P:4 Q:4 | | |
|  | The large outer circular boundary becomes invisible, while all nodes change from hollow outlines to lightly filled circles in their respective colors to match their outline color. The lower orange node is relabeled from $b_1$ to $u_1$ and move the label from the right side to the left side of the orange node. The central blue node is relabeled from $u_1$ to $y$ and move the label from the left side to the right side of the blue center node. The three inner gray nodes on the left shift downward. On the outer ring, label $u_3$ moves from nearby the node positioned at $10^\circ$ to nearby the node positioned $-30^\circ$. All inner pathes change to a slightly thicker style. A new green connection is added between the orange node and the node $u_3$. The former red connection from the central blue node is removed. The black connection between center blue node $y$ and the node on the $60^\circ$ moves to between the first blue node on the top and the node on the $60^\circ$. The black connection from the first blue node on the top to the node at approximately $60^\circ$ is removed. The large outer circular boundary from the node on left of the top node to node $u_3$ changes from black to green. The path between $y$ and $u_1$ changes from orange to blue. |  |  E:4 P:4 Q:6 |  E:2 P:2 Q:2 |  E:5 P:5 Q:6 |
|  | In the row for $g'(s)$, change the sign from '+' to '-'. In the row for $g(s)$, change the starting value at $-\infty$ from $-\infty$ to $+\infty$ and reverse the direction of the arrow so it points downwards towards $-z_2$. In the row for $\varphi'(s)$, swap the signs in the intervals: change '-' to '+' in the interval between $-\infty$ and $s_0$, and change '+' to '-' in the interval between $s_0$ and $+\infty$. In the row for $\varphi(s)$, change the starting value at $-\infty$ from $+\infty$ to $-\infty$ and reverse the direction of the first arrow so it points upwards towards $\varphi(s_0)$. In the row for $\varphi(s)$, reverse the direction of the second arrow so it points downwards from $s_0$ towards $z_1$. In the row for $\varphi(s),\varphi(s_0)$ is removed. |  |  E:6 P:7 Q:7 | |  E:5 P:6 Q:7 |
|  | The overall topology of the diagram is changed from a single central tree (one top node branching down to a row of three) to a split structure (two top nodes on the left and right with ellipsis dots between them, branching down to a single central node at the bottom). The top-level node text is changed from a single $x^t = f_1(\mu_k^t)$ to two separate nodes: $\bar{x}_{k+1}^t = f_1(\mu_{k+1}^t)$ on the left and $\bar{x}_K^t = f_1(\mu_K^t)$ on the right. The second row nodes are changed from a single $\mu_k^t = f_2(\mu_k^{t-1}, a_k^{t-1}, x_k^{t-1})$ equation to two separate equations: $\mu_{k+1}^t = f_2(\mu_{k+1}^{t-1}, a_{k+1}^{t-1}, x_{k+1}^{t-1})$ on the left and $\mu_K^t = f_2(\mu_K^{t-1}, a_K^{t-1}, x_K^{t-1})$ on the right. The third row nodes are changed from a single $a_k^{t-1} = f_3(\mu_k^{t-1}, \mu_{k+1}^{t-1}, \ldots, \mu_K^{t-1}; x_{k-1})$ equation to two separate equations: $a_{k+1}^{t-1} = f_{3,k+1}(\mu_k^{t-1}, \ldots, \mu_K^{t-1}; x_{k-1})$ on the left and $a_K^{t-1} = f_{3K}(\mu_k^{t-1}, \ldots, \mu_K^{t-1}; x_{k-1})$ on the right. The fourth row is changed from a row of three $\mu$ equations to a single central equation $\mu_k^{t-1} = f_2(\mu_k^{t-2}, a_k^{t-2}, x^{t-2})$. The bottom row is changed from a row of three $x$ equations to a single central equation $x^{t-2} = f_1(\mu_k^{t-2})$. The connectivity arrows are updated to match the new structure: arrows now flow from the two top nodes down to the two middle nodes, which then converge onto the single bottom node. |  | |  E:5 P:5 Q:4 |  E:7 P:6 Q:7 |

Table 18: Scientific figure edits by Gemini-3.1-Pro, GPT-5.6-Sol, and EdiTikZ-9B-RL. Models receive the source image and VLM-generated edit instruction. Human annotations score edit application (E), source preservation (P), and visual quality (Q). Overall quality: ■ very good, ■ good, ■ bad, ■ very bad.

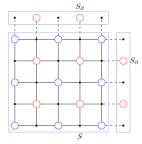
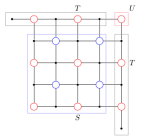
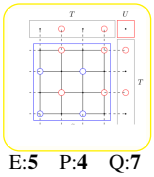
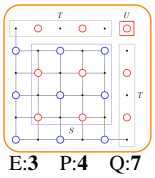
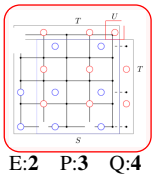
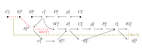
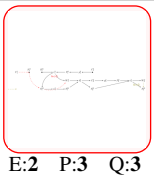
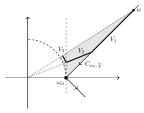
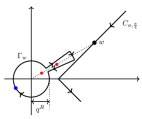
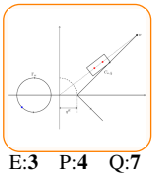
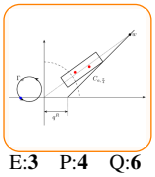
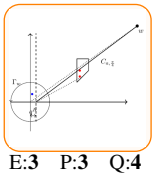
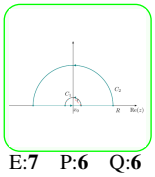
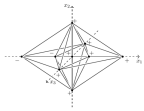
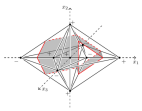
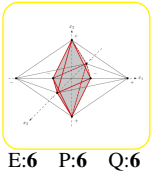
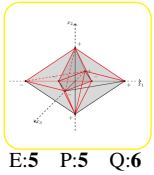
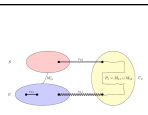
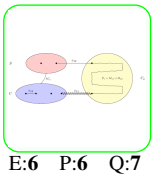
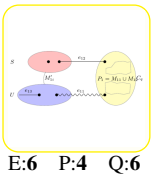
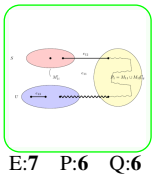
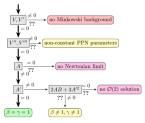
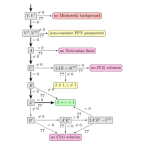
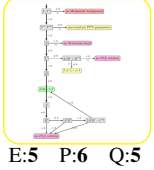
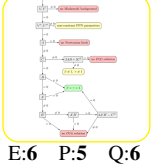
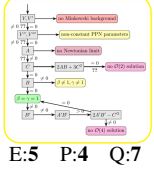
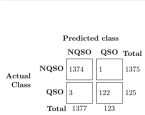
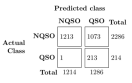
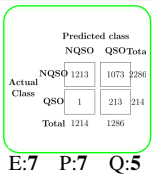
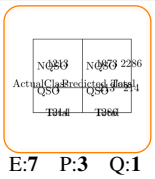
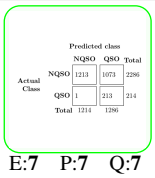
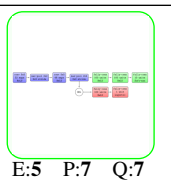
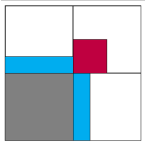
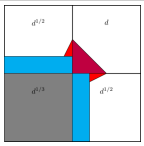
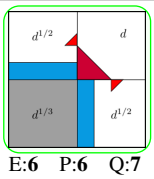
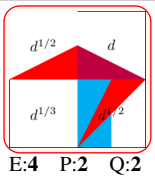
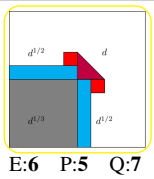
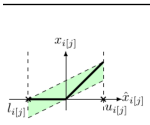
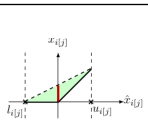
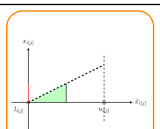
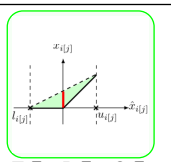
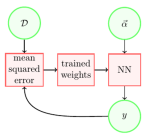
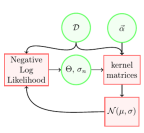
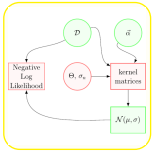
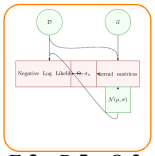
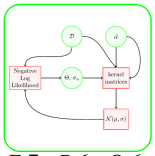
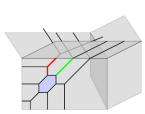
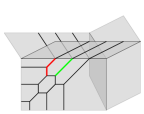
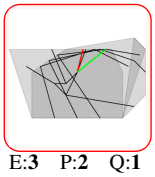
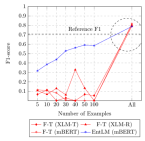
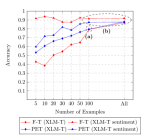
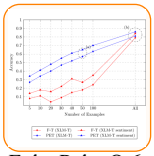
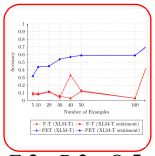
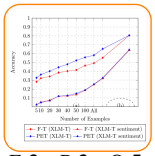
| Source | Edit Instruction | Ground Truth | Gemini-3.1-Pro | GPT-5.6-Sol | EdiTikZ-9B-RL |
|---|---|---|---|---|---|
|  | The grid connectivity is changed. Specifically, the dashed lines connecting the outer boundary nodes to the main grid are replaced by solid lines. Solid lines are added between neighboring nodes nodes of the outer boundary. The blue rectangular region labeled 'S' is resized and repositioned. It now encloses the top-right 3x3 block of the grid, whereas it previously enclosed the entire 4x4 blocks. The label '$S_\partial$' at the top is changed to '$T$'. The label '$S_\partial$' on the right is changed to '$T$'. A new red rectangular box with a red node is added in the top-right corner. A new label '$U$' is added above this red box. |  |  E:5 P:4 Q:7 |  E:3 P:4 Q:7 |  E:2 P:3 Q:4 |
|  | Remove the blue transition node labeled $r_1^1$ and the nodes labeled $p_1^1$, $P_1^1$, $W_1^1$, and $S^1 2_1$ and all connecting arrows at the left side of the diagram. Move the node labeled $C_1^1$ to the position previously occupied by $r_1^1$, directly to the left of $S_1^{11}$. Remove the blue dashed rectangular box and its label 'synchronisationsegment($r_1^1$)'. Remove the text annotation 'No more transitions are added due to 2.b.' located in the top right corner. Add a red dashed path connecting $C_1^1$ to $S_1^{11}$, then to $S_1^{22}$, and finally to $W_1^2$. Add the red text label '$\text{u}_\text{c}(r_1^1)$' positioned below the red path to $S_1^{22}$. Add an short vertical olive dashed path connecting $W_2^2$ to $S_2^{22}$ labeled '$\text{u}_\text{c}(r_2^2)$' in olive to the right of the path. Add another short horizontal path connecting $C_1^1$ to $S_1^{11}$ just below the red path. The two middle points of these short paths are connected with two olive dashed line segments extending from the top left downwards and then to the right. |  |  E:2 P:3 Q:3 | | |
|  | The main diagonal ray and its vertex move leftward from approximately x=0.5 to x=0.3. The ray label changes from $C_{\omega_C, \frac{\pi}{4}}$ to $C_{\alpha, \frac{\pi}{4}}$ and move towards top right. The shaded region, dashed vertical line, dashed quarter-circle, solid arc, and segmented ultra-thick path to $w$ are removed. The labels $V_1$, $V_2$, $V_3$, and $\omega_C$ are removed. A circular contour labeled $\Gamma_w$ is added near the origin, with arrowed arcs and two short arrowed internal segments around the dotted line to $w$. Two small red points are added along the dotted line and one blue point is added on the lower-left part of $\Gamma_w$. A horizontal double-headed arrow labeled $q^R$ is added below the x-axis. The main rays and the newly restored contour segments are drawn very thick. |  |  E:3 P:4 Q:7 |  E:3 P:4 Q:6 |  E:3 P:3 Q:4 |
|  | Remove the second panel (labeled (B)) located on the right side of the figure, which depicts a contour in the lower half-plane. Remove the panel label '$(A)$' from the top-left corner of the remaining diagram. Change the label '$C_R$' on the large outer semicircle to '$C_2$'. Change the label '$C_\varepsilon$' on the small inner semicircle to '$C_1$'. Change the label '$\varepsilon$' indicating the radius of the small semicircle to '$r$'. Change the label '$O$' at the origin to '$x_0$'. Add the label '$R$' at the rightmost point of the large semicircle on the real axis. Remove the arrowhead from the red dashed line representing the radius $r$. |  | | |  E:7 P:6 Q:6 |
|  | The $x_3$ axis (dashed line pointing towards the viewer) is extended to a longer length, and its label '$x_3$' is moved further out along the axis. The sign label '$+$' at the point on the positive $x_1$ axis (inner point) is changed to '$-$'. The sign label '$-$' at the point on the negative $x_1$ axis (inner point) is changed to '$+$'. The sign label '$+$' at the point on the positive $x_3$ axis (front point) is changed to '$-$'. The sign label '$+$' at the point on the negative $x_3$ axis (back point) is changed to '$-$'. A gray shaded region is added covering a large irregular area inside the diagram, roughly spanning from the left vertex marked '$-$' across the central area toward the right vertex marked '$+$', overlaying the underlying triangulated structure. Two closed red outline shapes are added: a larger red polygon/curve (partly dashed, partly solid) tracing an irregular boundary around the left-to-center portion of the diagram near the '$-$' vertex, and a smaller red triangular outline near the right-center portion of the diagram close to the '$-$' label on the right side, both overlapping the gray shaded region. |  | |  E:6 P:6 Q:6 |  E:5 P:5 Q:6 |
|  | The wavy line connecting the red ellipse (S) to the yellow ellipse (Cq) is removed together with its label 'e02'. The wavy line connecting the two points inside the blue ellipse (U) is removed. A wavy line connecting the blue ellipse (U) to the yellow ellipse (Cq) is added to connect the rightmost point in U to the bottom point in Cq at it is labeled 'e11'. The label 'e01' on the top horizontal line is changed to 'e12'. The label 'e01" on the left horizontal line in U is changed to 'e13'. The label 'e02" on the right wavy line in U is removed. The label 'M0i" is changed to 'M1i". The label 'P0 = M01 U M02' inside the yellow ellipse is changed to 'P1 = M11 U M12'. |  |  E:6 P:6 Q:7 |  E:6 P:4 Q:6 |  E:7 P:6 Q:6 |
|  | The yellow result node 'beta != 1, gamma != 1' (positioned beside the bottom-row 'beta = gamma = 1' node) is repositioned to be reached from the 'C' branch's '$2AB + 3C^2$' box via the '!=0' path. The green result node labeled 'beta = gamma = 1' is moved from the bottom of the main vertical flow, positioned directly below 'A' and reached by the '=0' branch arrow, to its original position on the right side, reached via a bold arrow from 'A'. The entire lower section of the flowchart is added. This includes the nodes 'B', 'B", 'A'B", and '$2A'B' - C'^2$', the result node 'no O(4) solution', and the arrows connecting these added nodes. The mathematical expression in the decision box to the right of the fourth level is changed from '$2AB + 3A'^2$' to '$2AB + 3C^2$', with its '=0' branch continuing to point to 'no O(2) solution' and its '??' branch left unresolved as before. The decision node labeled 'A' at the fourth level of the flowchart is changed to a node labeled 'C'. |  |  E:5 P:6 Q:5 |  E:6 P:5 Q:6 |  E:5 P:4 Q:7 |

Table 19: Scientific figure edits by GPT-5.6-Sol, Qwen3.5-4B, and EdiTikZ-4B-RL. Models receive the source image and VLM-generated edit instruction. Human annotations score edit application (E), source preservation (P), and visual quality (Q). Overall quality: ■ very good, ■ good, ■ bad, ■ very bad.

| Source | Edit Instruction | Ground Truth | GPT-5.6-Sol | Qwen3.5-4B | EdiTikZ-4B-RL |
|---|---|---|---|---|---|
|  | The value in the top-left cell (Actual NQSO, Predicted NQSO) changed from 1374 to 1213. The value in the top-right cell (Actual NQSO, Predicted QSO) changed from 1 to 1073. The value in the bottom-left cell (Actual QSO, Predicted NQSO) changed from 3 to 1. The value in the bottom-right cell (Actual QSO, Predicted QSO) changed from 122 to 213. |  |  E:7 P:7 Q:5 |  E:7 P:3 Q:1 |  E:7 P:7 Q:7 |
|  | In the first blue node, change the text '64 maps' to '32 maps'. In the second blue node (max-pool), change the text '3x3' to '2x2'. In the third blue node, change the text '64 maps' to '48 maps'. In the fourth blue node (max-pool), change the text '3x3' to '2x2'. Remove the fifth blue node labeled 'conv 5x5 128 maps ReLU' from the main sequence. In the first green node, change the text '3072 units' to '100 units'. In the second green node, change the text '2048 units' to '100 units'. Remove the first red node labeled 'fully-conn 1024 units ReLU' from the bottom branch. In the second red node, change the text '1024 units' to '100 units'. Move the 'GRL' circle node one position to the left so it is located below the second max-pool node instead of the third convolution node. Change the connection arrow pointing to the 'GRL' node so it originates from the second max-pool node instead of the third convolution node. Change the connection arrow pointing to the first green node so it originates from the second max-pool node instead of the third convolution node. Change the node fill colors from clue to grey, from bright green to a darker green, and from a bright red to a darker red/brown. |  | | |  E:5 P:7 Q:7 |
|  | The purple square in the top-right quadrant is replaced by a purple right-angled triangle. The triangle's vertices are at the center of the figure, the point to the right of the center on the horizontal midline, and the point above the center on the vertical midline. Two small red right-angled triangles are added. One is located in the top-left quadrant, adjacent to the top-left corner of the new purple triangle. The other is in the bottom-right quadrant, adjacent to the bottom-right corner of the new purple triangle. Four mathematical labels are added to the four main quadrants of the figure. The label '$d^{1/3}$' is added to the gray square in the bottom-left. The label '$d^{1/2}$' is added to the white square in the top-left. The label '$d^{1/2}$' is added to the white square in the bottom-right. The label '$d$' is added to the white square in the top-right. |  |  E:6 P:6 Q:7 |  E:4 P:2 Q:2 |  E:6 P:5 Q:7 |
|  | The light green shaded region is changed from a parallelogram (bounded by y=0.5x and y=0.5x-0.5) to a triangle (bounded by the x-axis, the line y=0.5x, and the vertical line x=1). The dashed line segment forming the lower boundary of the original parallelogram (from (0, -0.5) to (2, 0.5)) is removed. The dashed line segment forming the upper boundary of the region (from (0,0) to (2,1)) is made thicker (ultra thick). A thick, red vertical line segment is added, extending from the origin (0,0) upwards to (0, 0.5). The vertical dashed lines at x=0 and x=2 are extended downwards to y=-0.5. The horizontal axis arrow is extended further to the right, and its label '$\hat{x}_{i[j]}$' is moved further to the right. The vertical axis arrow is extended further downwards and upwards, and its label '$x_{i[j]}$' is moved further upwards. |  |  E:3 P:4 Q:6 | |  E:7 P:7 Q:7 |
|  | Change the text in the top-right square node from 'NN' to 'kernel matrices'. Change the text in the middle square node from 'trained weights' to '$\Theta, \sigma_n$' and change its shape from a square to a circle. Change the text in the left square node from 'mean squared error' to 'Negative Log Likelihood'. Change the text in the bottom-right circle node from '$y$' to '$\mathcal{N}(\mu, \sigma)$' and change its shape from a circle to a square. Move the top-left circle node labeled '$\mathcal{D}$' from above the left square node to above the middle circle node. Change the straight arrow from the top-left circle '$\mathcal{D}$' to point to the left square node 'Negative Log Likelihood' with a curved arrow. Add a new curved arrow from the top-left circle '$\mathcal{D}$' pointing to the top-right square node 'kernel matrices'. |  |  E:4 P:6 Q:6 |  E:3 P:5 Q:2 |  E:7 P:6 Q:6 |
|  | The light blue filled hexagonal region is removed from the front face of the structure, located between the red and green highlighted paths. The red line is shortened to touch only the edge connecting the front and upper faces. The short black grid line on the upper-left top surface extending farther backwards is shortened. The black diagonal grid-line connected to the red line extending farther backwards on the upper-right top face is now shortened. |  | | |  E:3 P:2 Q:1 |
|  | The horizontal dashed line at y=0.707 spanning the width of the plot, labeled 'Reference F1', is removed. The legend entries are updated to: 'F-T (XLM-T)', 'F-T (XLM-T sentiment)', 'PET (XLM-T)', and 'PET (XLM-T sentiment)'. The single dashed circle annotation (without a label) enclosing the convergence point of all curves near x='All' is removed. Two dashed oval annotations are added: one labeled '(a)' positioned around the point at x=100, and one labeled '(b)' positioned around the cluster of points near x='All'. The data series using red asterisk markers (F-T mBERT), staying low and volatile around 0.05-0.14 across most x-values with minor fluctuations, then rising sharply to approximately 0.83 at 'All', is removed. A new data series using blue triangle markers (PET XLM-T sentiment) is added as a distinct series. The plotted data series for the blue diamond-marker curve (previously EntLM mBERT) is replaced with new coordinate values representing PET (XLM-T), starting at approximately 0.53 (x=5), rising through 0.61 (x=10), 0.67 (x=20), 0.82 (x=30), 0.73 (x=40), 0.76 (x=50), 0.79 (x=100), and reaching approximately 0.87 at 'All'. The plotted data series for the red triangle-marker curve (previously F-T XLM-R) is replaced with new coordinate values representing F-T (XLM-T sentiment), starting high at approximately 0.92 (x=5), rising to 0.94 (x=10), staying around 0.92 (x=20), dipping to 0.87 (x=30-40), rising to 0.92 (x=50), dropping to 0.8 (x=100), and reaching approximately 0.91 at 'All'. The plotted data series for the red diamond-marker curve (previously F-T XLM-T on F1-score scale) is replaced with new coordinate values representing F-T (XLM-T) on the accuracy scale, starting at approximately 0.43 (x=5), dropping to 0.38 (x=10), then rising steadily through 0.5 (x=20), 0.55 (x=30), 0.62 (x=40), 0.64 (x=50), 0.8 (x=100), and reaching approximately 0.87 at 'All'. The Y-axis label is changed from 'F1-score' to 'Accuracy'. |  |  E:4 P:4 Q:6 |  E:2 P:2 Q:5 |  E:3 P:3 Q:5 |