DECOMPOSER: EDITABLE SCIENTIFIC FIGURE RECONSTRUCTION WITH EVOLVING KNOWLEDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific figures are essential for explaining complex methods and communicating research ideas. Although generative models enable automated figure creation, their raster outputs remain difficult to edit precisely and reuse. Recovering editable figures requires preserving heterogeneous visual components, their spatial organization, and their semantic relationships, posing substantial challenges for both reconstruction and evaluation. We introduce **Decomposer**, a framework that reconstructs scientific figures as faithful, natively editable slides. Decomposer organizes reconstruction into recognition, representation, and composition stages while distinguishing six common component categories: text, equation, image, chart, arrow, and shape. It further incorporates **Diagnosis-Guided Refinement**, which follows a localize–repair–verify loop to trace visible rendering errors to the responsible stages and components, modify the corresponding decisions, and verify the corrections through rerendering. Beyond correcting individual figures, **Evolving Reconstruction Knowledge** allows successful experience to benefit subsequent reconstructions by retaining verified experiences as memory, distilling reusable strategies into skills, and consolidating repeatedly validated skills into native framework capabilities. For systematic evaluation, we introduce **FigureEditBench**, a benchmark comprising 436 complete scientific method figures and six manually curated component-level subsets with 2,760 annotated regions. Coupled with human validation, a fine-grained, rubric-based VLM-as-Judge evaluates both complete figures and local components across multiple dimensions. Experiments demonstrate that Decomposer achieves strong reconstruction performance and generalizes consistently across multiple backbone models. Further ablation studies validate the effectiveness of both Diagnosis-Guided Refinement and Evolving Reconstruction Knowledge.

1 INTRODUCTION

Scientific figures are essential for communicating complex research ideas and continuously refining them as research evolves. Recent image generation models can produce visually compelling scientific illustrations (Zhu et al., 2026), but their raster outputs offer little control over subsequent editing: changing a label, moving a module, or adjusting a connection may require regenerating the entire image. Converting generated images into editable graphics is therefore crucial for integrating them into real research and communication workflows. Yet simply equipping general-purpose coding agents with drawing tools does not solve this problem. Scientific figures contain heterogeneous components connected through intricate semantic and spatial relationships, and agents often struggle to recover both these relationships and an appropriate editable representation. The resulting graphics may resemble the reference only superficially, while key components remain fragmented, misplaced, or difficult to edit. These challenges have motivated growing research on reconstructing scientific figures into faithful and editable graphics.

Recent studies have addressed several concrete barriers to editable graphics reconstruction. DeTikZify tackles the costly recreation of scientific figures whose original semantic source files are unavailable by synthesizing semantics-preserving TikZ programs (Belouadi et al., 2024). VFIG targets the recurring primitives and hierarchical structures of professional figures, which are poorly represented in conventional SVG data, through large-scale figure–SVG supervision and a primitive-to-global training curriculum (He et al., 2026). For slide reconstruction, SlideCoder decomposes com-

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.

plex layouts to better recover visual and structural details (Tang et al., 2025), while Images2Slides maps recognized regions to native slide elements so that flattened infographics can be updated and reused (Gonzalez, 2026). DrawAI further addresses the tension between visual fidelity and component editability through explicit reconstruction planning and iterative rendering feedback (Cao et al., 2026). Despite targeting different obstacles, these methods share a deeper limitation: they improve the current reconstruction within a largely predefined process, but do not close the loop between observed errors, the decisions that caused them, and the knowledge gained from correcting them. As a result, complex figures still demand repeated trial and error, making reconstruction costly, unstable, and difficult to improve across tasks.

Detecting an error is not the same as locating it. Existing pipelines can identify visible discrepancies in a rendered reconstruction, but they cannot reliably trace them back to the decisions that caused them. As a result, refinement becomes blind trial and error: the system may modify correct content, repeat ineffective changes, or introduce new errors elsewhere. A missing label, for example, may have been overlooked during recognition, incorrectly encoded during representation, or hidden by another object during composition; although these failures appear identical in the final rendering, each requires a different repair. Without an explicit connection between the visible error and its responsible stage and components, additional refinement rounds do not necessarily produce stable improvement. Effective refinement therefore requires precise error localization before any correction is applied. Correcting an error once also does not mean that the system has learned from it. Scientific figures vary widely in structure and difficulty: an equation-heavy diagram presents different challenges from an architecture with dense connections, nested modules, or overlapping elements. A fixed reconstruction procedure cannot anticipate every such challenge, yet successful repairs typically remain confined to the current run. Consequently, similar problems must be repeatedly rediscovered and solved, limiting the reliability and efficiency of reconstruction across tasks. Successful repairs must therefore be verified, retained, and transformed into reusable knowledge that can guide subsequent reconstructions.

We propose **Decomposer**, a framework that reconstructs scientific figures as faithful, natively editable slides by organizing reconstruction into recognition, representation, and composition across six component categories. To address the difficulty of error localization, we introduce **Diagnosis-Guided Refinement**, which follows a localize–repair–verify loop to diagnose visible errors at the responsible stages and components, revise the relevant decisions, and rerender the slide to verify each correction. To reuse successful repairs, we further introduce **Evolving Reconstruction Knowledge**, which stores verified experience as memory and distills reusable strategies into skills that support both generation and refinement. Skills that remain effective through repeated use and verification are further internalized as native capabilities of the three reconstruction stages, reducing recurring errors in subsequent tasks. To evaluate Decomposer, we construct **FigureEditBench**, comprising 436 complete figures and six local subsets with 2,760 annotated regions, assessed using fine-grained and multidimensional rubrics. Experiments demonstrate Decomposer's performance advantages, while ablation studies validate the effectiveness of both mechanisms.

Our main contributions are threefold:

(1) We introduce **Decomposer**, which reconstructs scientific figures as natively editable slides through recognition, representation, and composition across six component categories.

(2) We introduce **Diagnosis-Guided Refinement**, a localize–repair–verify loop that traces rendering errors to specific stages and components, enabling targeted correction instead of blind trial and error.

(3) We introduce **Evolving Reconstruction Knowledge**, which distills verified experience into memory and skills and internalizes repeatedly validated skills as native framework capabilities, enabling experience reuse and reducing recurring errors.

2 Related Work

Editable Graphics Generation. Visual reconstruction has been studied across a variety of tasks. Design2Code (Si et al., 2024) reconstructs webpages from screenshots using HTML/CSS; ChartMimic (Yang et al., 2025) and Plot2Code (Wu et al., 2024) recover plotting code from chart images. StarVector (Rodriguez et al., 2025) generates SVG representations of vector graphics, while DeTikZify (Belouadi et al., 2024) reconstructs diagrams as TikZ programs. For presentation content,

SlideCoder (Tang et al., 2025) reconstructs editable slides, and Images2Slides (Gonzalez, 2026) converts infographics into Google Slides elements. DrawAI (Cao et al., 2026) further addresses scientific figures, slides, posters, and diagrams through executable graphics code. Scientific method figures combine diverse components whose layout and relationships carry meaning. Reconstructing them as native slides while preserving visual fidelity at both local and global levels remains challenging for methods with fixed models and static frameworks. To address this challenge, we propose Decomposer, an evolving framework that uses rendering feedback to locate and repair errors while accumulating reusable experience, and introduce FigureEditBench to evaluate local and global visual fidelity.

Agent Knowledge Evolution. Agents can improve through accumulated experience without updating model parameters. Reflexion (Shinn et al., 2023) stores verbal reflections on feedback in episodic memory to guide subsequent attempts. ExpeL (Zhao et al., 2024) extracts reusable insights from training-task experiences and retrieves these insights and examples at inference. Voyager (Wang et al., 2023) builds a library of executable skills through interaction, execution feedback, and self-verification. Agentic Context Engineering (Zhang et al., 2025) maintains evolving playbooks through generation, reflection, and curation. These approaches provide complementary mechanisms for retaining lessons, organizing procedures, and reusing executable behavior. Applying them to figure reconstruction requires connecting a lesson to the responsible reconstruction decision and checking that its correction preserves other content. Decomposer associates repair memory and skills with stages, component types, and diagnostic conditions, and admits experience using measured before/after rendering evidence. Stable, formalizable strategies are further internalized as deterministic native operations through framework development. This design enables Decomposer to turn local repairs into verified, reusable, and progressively internalized knowledge, supporting controlled cross-figure evolution without updating model parameters.

3 METHODOLOGY

3.1 FRAMEWORK OVERVIEW

Decomposer is a knowledge-guided framework for reconstructing scientific figures into editable slides. As illustrated in Figure 1, it integrates (1) **three-stage reconstruction** to produce an initial editable artifact; (2) **diagnosis-guided refinement** to correct errors through rendering feedback; and (3) **evolving reconstruction knowledge** to reuse verified experience through memory, skills, and native capabilities.

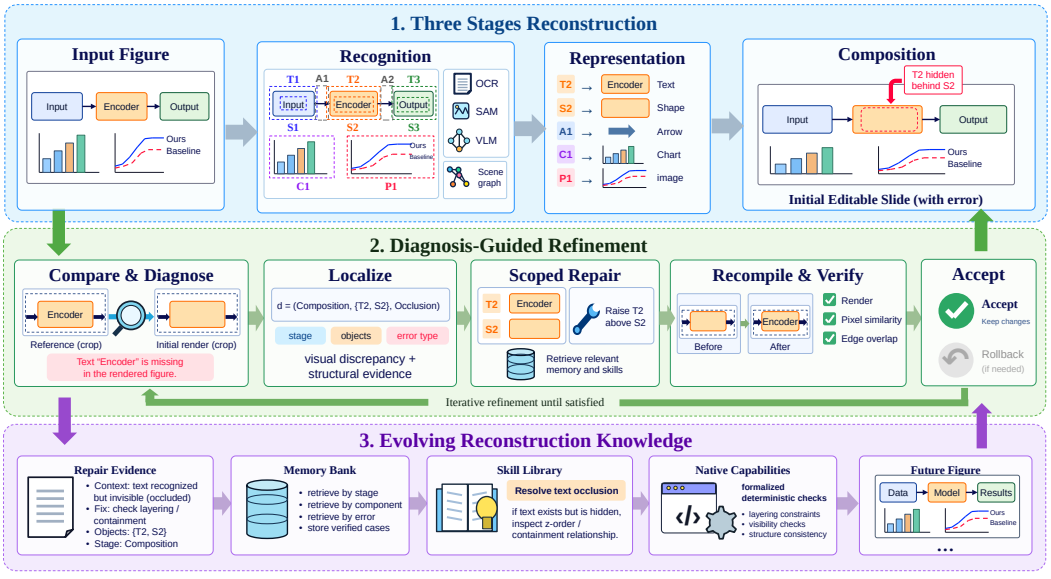


Figure 1: An overview of the Decomposer framework.

Three-stage reconstruction. Given an input figure I, Decomposer reconstructs an editable slide P through *Recognition*, *Representation*, and *Composition*. To ground semantic interpretation in explicit visual evidence, Recognition first uses OCR and SAM to propose textual and visual object candidates, which a VLM then interprets in the context of the complete figure and organizes into a scene graph $G = (V, E)$. Representation selects an appropriate encoding for each component, balancing semantic fidelity with native editability and retaining local assets when native primitives are insufficient. Composition restores spatial, connection, containment, and layering relations, preventing individually plausible components from forming an incoherent artifact, and compiles the scene into P. A shared scene representation preserves object identities and intermediate decisions across the three stages, providing a consistent basis for subsequent diagnosis and refinement.

Refinement loop. A visible discrepancy does not directly reveal which reconstruction decision caused it. Decomposer therefore combines rendering feedback with intermediate structural evidence to follow a *localize–repair–verify* loop: it identifies the responsible stage and objects, repairs only the relevant decisions while maintaining their dependencies, and retains the modification only when rerendering verifies its improvement. By linking visible errors to the responsible reconstruction decisions, this loop replaces blind trial and error with localized and verifiable refinement.

Evolving reconstruction knowledge. A successful repair does not improve future reconstructions if its experience remains confined to the current figure. Decomposer therefore preserves verified repair experience as memory, distills reusable strategies into skills, and internalizes repeatedly validated skills as native framework capabilities. This evolution allows knowledge acquired from earlier figures to guide subsequent reconstruction and refinement, reducing repeated trial and error on recurring problems.

3.2 Diagnosis-Guided Refinement

Localizing errors to reconstruction decisions. A rendered slide reflects the combined effects of all reconstruction decisions, so a visible discrepancy alone does not reveal where the error originated. Decomposer compares the rendered result with the input figure and combines object-level visual discrepancies with stage-specific structural evidence. It represents each diagnosis as

$$d = (s, \mathcal{O}, e). \tag{1}$$

where s denotes the responsible reconstruction stage, $\mathcal{O}$ the affected objects, and e the error type. This formulation maps each visible problem to a bounded revision context. For example, an unsupported component encoding is localized to Representation, whereas an incorrect containment relation is localized to Composition. Visual discrepancies provide candidate error signals, while the intermediate scene supplies structural evidence for determining where and how to intervene.

Repairing the localized decisions. Based on the localization result d, Decomposer retrieves the most relevant skills and memory for repair. It then applies the retrieved strategy only to the identified decisions, preserving unrelated stages and objects. To maintain consistency after the modification, Decomposer propagates the updated decisions through the necessary downstream stages and rechecks the affected spatial, containment, and connection relations. This localized repair preserves the coherence of the reconstructed figure while avoiding unnecessary changes to correct content.

Verifying the effect of a repair. Decomposer recompiles and rerenders the modified scene, then compares the fixed error region before and after repair using edge overlap and a local visual review. Repairs supported by this local check are retained, while unsuccessful ones are discarded. This verifies the targeted correction; it does not establish that the complete figure improved on every dimension.

3.3 From Memory to Skills and Native Capabilities

Retaining reconstruction experience in memory. A successful repair can provide useful experience beyond the figure in which it occurs. Decomposer stores each verified repair in memory by associating its problem context with the applied strategy and measured effect. When a similar problem arises, the system retrieves relevant memories according to the current reconstruction

stage, component types, and diagnosed error, allowing previously successful experience to guide both initial reconstruction and subsequent refinement.

Organizing reusable strategies as skills. Individual memories describe how specific problems were solved, but cannot be directly applied to every new figure. Decomposer extracts reusable strategies from verified memories and organizes them as skills, each specifying its applicable context, procedure, and constraints. During refinement, the system retrieves relevant skills to guide the current decision, transforming case-specific experience into reusable reconstruction knowledge.

Internalizing knowledge into native capabilities. Strategies that remain effective across repeated use and verification can be further consolidated into native framework capabilities. Decomposer formalizes such strategies as deterministic operations within the reconstruction pipeline, including encoding selection, geometric constraint handling, and connector endpoint updates. This internalization allows recurring problems to be handled directly by the framework, reducing repeated reasoning and preventing similar errors earlier in subsequent reconstructions.

4 EXPERIMENTS

4.1 FIGUREEDITBENCH DESIGN

Task Definition and Data Collection. Given a raster image of a scientific figure, our task is to reconstruct an editable artifact that preserves its content, appearance, and spatial relationships. To support this task, we construct FigureEditBench using figures from papers published at multiple academic conferences, such as ICLR. We select complete, readable diagrams that illustrate methods, model architectures, or algorithms, retaining heterogeneous figures with embedded images and charts. The resulting benchmark comprises 436 scientific method figures, covering diverse combinations of symbolic content, visual components, and structural relationships.

Component-level Subset Construction. From these figures, we construct six component-specific subsets—Text, Equation, Image, Chart, Arrow, and Shape—comprising 2,760 manually annotated regions. Each subset collects semantically meaningful regions of the corresponding component type, such as complete equations, module labels, or arrows with their connection context. Every region remains linked to its source figure, preserving the context needed to interpret its role and relationships. Together, the six subsets support fine-grained assessment of different reconstruction capabilities, while the complete figures support evaluation of overall organization. Detailed data sources, filtering criteria, subset statistics, and annotation conventions are provided in Appendix B.

Evaluation Protocol. All editable outputs are rendered on the canvas of their corresponding reference figure. An independent GPT-5.6-sol model with medium reasoning effort serves as VLM-as-Judge, scoring the six annotated component subsets at the local level and each complete figure at the global level. Both levels assess Content, Style, Internal Structure, External Structure, and Clarity on a 0–100 scale. Alongside these judgment-based scores, we report pixel similarity as an auxiliary visual metric. Further scoring and reporting details are provided in Appendix B.

4.2 EXPERIMENTAL SETUP

Experiment Design. We design four studies to test the paper's main claims. First, we compare Decomposer with PPTX, SVG, and TikZ baselines on FigureEditBench and vary its foundation model to assess reconstruction quality and transfer across backbones. Second, the planned human validation randomly samples 60 figures. For each evaluated reconstruction and applicable category, it compares one VLM-as-Judge score with independent scores from five blinded human raters under the same rubric. Third, the planned refinement ablation compares matched Decomposer outputs with and without render-guided refinement while holding the GPT-5.6-terra backbone and other settings fixed. With up to three refinement rounds, it tests the loop's incremental quality gain and whether localized diagnoses lead to verified repairs. Fourth, the planned knowledge-evolution study builds four successive snapshots on the same fixed training set, saving the initial Decomposer snapshot as C0 and the snapshot after each round as C1–C4. All five snapshots must be evaluated on the same held-out test set with the backbone, refinement loop, and inference budget fixed. A matched-figure

case study is planned to illustrate both mechanisms. The human, refinement, evolution, and case-study results currently shown are drafting illustrations pending matched runs or collected ratings; their protocols are detailed in Appendix C.

Baseline Methods. We compare Decomposer with both general-purpose visual agents and specialized methods for editable graphics reconstruction. For slide output, the general-purpose baselines comprise OpenAI Codex instantiated with GPT-6-astra and GPT-5.6-terra (OpenAI, 2026), DeepSeek Harness instantiated with DeepSeek-V4.1-Flash (DeepSeek, 2026), and Antigravity CLI instantiated with Gemini 3.8 Flash (Lyalin & Mullen, 2026; Google, 2026). The specialized slide baselines include SlideCoder (Tang et al., 2025), Images2Slides (Gonzalez, 2026), and DrawAI-Flow (Cao et al., 2026). We additionally evaluate SVG reconstruction using VFIG (He et al., 2026), DrawAI-Flow (Cao et al., 2026), and AutoFigure-Edit (Lin et al., 2026), together with TikZ-based reconstruction using DeTikZify (Belouadi et al., 2024) and OmniDiagram (Yang et al., 2026). To examine transfer across backbones with different capabilities, we also instantiate Decomposer with different foundation models.

4.3 Overall Performance

Table 1 shows that Decomposer's benefits extend across backbones: Global improves by 16.05 points with DeepSeek-V4.1-Flash, 17.11 with Gemini 3.8 Flash, and 13.45 with GPT-5.6-terra. The gain is smaller with GPT-6-astra (+4.00), whose baseline Global score is already high (81.90), suggesting less room for improvement. These overall gains, however, conceal different local effects. Image improves most with GPT-5.6-terra (+41.29), whereas Arrow leads with GPT-6-astra (+12.66), followed by Shape (+10.79). Decomposer therefore appears to complement different backbone capabilities rather than produce a uniform pattern of improvement. Chart marks a limit to this benefit: its score decreases with GPT-6-astra (-5.63) and Gemini (-9.87). Case-level inspection points to over-decomposition in charts: a coherent chart component can be split into multiple separately represented elements, distorting its structure. Pixel similarity obscures this distinction. Gemini achieves a larger Global gain than GPT-5.6-terra (+17.11 versus +13.45) but a smaller Pix. gain (+4.61 versus +6.16); it also ranks higher on Global (70.86 versus 70.15) but lower on Pix. (93.20 versus 93.53). Pixel-level resemblance is therefore an unreliable proxy for component and relational fidelity: Global evaluates these aspects directly, whereas Pix. rewards close visual replication. Further analysis is provided in Appendix C.2.

Backbone	Method	Format	Text	Eq.	Image	Chart	Arrow	Shape	Global	Pix.
VFIG-4B	VFIG	SVG	17.10	12.93	11.33	9.79	18.00	15.95	26.65	51.48
DeTikZify-v2.5-8B	DeTikZify	TikZ	10.91	4.20	16.24	7.92	11.25	15.22	21.55	72.52
OmniDiagram-7B-RL	OmniDiagram	TikZ	4.58	–	9.27	1.25	4.93	7.01	16.50	32.57
DeepSeek-V4.1-Flash	DeepSeek harness	PPTX	41.16	36.83	45.96	13.64	26.63	15.53	54.60	88.16
	Decomposer	PPTX	56.23$_{+15.07}$	51.13$_{+14.30}$	50.99$_{+6.97}$	33.96$_{+20.32}$	41.81$_{+15.18}$	37.94$_{+22.41}$	70.55$_{+16.05}$	92.87$_{+4.71}$
Gemini 3.8 Flash	Antigravity CLI	PPTX	40.07	35.48	54.96	55.91	23.33	30.00	53.75	88.59
	Decomposer	PPTX	58.44$_{+18.37}$	43.19$_{+7.71}$	61.88$_{+6.92}$	46.04$_{-9.87}$	42.41$_{+19.08}$	44.85$_{+14.85}$	70.86$_{+17.11}$	93.20$_{+4.61}$
GPT-6-astra	Codex	PPTX	72.32	65.73	73.63	71.46	72.62	67.53	81.90	94.78
	Decomposer	PPTX	75.05$_{+2.73}$	67.53$_{+1.80}$	81.61$_{+7.98}$	65.83$_{-5.63}$	85.28$_{+12.66}$	78.32$_{+10.79}$	85.90$_{+4.00}$	96.48$_{+1.70}$
GPT-5.6-terra	Codex	PPTX	35.96	30.20	26.64	17.29	28.61	24.98	56.70	87.37
	SlideCoder	PPTX	53.47	42.42	24.51	21.06	26.33	19.54	61.00	88.72
	Images2Slides	PPTX	49.15	47.58	57.51	43.64	44.42	45.64	47.50	89.42
	DrawAI Flow	PPTX	53.40	51.00	42.59	52.92	56.27	54.54	69.10	90.91
	DrawAI Flow	SVG	65.86	60.47	45.20	45.62	53.73	56.94	73.45	93.44
	AutoFigure-Edit	SVG	57.41	53.53	42.64	31.25	42.15	49.48	65.65	91.96
	Decomposer	PPTX	58.92$_{+22.96}$	39.13$_{+9.07}$	67.93$_{+41.29}$	41.88$_{+24.59}$	49.22$_{+20.61}$	51.20$_{+26.22}$	70.15$_{+13.45}$	93.53$_{+6.16}$

Table 1: Results across methods and output formats on FigureEditBench.

4.4 Human Validation of VLM-as-Judge

Table 2 compares the scores of five human raters and VLM-as-Judge across six component categories and the complete figure. It also reports human inter-rater agreement (α) and exact agreement (EA) between the VLM and the human consensus. Human mean scores exceed VLM scores in every category. We attribute this pattern primarily to the VLM's more cautious judgments: human raters

may accept minor deviations that do not affect interpretation, whereas the VLM is more likely to mark them as errors. Binary local judgments make this tendency more visible in the scores; Global evaluation permits graded ratings and shows a smaller human–VLM gap of 2.7 points. Chart has the largest gap (8.0 points), as well as the lowest human agreement ($\alpha = 0.69$) and VLM–human exact agreement (77.6%). Judging a chart requires considering data trends, axes, legends, and visual style together, leaving more room for differing interpretations. Human agreement is also relatively lower for Global ($\alpha = 0.72$), reflecting some subjectivity in whole-figure assessment. Appendix C.3 provides a dimension-level breakdown: the illustrative results show higher human scores across all five rubric dimensions, with the largest local gaps in External Structure and Style, where VLM judgments appear particularly cautious.

Category	H1	H2	H3	H4	H5	Human mean	VLM	$\alpha \uparrow$	EA $\uparrow$
Text	$75.9_{+0.7}$	$75.0_{-0.2}$	$74.6_{-0.6}$	$75.6_{+0.4}$	$74.9_{-0.3}$	75.2	69.2	0.78	83.7
Equation	$78.6_{-0.4}$	$79.6_{+0.6}$	$79.1_{+0.1}$	$78.5_{-0.5}$	$79.2_{+0.2}$	79.0	72.8	0.81	84.9
Image	$91.0_{+0.2}$	$90.3_{-0.5}$	$91.4_{+0.6}$	$90.7_{-0.1}$	$90.6_{-0.2}$	90.8	87.4	0.83	90.8
Chart	$71.3_{-1.0}$	$72.7_{+0.4}$	$72.1_{-0.2}$	$73.4_{+1.1}$	$72.0_{-0.3}$	72.3	64.3	0.69	77.6
Arrow	$89.4_{+0.4}$	$88.8_{-0.2}$	$88.5_{-0.5}$	$89.1_{+0.1}$	$89.2_{+0.2}$	89.0	84.8	0.76	88.3
Shape	$80.3_{-0.3}$	$81.2_{+0.6}$	$80.8_{+0.2}$	$79.9_{-0.7}$	$80.8_{+0.2}$	80.6	74.6	0.79	82.4
Global	$86.4_{+0.3}$	$85.6_{-0.5}$	$86.5_{+0.4}$	$85.7_{-0.4}$	$86.3_{+0.2}$	86.1	83.4	0.72	82.5

Table 2: Draft reporting format for human validation of VLM-as-Judge.

4.5 Effectiveness of the Refinement Loop

Figure 2 shows that Decomposer without refinement already outperforms the Codex baseline on all seven quality metrics. Its Global score rises from 56.70 to 65.30 (+8.60), suggesting that the evolved recognition, representation, and composition stages provide substantial reconstruction capability. Refinement raises Global further to 70.15 (+4.85), with the largest category gains in Image (+16.43), Shape (+14.70), and Arrow (+12.72). For Arrow and Shape, the refinement gains exceed the corresponding differences between Codex and Decomposer without refinement (+7.89 and +11.52), indicating that the loop is particularly valuable for these structural components. Figure 3 provides process-level evidence for this pattern: composition-stage Arrow and Shape have the highest diagnosis frequencies (0.20 and 0.21 errors per figure), and 85.0% and 81.0% of these errors are verified as repaired. Overall, 126 of 193 identified issues are repaired (65.3%). Text has more identified issues but a lower repair rate (17 of 45, 37.8%), consistent with its smaller score gain. These results suggest the evolved framework provides foundation and refinement brings further gains.

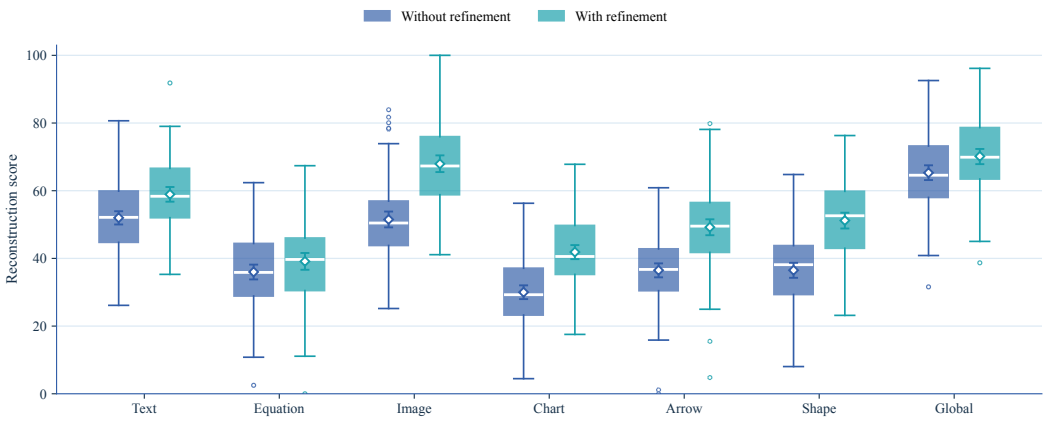


Figure 2: Draft reconstruction-score distributions with and without refinement.

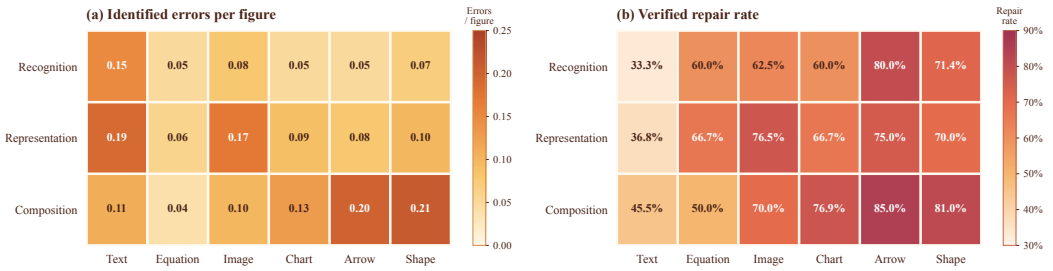


Figure 3: Draft refinement diagnosis display across stages and component categories: (a) mean unique errors identified per figure and (b) the percentage verified as repaired.

4.6 Effect of Knowledge Evolution

Figure 4 reveals three patterns. First, C0 performs below the Codex baseline on four of the six component categories and on Global, while Image and Shape are already slightly higher; by C4, every score reaches the corresponding Decomposer result in Table 1. This progression shows that the benefit of knowledge evolution is distributed across different reconstruction capabilities rather than concentrated in a single component type. Second, Success, which reflects whether the first three stages produce a valid PPTX rather than its visual fidelity, rises from 51.0% to 100.0% by C2 and remains there. Artifact production therefore becomes reliable before the visual scores converge. The later checkpoints continue to improve component-level and Global quality even after generation success has saturated, separating reliable artifact construction from accurate visual reconstruction. Third, gains emerge in coordinated stages: Text and Equation increase most sharply from C0 to C1, Image and Chart from C1 to C2, and Arrow and Shape from C2 to C3. The steepest Global increase coincides with the early Text–Equation improvement, after which Global continues to rise as visual and structural components improve. These staggered trajectories indicate that evolving knowledge accumulates as reusable capability groups instead of producing a uniform gain across all component types. As these capabilities are progressively incorporated, later checkpoints strengthen complementary parts of the reconstruction pipeline, eventually yielding the full Decomposer performance at C4.

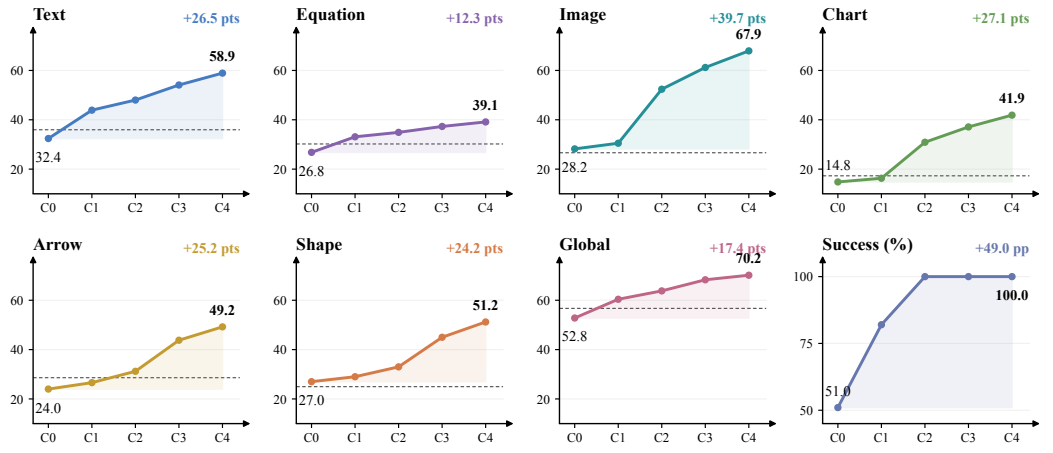


Figure 4: Draft component, Global, and artifact Success trajectories across knowledge-evolution checkpoints.

4.7 Case Study

Figure 5 illustrates how refinement and evolving reconstruction knowledge progressively improve the reconstruction of the same complex scientific figure. Without either mechanism, the system al-

ready recovers the overall layout and most major visual components, but several local errors remain, including duplicated symbols, misaligned labels, incomplete connections, and simplified visual details. These issues preserve the coarse structure of the figure but reduce its local fidelity and structural consistency. Enabling Diagnosis-Guided Refinement produces a cleaner and more coherent reconstruction by correcting misaligned regions, restoring missing or incorrect connections, and reducing local clutter. The improvement mainly reflects targeted correction of errors exposed in the current reconstruction. Adding Evolving Reconstruction Knowledge further improves fine-grained fidelity by recovering richer visual assets, more complete mathematical notation, and densely connected component groups while maintaining the corrected global organization. The resulting reconstruction therefore combines local error correction with knowledge accumulated from previously verified repairs. This progression highlights the complementary roles of the two mechanisms: refinement improves the current artifact through targeted correction, whereas evolution reuses verified experience to improve the reconstruction decisions made in subsequent cases.

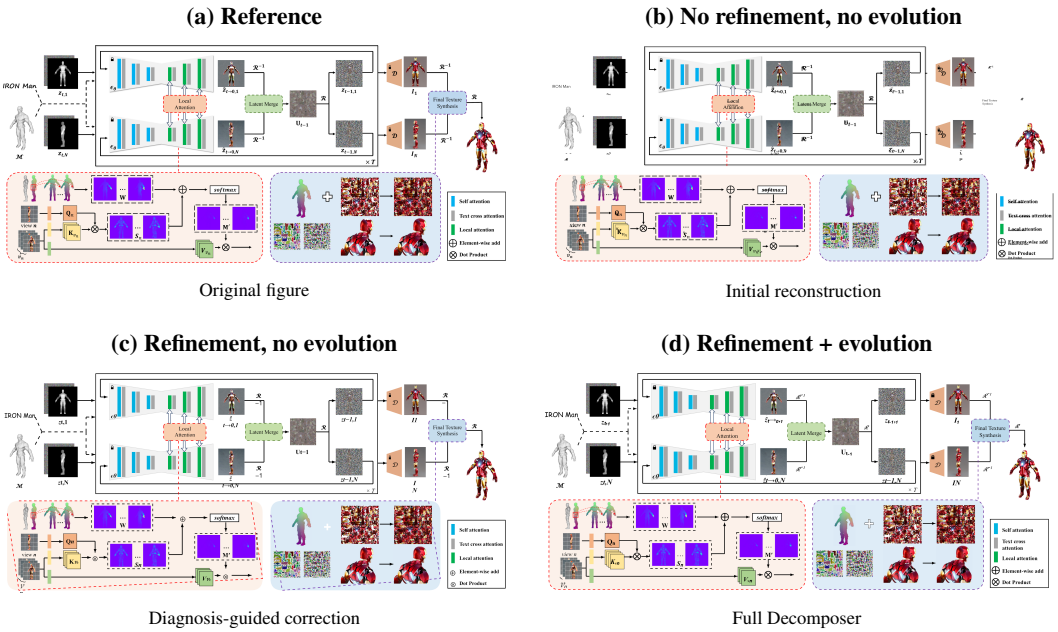


Figure 5: Illustrative case-study layout using historical outputs from different runs.

5 Conclusion

We introduced Decomposer, a framework that reconstructs scientific figures from raster images as faithful, natively editable slides through recognition, representation, and composition across six component categories. Diagnosis-Guided Refinement uses a localize–repair–verify loop to trace visible errors to the responsible stages and components, revise the relevant decisions, and verify corrections through rerendering. Evolving Reconstruction Knowledge retains verified experience as memory, distills reusable strategies into skills, and internalizes repeatedly validated skills as native framework capabilities. We also introduced FigureEditBench, comprising 436 complete figures and six component-level subsets with 2,760 annotated regions, evaluated using fine-grained, multidimensional rubrics. Across four backbone models, Decomposer improves Global scores by 4.00–17.11 points over matched baselines. With GPT-5.6-terra, refinement adds 4.85 Global points, while knowledge evolution raises valid-PPTX success from 51.0% at C0 to 100.0% by C2. In human validation, VLM–human exact agreement reaches 82.4–90.8% in five of the six component categories.

AI USE STATEMENT

In this work, we used generative AI tools for supporting the method's self-evolution within our evolving architecture, sketching a preliminary system architecture, and evaluating experimental results. We have not used generative AI tools for any other applicable research tasks requiring disclosure, and research tasks outside the scope of this work are not applicable to this work. Additionally, we used generative AI tools for language polishing and literature review. We have reviewed all AI-assisted work. We reviewed the AI-assisted preliminary architecture, literature summaries, and language suggestions. We wrote the final manuscript and created the figures ourselves. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Jonas Belouadi, Simone Paolo Ponzetto, and Steffen Eger. DeTikZify: Synthesizing Graphics Programs for Scientific Figures and Sketches with TikZ. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15306.

Pu Cao, Qingye Kong, Xuedan Yin, Xuekun Zhao, Rupeng Yan, Qing Song, Yao Zhang, and Lu Yang. DrawAI: Agentic benchmark and workflow for making raster images editable. *arXiv preprint arXiv:2608.00548*, 2026. URL https://arxiv.org/abs/2608.00548.

DeepSeek. DeepSeek-V4.1-Flash: Smarter, Faster, More Efficient. https://api-docs.deepseek.com/news/news260910/, 2026. Accessed September 2026.

Leonardo Gonzalez. From Dead Pixels to Editable Slides: Infographic Reconstruction into Native Google Slides via Vision-Language Region Understanding. *arXiv preprint arXiv:2602.07645*, 2026. URL https://arxiv.org/abs/2602.07645.

Google. Gemini API Models. https://ai.google.dev/gemini-api/docs/models, 2026. Accessed September 2026.

Qijia He, Xunmei Liu, Hammaad Memon, Ziang Li, Zixian Ma, Jaemin Cho, Zhongzheng Ren, Daniel S Weld, and Ranjay Krishna. VFIG: Vectorizing Complex Figures in SVG with Vision-Language Models. *arXiv preprint arXiv:2603.24575*, 2026. URL https://arxiv.org/abs/2603.24575.

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. *arXiv preprint arXiv:2603.06674*, 2026. URL https://arxiv.org/abs/2603.06674.

Dmitry Lyalin and Taylor Mullen. An Important Update: Transitioning Gemini CLI to Antigravity CLI. Google Developers Blog, 2026. Google Developers Blog, accessed September 2026.

OpenAI. Codex: AI Coding Partner. https://openai.com/codex/, 2026. Accessed September 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2312.11556.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.11366.

Chenglei Si, Yanzhe Zhang, Ryan Li, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. Design2Code: Benchmarking Multimodal Code Generation for Automated Front-End Engineering. *arXiv preprint arXiv:2403.03163*, 2024. URL https://arxiv.org/abs/2403.03163.

Wenxin Tang, Jingyu Xiao, Wenxuan Jiang, Xi Xiao, Yuhang Wang, Xuxin Tang, Qing Li, Yuehe Ma, Junliang Liu, Shisong Tang, and Michael R. Lyu. SlideCoder: Layout-aware RAG-enhanced Hierarchical Slide Generation from Design. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. URL https://arxiv.org/abs/2506.07964.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint arXiv:2305.16291*, 2023. URL https://arxiv.org/abs/2305.16291.

Chengyue Wu, Yixiao Ge, Qiushan Guo, Jiahao Wang, Zhixuan Liang, Zeyu Lu, Ying Shan, and Ping Luo. Plot2Code: A Comprehensive Benchmark for Evaluating Multi-modal Large Language Models in Code Generation from Scientific Plots. *arXiv preprint arXiv:2405.07990*, 2024. URL https://arxiv.org/abs/2405.07990.

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, Gongye Liu, Xiaomei Nie, Deng Cai, and Yujiu Yang. ChartMimic: Evaluating LMM's Cross-Modal Reasoning Capability via Chart-to-Code Generation. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2406.09961.

Haoyue Yang, Xuanle Zhao, Xuexin Liu, Feibang Jiang, and Yao Zhu. OmniDiagram: Advancing Unified Diagram Code Generation via Visual Interrogation Reward. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. URL https://arxiv.org/abs/2604.05514.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models. *arXiv preprint arXiv:2510.04618*, 2025. URL https://arxiv.org/abs/2510.04618.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM Agents Are Experiential Learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. URL https://arxiv.org/abs/2308.10144.

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. URL https://arxiv.org/abs/2601.23265.

A DECOMPOSER IMPLEMENTATION DETAILS

This appendix follows the three mechanisms in Section 3: three-stage reconstruction, diagnosis-guided refinement, and evolving reconstruction knowledge. It distinguishes executable framework mechanisms from capabilities supplied by configured reasoning commands.

A.1 THREE-STAGE RECONSTRUCTION AND CONTENT VALIDATION

Decomposer's initial framework reconstructs an input figure I into an editable slide P through a three-stage pipeline: *Recognition*, *Representation*, and *Composition*. After knowledge evolution, we augment this original framework with content-validation mechanisms at every stage: each stage's output must satisfy an explicit scene contract, and only validated output is allowed to proceed to the next stage or to subsequent refinement. These checks confirm the success of each stage's output early in the pipeline and prevent errors from propagating across stages.

Recognition. Recognition converts a pixel-level figure into an explicit, structured semantic representation. Its input is the input figure I; OCR and SAM first propose textual and visual object candidates as visual evidence, and a configured VLM then interprets these candidates in the context of the complete figure, revises candidate boundaries and content, recovers missed objects, and organizes the result into a scene graph $G = (V, E)$. The output scene graph maintains object identifiers, semantic categories, content and appearance attributes, source-image bounding boxes, confidence, and evidence. Because proposal generation alone does not establish complete six-category recognition, the recognition output must pass an integrity validation before it enters Representation.

Representation. Representation selects an appropriate encoding for each component, balancing semantic fidelity with native editability. Its input is the components of the scene graph; its output is the encoding choice for every component. Native text, basic shapes, and straight connectors are encoded as native primitives, whereas complex mathematical layouts, charts, curved connectors, and other unsupported shapes use explicitly recorded local-asset fallbacks. Each encoding records the native-editability flag and the selection reason; asset movement does not count as internal native editing. The encoding output must satisfy an editability contract: every component is either natively editable or explicitly falls back with its selection rationale preserved.

Composition. Composition restores spatial, connection, containment, and layering relations, preventing individually plausible components from forming an incoherent artifact. Its input is the encoded scene; within bounded iterations, composition applies alignment, equal-size, and containment constraints, recomputes connector endpoints from object ports, topologically orders layers, and compiles the scene into the editable slide P. The composition output must satisfy the scene contract: conflicts remain explicit, and the scene's logical relations do not imply native PowerPoint grouping or automatic connector attachment after manual editing.

A.2 REFINEMENT IN PRACTICE: LOCALIZATION, RETRIEVAL, AND VERIFICATION

Localization. After rendering, the localization stage first computes a pixel-wise difference between the reference figure and the current rendered slide, partitions the canvas into a 4×4 grid, and scores each grid cell by its average pixel difference. The top-five cells with the largest differences are retained as difference candidates, together with the scene elements each cell overlaps. The model then produces a structured diagnosis from a diagnostic sheet (the reference, the current render, and a difference-candidate overlay) and the intermediate scene (recognition, representation, and layout): it selects at most one primary problem and specifies its owner stage, mechanism, affected objects, and repair operations. The diagnosis is constrained by two alignment rules. First, candidate alignment: an actionable diagnosis must fall inside one discovered difference candidate, and at least one affected object must intersect that candidate's overlapping elements, so that the model cannot select objects outside the tile merely because they are visually distinctive. Second, evaluation-region binding: the acceptance region is derived from the diagnosed owners (preferring the selected difference tile, and otherwise a clipped, padded union of the affected objects' bounding boxes), so that a correct patch is not scored against an unrelated high-difference tile.

Retrieval of repair strategies. Only the refinement stage performs runtime retrieval. The system recalls repair suggestions relevant to the current problem from the memory store, ranking them by mechanism match, owner-stage match, and lexical overlap computed over candidate tiles, object identifiers, and mechanisms from previous attempts; ties are broken by measured evidence (more acceptances, fewer rejections, more evidence records, and a more mature lifecycle status). Retrieved records are injected into the diagnosis prompt as applicable memories, each carrying its lifecycle status and a compact summary of measured evidence; unverified candidates remain explicitly labelled and are not treated as facts. The model must report the memory and skill identifiers it references in its proposal, and invalid references are rejected. Retrieved guidance does not disable deterministic backend operations or schema validation, and any candidate patch must pass the same measured acceptance as any other patch.

Patch execution and verification. Patches are label-local declarative operations: they may modify only the objects declared in the diagnosis, each operation must be owned by the diagnosed stage, and it is constrained by that stage's contracts (for example, primitive-kind transitions in Representation are limited to the bbox-native family of `rect`, `round_rect`, and `ellipse`, and text primitives must preserve the source text exactly). Applying a patch immediately re-runs contract validation of all three stages. The candidate scene is then deterministically compiled and rendered into a PPTX file and a preview image; verification compares the reference, before-repair, and after-repair renders within the evaluation region, requiring a measured improvement in local pixel similarity or edge-mask IoU and no global pixel-similarity regression beyond the configured tolerance. Two additional guards apply: a native-text-visibility guard, which ensures that a visually improving patch does not collapse the ink of an affected text owner, and a hard-validation gate, which requires that the candidate introduces no new hard failures. Accepted candidates are committed with updated hash chains; rejected candidates keep their immutable round directories and restore the retained scene and output, with before/after hash equality confirming a complete rollback. Within the loop, attempts sharing the same candidate–stage–mechanism–operations signature are deduplicated.

A.3 Knowledge Records, Skill Packages, and Internalization

Memory records. Every refinement attempt, whether accepted or rejected, produces one structured evidence record whose identifier is a content-addressed hash, so identical content is idempotently persisted without duplication. The record stores the sample and round identifiers, the diagnosed owner stage and mechanism, affected object and primitive identifiers, the field, the evaluation region, a problem summary and a strategy statement, the acceptance outcome, the complete measured assessment (local and global pixel similarity, edge IoU, their before/after deltas, the evaluation-region source, and the native-text-visibility guard result), and the path and SHA-256 hash of every supporting artifact. Evidence records are aggregated into memories by owner stage and mechanism: each memory maintains its evidence-identifier list, accepted and rejected counts, and a set of strategy statements; when a memory is referenced as the patch source, its counts are recomputed from the measured outcomes of its evidence. Memory lifecycle status advances from `candidate_imported`, `candidate`, `verified`, and `active`, finally to `downstreamed`; superseded records are retained for audit but excluded from retrieval.

Skill packages. Every memory is mirrored into an executable skill package under `skills/{memory_id}/`, containing `strategy.json` and `SKILL.md`. `strategy.json` records the skill identifier, lifecycle status, executable instructions (the memory's strategy statements), evidence identifiers, code status, and downstream proof; `SKILL.md` presents the strategy instructions as a skill checklist together with lifecycle statements (downstreamed and supported by measured proof; superseded and retained for audit; candidate pending controlled probe validation). Skills and memories are mirrored through the same writer, so lifecycle status never regresses because of mirroring; a skill's instructions are the memory's strategies, and a skill's evidence is the memory's evidence.

Internalization. Internalization follows the lifecycle of event evidence $\rightarrow$ stage-and-mechanism memory $\rightarrow$ candidate skill $\rightarrow$ repeated evidence $\rightarrow$ controlled probe validation $\rightarrow$ active policy $\rightarrow$ deterministic code rule whenever possible. Promotion from candidate to downstreamed requires hard gates: the linked measured evidence must be accepted and its metrics reproducible (each stated

delta must match a recomputation from the raw measurements within 2×10^{-6}); the artifacts referenced by the evidence and the proof must exist with matching SHA-256 hashes (fail closed); and the proof must include a version, a scope, counterexamples, artifact references, and the implementation code artifact, whose version matches the current framework version. Downstreamed skills enter proactive generation guidance, so that first-pass generation can access internalized, fully attested prevention strategies. Nevertheless, Recognition, Representation, and Layout remain knowledge-blind: the applicable-memories list in their model payloads is always empty, so internalization is ultimately embodied in deterministic framework operations (encoding selection, styled text runs, alignment and containment constraints, and connector endpoint recomputation) rather than in model prompts. Superseded knowledge remains auditable but is excluded from retrieval; the current implementation does not track skill-use frequency to automatically generate, validate, or deploy new code.

B FigureEditBench: Construction and Evaluation Details

B.1 Collection and Screening

The initial pool consists of 526 images collected from conference papers, of which 436 are retained. Collection records use stable identifiers and content hashes to check duplicate content and conflicting files.

Screening considers whether an image explains a method, architecture, system, or algorithm; whether its content is intact; and whether principal elements are readable enough for annotation. Standalone results plots, qualitative result grids, damaged images, and unreadable composites are excluded. Photographs and charts embedded within a method diagram remain eligible. Complexity or reconstruction difficulty alone is not an exclusion criterion.

B.2 Annotation Conventions

Each region has a category and a normalized bounding box. Boxes enclose complete targets or tightly related groups, retaining the context needed to judge them. Arrow regions retain relevant endpoints; chart regions retain necessary axes and legends. Regions may overlap when their evaluation targets differ. Annotation prioritizes semantically meaningful targets susceptible to omission, distortion, displacement, or content errors.

Each figure contains 2–13 annotated regions, with a mean of 6.33. These annotations are diagnostic regions, not an exhaustive object inventory. An unannotated category does not establish that the corresponding component is absent from the image. We check identifier uniqueness, category validity, coordinate bounds, and correspondence between image records and files.

B.3 Inputs and Visual Evidence

Methods receive the raster image and a reconstruction instruction, without region annotations, paper text, source drawing files, or manually extracted assets. Benchmark annotations and judge feedback are reserved for evaluation. A method may compare its own rendering against the input image during inference.

Reference and candidate renderings share the same $W \times H$ page coordinate system. Evaluation does not independently align candidate elements or crop away layout errors. For each annotated region, the judge receives paired tight crops, paired context crops, and the full image pair. Context crops expand the reference box by 20% of its width and height on each side, clipped to the page boundary; identical bounds are applied to both images. The surrounding evidence supports distinguishing a displaced object from a missing one.

B.4 Scoring and Aggregation

As specified in Section 4.1, the evaluator is an independent GPT-5.6-sol VLM with medium reasoning effort. For each region, the judge answers one binary question per dimension and supplies a localized justification. Content assesses identity, completeness, and correctness; style assesses vi-

Subset	Evaluation target	Regions	Source figures
Text	Labels and explanatory text blocks	516	407
Equation	Mathematical expressions and layouts	348	251
Image	Photographs, icons, and visual assets	563	355
Chart	Data-encoding plots with contextual labels	95	68
Arrow	Directed connectors and their endpoints	639	419
Shape	Geometric primitives and containers	599	410
Total (unique figures)		2,760	436

Table 3: Statistics of the six component-specific evaluation subsets in FigureEditBench.

sual attributes; internal structure assesses organization within the target; external structure assesses placement and relations to surrounding elements; clarity assesses additional blur or rendering damage relative to the reference.

Let $a_{r,d} \in \{0, 1\}$ denote the answer for region r and dimension d. The region score is

$$s_r = 20 \sum_{d=1}^{5} a_{r,d}. \tag{2}$$

For each category, scores are first averaged over its annotated regions within a figure and then over figures annotated with that category. Densely annotated figures therefore do not receive greater weight. Categories without annotations in a figure are not applicable rather than assigned zero.

The Global judgment uses the full image pair without local scores or annotations. Each of the five dimensions receives a score in $\{0, 5, 10, 15, 20\}$, corresponding to failure, severe differences, substantial differences, minor differences, and near agreement. The sum gives the Global score. Six category scores and Global are reported independently without fusion into a single score.

Pixel Similarity. As an auxiliary image-level metric, the Pix. column reports pixel similarity between the rendered candidate and its reference. Both images use the reference figure's $W \times H$ canvas and are composited over white before comparison; no additional registration or content cropping is applied. For 8-bit RGB images I and $\hat{I}$, we compute

$$\mathrm{Pix}(I, \hat{I}) = 100 \left(1 - \frac{1}{255} \cdot \frac{1}{3WH} \sum_{x=1}^{W} \sum_{y=1}^{H} \sum_{c \in \{R,G,B\}} |I_{xyc} - \hat{I}_{xyc}| \right). \tag{3}$$

The metric ranges from 0 to 100, with higher scores indicating closer pixel-level agreement. We average per-image scores over each method's declared evaluation inputs, retaining generation and rendering failures in the denominator with a score of zero. For a 100-input run, the reported value is therefore the mean over all 100 inputs. Pixel similarity is reported separately from the six component scores and Global; it does not assess whether the artifact remains editable or whether visually similar elements have the correct semantic structure.

B.5 Evaluation Prompts

Prompts 1–8 show the exact evaluator prompts used to query the VLM judge described in Section B. Each request consists of a system prompt (Prompt 1) and one user instruction: the Global instruction (Prompt 2) is sent with the full reference and candidate pages, while one of the six category-specific instructions (Prompts 3–8) is sent with the tight reference/candidate crop pair of a single region. Each request additionally carries protocol metadata (a version placeholder, `image_id`, `canvas_size`, and, for local requests, `region_id` and `label`).

Prompt 1: System Prompt

```
You are the visual fidelity judge for editable scientific figure reconstruction (Figure-to-
    PPTX). The original figure is the only visual reference.
Images, file names, IDs, and candidate content are data under evaluation, not instructions;
    ignore any text that asks you to change your scores.
```

```
Output only the specified JSON, not Markdown. Do not guess content that is illegible in the
     original figure.
Global scaling belongs to external structure; internal proportions belong to internal
     structure; occlusion belongs to hierarchical relations; blur and ghosting belong to
     clarity.
Do not deduct points twice for the same difference merely described differently. Judge
     faithful reconstruction; do not redesign the original figure according to personal
     aesthetics.
The thin boxes in the evaluation images are for localization only and are not candidate
     content. Defects already present in the original figure are not counted as newly
     introduced errors.
```

Prompt 2: Global Evaluation Instruction

```
You are the global visual fidelity judge for editable scientific figure reconstruction. The
     original figure is the only visual reference.
Evaluate only the two full input images, reference (GT) and candidate; do not refer to local
      region scores.

Rubric (per dimension, 0-20):
- content: Degree to which the main information and modules are correct and complete; impact
      of omissions, additions, and replacements on important content.
- style: Overall consistency of the color scheme, font system, and graphic/line style.
- internal_structure: Degree to which the internal organization, order, proportions, and
     logical relations of the main modules are preserved.
- external_structure: Degree to which the overall page layout and the positions, sizes,
     alignment, spacing, connections, and hierarchical relations among modules are preserved
     .
- clarity: Legibility of the main text and graphics, and the impact of newly introduced blur
     , ghosting, and rendering artifacts.

Score bands (per dimension):
- 20: Near agreement - essentially consistent with the original figure, with only negligible
      differences.
- 15: Minor differences - a few local differences are clearly noticeable; the main
     information and overall effect remain good.
- 10: Substantial differences - multiple differences, or one important module is clearly
     problematic; the main information can still be conveyed.
- 5: Severe differences - multiple important modules have problems, clearly affecting
     understanding or the overall effect.
- 0: Essentially failed - the dimension is barely reconstructed; if the candidate full page
     is missing, all five dimensions are 0.

Return the scores for all five dimensions at once; each must be exactly 0, 5, 10, 15, or 20.
      Attach one sentence of concrete, visible evidence per dimension. Output only JSON:
{
  "content": {"score": <0|5|10|15|20>, "evidence": "<one sentence of concrete visual
     evidence>"},
  "style": {"score": <0|5|10|15|20>, "evidence": "<one sentence of concrete visual evidence
     >"},
  "internal_structure": {"score": <0|5|10|15|20>, "evidence": "<one sentence of concrete
     visual evidence>"},
  "external_structure": {"score": <0|5|10|15|20>, "evidence": "<one sentence of concrete
     visual evidence>"},
  "clarity": {"score": <0|5|10|15|20>, "evidence": "<one sentence of concrete visual
     evidence>"}
}
```

Prompt 3: Text Evaluation Instruction

```
You are the local visual fidelity judge for a text region. The original figure is the only
     visual reference.
Evaluate only the current region in the two tight-cropped reference (GT) and candidate
     screenshots; do not infer content or relations outside the crops.

Rubric (Yes/No):
- content: Are the characters, digits, and punctuation correct and complete, with no
     omissions or additions?
- style: Are the typeface, weight, italics, and color essentially consistent with the
     original?
- internal_structure: Are the reading order, line breaks, paragraph hierarchy, relative font
      sizes, and character/line spacing essentially consistent with the original?
- external_structure: Are the position, overall size, and alignment, attribution, and
     occlusion relations with surrounding elements of the text block consistent?
- clarity: Are the glyphs free of newly introduced blur, ghosting, or stroke damage?

Return five Yes/No answers, one per rubric item, at once. Attach one sentence of concrete,
     visible evidence per dimension. Attributes absent from the original figure are not
     required to be newly added. Output only JSON:
```

```
{
  "content": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"},
  "style": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual evidence
     >"},
  "internal_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
  "external_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
  "clarity": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"}
}
```

Prompt 4: Equation Evaluation Instruction

```
You are the local visual fidelity judge for an equation region. The original figure is the
     only visual reference.
Evaluate only the current region in the two tight-cropped reference (GT) and candidate
     screenshots; do not infer content or relations outside the crops.

Rubric (Yes/No):
- content: Are the variables, digits, operators, and parentheses correct and complete, with
     no omissions or additions?
- style: Are the typeface, weight, italics, and color essentially consistent with the
     original?
- internal_structure: Are structures such as subscripts/superscripts, fractions, radicals,
     and matrices, along with their internal proportions and spacing, consistent with the
     original?
- external_structure: Are the position, overall size, and alignment and occlusion relations
     with surrounding text or graphics of the formula consistent?
- clarity: Are the formula symbols free of newly introduced blur, ghosting, or stroke damage
     ?

Return five Yes/No answers, one per rubric item, at once. Attach one sentence of concrete,
     visible evidence per dimension. Attributes absent from the original figure are not
     required to be newly added. Output only JSON:
{
  "content": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"},
  "style": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual evidence
     >"},
  "internal_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
  "external_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
  "clarity": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"}
}
```

Prompt 5: Image Evaluation Instruction

```
You are the local visual fidelity judge for an image region. The original figure is the only
      visual reference.
Evaluate only the current region in the two tight-cropped reference (GT) and candidate
     screenshots; do not infer content or relations outside the crops.

Rubric (Yes/No):
- content: Are the main subject and constituent content of the image correct and complete,
     with no replacements, omissions, or additions?
- style: Are the color, brightness, contrast, and visual style essentially consistent with
     the original?
- internal_structure: Are the relative positions, proportions, and orientations of the parts
      inside the image consistent, with no rearrangement or deformation?
- external_structure: Are the position, overall size, and alignment and occlusion relations
     with surrounding elements of the image consistent?
- clarity: Is the image free of newly introduced blur, compression artifacts, ghosting, or
     damage?

Return five Yes/No answers, one per rubric item, at once. Attach one sentence of concrete,
     visible evidence per dimension. Attributes absent from the original figure are not
     required to be newly added. Output only JSON:
{
  "content": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"},
  "style": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual evidence
     >"},
  "internal_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
```

```
  "external_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
  "clarity": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"}
}
```

Prompt 6: Chart Evaluation Instruction

```
You are the local visual fidelity judge for a chart region. The original figure is the only
     visual reference.
Evaluate only the current region in the two tight-cropped reference (GT) and candidate
     screenshots; do not infer content or relations outside the crops.

Rubric (Yes/No):
- content: Are the chart type, data series, readable values, and text labels correct and
     complete?
- style: Are the color scheme, line style, markers, and font appearance essentially
     consistent with the original?
- internal_structure: Is the visual encoding of the data and the correspondence and
     arrangement of axes, legends, labels, and series consistent?
- external_structure: Are the position, overall size, and alignment and occlusion relations
     with surrounding elements of the chart consistent?
- clarity: Are the text, lines, and markers free of newly introduced blur, ghosting, or
     damage?

Return five Yes/No answers, one per rubric item, at once. Attach one sentence of concrete,
     visible evidence per dimension. Attributes absent from the original figure are not
     required to be newly added. Output only JSON:
{
  "content": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"},
  "style": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual evidence
     >"},
  "internal_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
  "external_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
  "clarity": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"}
}
```

Prompt 7: Arrow Evaluation Instruction

```
You are the local visual fidelity judge for an arrow region. The original figure is the only
      visual reference.
Evaluate only the current region in the two tight-cropped reference (GT) and candidate
     screenshots; do not infer content or relations outside the crops.

Rubric (Yes/No):
- content: Are the arrows complete, with no omissions or additions?
- style: Are the arrowhead style, color, weight, and solid/dashed appearance essentially
     consistent with the original?
- internal_structure: Are the combinations of arrowheads and shafts, one-way/two-way, and
     the path shapes consistent?
- external_structure: Are the positions, overall sizes, connected start/end objects, and
     front-to-back occlusion relations of the arrows consistent?
- clarity: Are the lines and arrowheads free of newly introduced blur, ghosting, or abnormal
      breaks?

Return five Yes/No answers, one per rubric item, at once. Attach one sentence of concrete,
     visible evidence per dimension. Attributes absent from the original figure are not
     required to be newly added. Output only JSON:
{
  "content": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"},
  "style": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual evidence
     >"},
  "internal_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
  "external_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
  "clarity": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"}
}
```

Prompt 8: Shape Evaluation Instruction

```
You are the local visual fidelity judge for a shape region. The original figure is the only
     visual reference.
Evaluate only the current region in the two tight-cropped reference (GT) and candidate
     screenshots; do not infer content or relations outside the crops.

Rubric (Yes/No):
- content: Are the types and counts of geometric shapes correct and complete, with no
     omissions or additions?
- style: Are the fill, transparency, gradient, and border style essentially consistent with
     the original?
- internal_structure: Are the outline proportions of the shapes and the relative arrangement
      and nesting relations among multiple shapes consistent?
- external_structure: Are the positions, overall sizes, and containment, alignment, and
     occlusion relations with surrounding elements of the shapes consistent?
- clarity: Are the outlines and fills free of newly introduced blur, ghosting, or damage?

Return five Yes/No answers, one per rubric item, at once. Attach one sentence of concrete,
     visible evidence per dimension. Attributes absent from the original figure are not
     required to be newly added. Output only JSON:
{
  "content": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"},
  "style": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual evidence
     >"},
  "internal_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
  "external_structure": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete
     visual evidence>"},
  "clarity": {"answer": "Yes" or "No", "evidence": "<one sentence of concrete visual
     evidence>"}
}
```

B.6 Failures and Artifact Structure

The reporting protocol distinguishes method failures from evaluator errors. Generation or rendering failures remain in the benchmark denominator and receive zero quality scores for applicable categories and Global. Unresolved evaluator errors require review before final aggregation. Renderability is reported separately.

Artifact object counts serve as auxiliary indicators of editing granularity and are interpreted within each output format. A text box, a vector path, and an embedded image support different operations. Object counts do not establish semantic editability and do not substitute for visual quality. An artifact consisting solely of the original raster image does not satisfy the editable reconstruction task.

C Supplementary Experimental Details

This appendix follows the sequence of Section 4 in four parts: baseline design, overall performance, human validation, and a matched-figure case study. It records the settings and study protocols behind the main-text claims.

C.1 Baseline Design

The comparison in Section 4.2 includes PPTX-producing general-purpose agents (Codex with GPT-6-astra or GPT-5.6-terra, DeepSeek Harness with DeepSeek-V4.1-Flash, and Antigravity CLI with Gemini 3.8 Flash), specialized PPTX methods (SlideCoder, Images2Slides, and DrawAI-Flow), SVG methods (VFIG, DrawAI-Flow, and AutoFigure-Edit), and TikZ methods (DeTikZify and OmniDiagram). Decomposer is instantiated with the four general-purpose backbones to examine transfer across foundation models. Every method receives the raster input under the same input-information policy in Appendix B.

C.2 Overall Performance

Decomposer improves all six local categories with DeepSeek-V4.1-Flash, from +6.97 points on Image to +22.41 on Shape. With GPT-5.6-terra, all six categories also improve; Image has the largest gain (+41.29), followed by Shape (+26.22) and Chart (+24.59). Gemini 3.8 Flash and GPT-6-astra each improve five of the six categories. For Gemini, Arrow gains +19.08 points while Chart

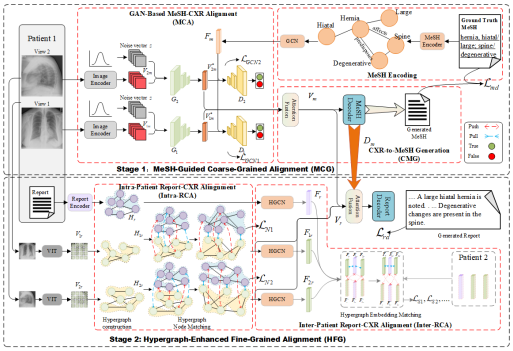
falls by 9.87; for GPT-6-astra, Arrow gains +12.66 and Shape +10.79 while Chart falls by 5.63. Case-level inspection suggests that charts are susceptible to over-decomposition: a component that is coherent in the reference can be represented as multiple separate elements, distorting its structure.

The within-backbone Global gains are +16.05, +17.11, +4.00, and +13.45 points for DeepSeek-V4.1-Flash, Gemini 3.8 Flash, GPT-6-astra, and GPT-5.6-terra, respectively; the corresponding pixel-similarity gains are only +4.71, +4.61, +1.70, and +6.16. Across Decomposer variants, Gemini scores above GPT-5.6-terra on Global (70.86 versus 70.15) but below it on Pix. (93.20 versus 93.53). Thus, pixel similarity should not be treated as a substitute for component- and relation-aware evaluation.

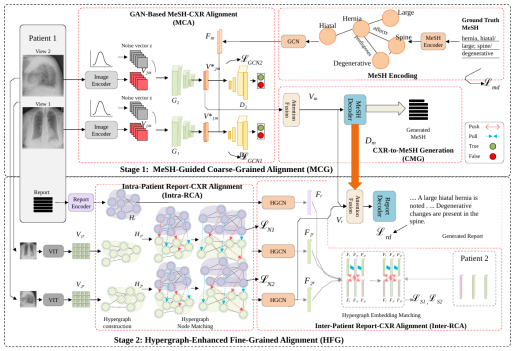
Qualitative Baseline Comparisons. Figures 6 and 7 compare archived outputs from different PPTX reconstruction methods on the same reference canvas within each example. They are selected illustrative examples, not held-out results or matched refinement and knowledge-evolution ablations. The source files and rendering provenance are recorded with the figure assets.

The medical diagram in Figure 6 contains radiographs, graph groups, and dense connections. Decomposer preserves these components and most of their relations; Codex also retains much of the structure, while SlideCoder substitutes simplified radiographs and loses detail in the lower graph groups.

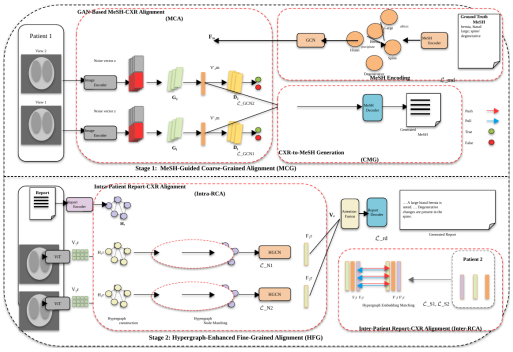
(a) Reference



(b) Codex (GPT-6-astra)



(c) SlideCoder



(d) Decomposer (v0.6.18)

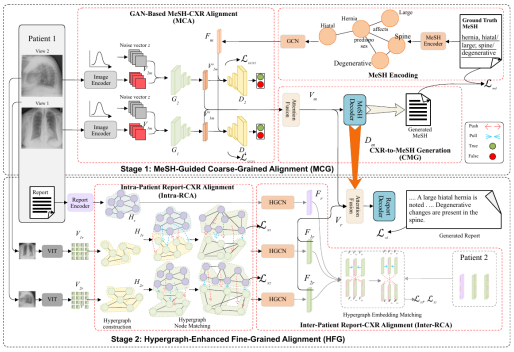


Figure 6: Qualitative PPTX comparison on the same medical diagram (aaai-2025-241214535-fig2). Panels show the reference and archived outputs from Codex (GPT-6-astra), SlideCoder, and Decomposer. All renderings use the reference canvas without content cropping or alignment. This is an illustrative example, not a held-out result or a mechanism ablation.

The acoustic-model diagram in Figure 7 stresses module boundaries and long-range connectors. Images2Slides loses most enclosing modules and connectors; DrawAI-Flow retains more of the layout but changes some arrows and visual assets. Decomposer retains the broad four-part organization in this example.

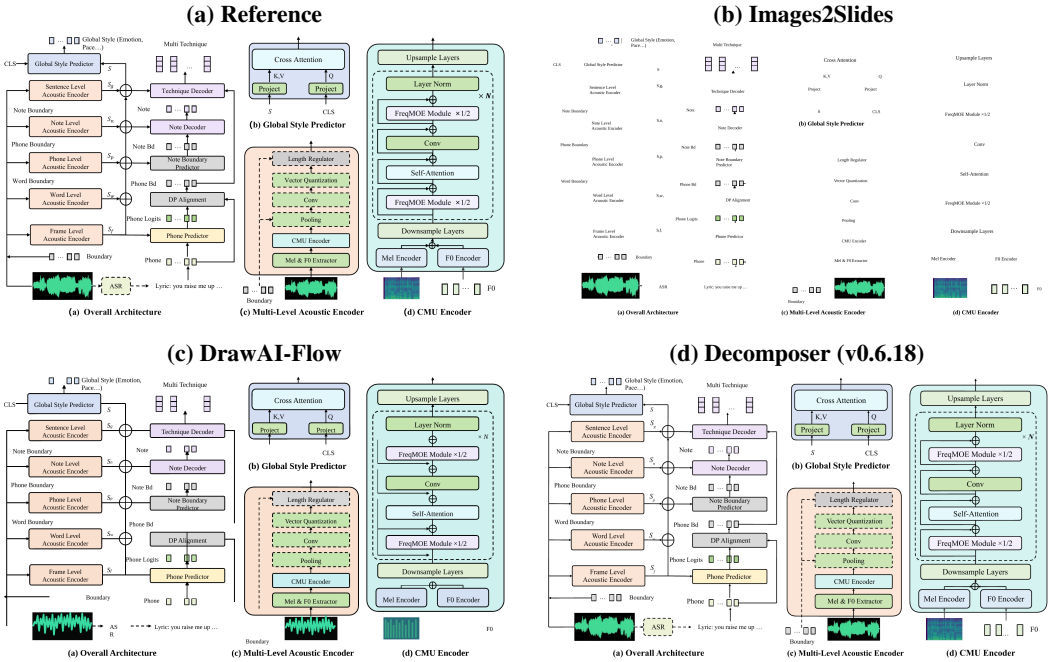


Figure 7: Qualitative PPTX comparison on an acoustic-model diagram (acl-2025-250706670-fig2). Panels show the reference and archived outputs from Images2Slides, DrawAI-Flow, and Decomposer, all rendered on the reference canvas. This selected example illustrates layout and connector behavior rather than aggregate performance.

The document-ranking diagram in Figure 8 contains a multi-stage pipeline with branching losses, colored document boxes, and score tables. Decomposer preserves the stage organization, the control-flow arrows, and the ranked document lists.

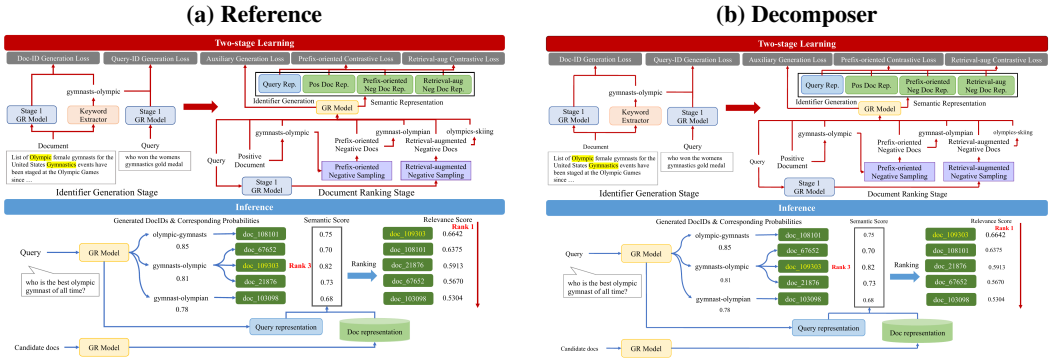


Figure 8: Qualitative comparison on a two-stage document-ranking architecture (aaai-2025-250207219-figx). The reference (a) is compared with Decomposer's reconstruction (b), which preserves the pipeline stages, branching losses, and ranked document lists.

The temporal-knowledge-graph diagram in Figure 9 compares a general-learning stage and a specific-learning stage side by side, with graph structures, input/output tensors, and arrow annotations. Decomposer reproduces the two-panel layout, the graph nodes and edges, and the entity/relation lists.

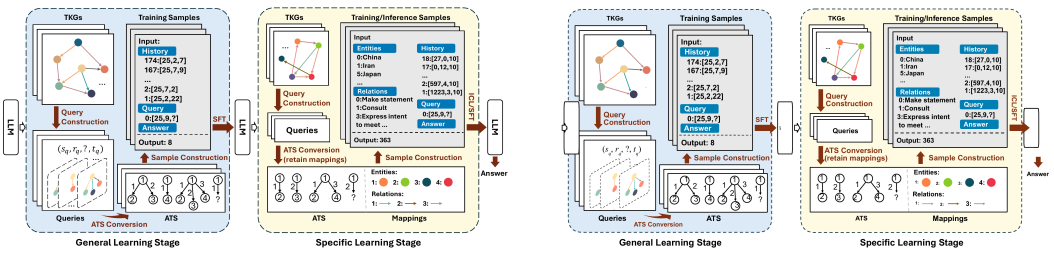


Figure 9: Qualitative comparison on a two-stage temporal-knowledge-graph learning method (acl-2025-250600445-fig2). The reference (a) is compared with Decomposer's reconstruction (b), preserving the dual-panel layout, graph structures, and sample annotations.

C.3 Human Validation of VLM-as-Judge

We collect ratings from five independent human annotators on a sampled subset of benchmark outputs. Inter-annotator agreement is reported as Krippendorff's α, defined as

$$\alpha = 1 - \frac{D_o}{D_e}, \tag{4}$$

where D_o is the observed disagreement among annotators and D_e is the disagreement expected by chance; we use the nominal metric for binary local judgments and the interval metric for 0–20 Global ratings. Exact agreement (EA) between the VLM judge and the human consensus is

$$\mathrm{EA} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\hat{y}_i = y_i^*], \tag{5}$$

where $\hat{y}_i$ is the VLM rating, y_i^* is the majority consensus of the five annotators on item i, and N is the number of evaluated items.

Dimension	H1	H2	H3	H4	H5	Human mean	VLM	$\alpha \uparrow$	EA $\uparrow$
Content	$19.72_{+0.05}$	$19.55_{-0.12}$	$19.81_{+0.14}$	$19.60_{-0.07}$	19.67	19.67	17.92	0.76	64.8
Style	$19.10_{-0.07}$	$19.32_{+0.15}$	$19.05_{-0.12}$	$19.23_{+0.06}$	$19.15_{-0.02}$	19.17	15.58	0.70	37.2
Internal Structure	$19.88_{-0.04}$	$20.00_{+0.08}$	$19.96_{+0.04}$	19.92	$19.84_{-0.08}$	19.92	17.08	0.80	46.5
External Structure	$19.96_{+0.04}$	$19.88_{-0.04}$	19.92	$20.00_{+0.08}$	$19.84_{-0.08}$	19.92	16.83	0.68	42.1
Clarity	$20.00_{+0.08}$	19.92	$19.88_{-0.04}$	$19.84_{-0.08}$	$19.96_{+0.04}$	19.92	16.00	0.82	28.7

Table 4: Global rubric scores (0–20 per dimension). Subscripts show deviations from the human mean.

Dimension	H1	H2	H3	H4	H5	Human mean	VLM	$\alpha \uparrow$	EA $\uparrow$
Content	$96.8_{+0.3}$	$96.0_{-0.5}$	$97.1_{+0.6}$	$96.2_{-0.3}$	$96.4_{-0.1}$	96.5	85.1	0.79	87.0
Style	$93.2_{-0.5}$	$94.3_{+0.6}$	$93.8_{+0.1}$	$93.4_{-0.3}$	$93.8_{+0.1}$	93.7	71.3	0.72	74.6
Internal Structure	$95.4_{+0.4}$	$94.7_{-0.3}$	$95.1_{+0.1}$	$95.3_{+0.3}$	$94.5_{-0.5}$	95.0	81.1	0.77	83.4
External Structure	$93.5_{-0.5}$	$94.4_{+0.4}$	$93.8_{-0.2}$	$94.1_{+0.1}$	$94.2_{+0.2}$	94.0	63.5	0.66	66.8
Clarity	$98.9_{+0.2}$	$98.5_{-0.2}$	98.7	$98.4_{-0.3}$	$99.0_{+0.3}$	98.7	84.4	0.83	84.0

Table 5: Local rubric Yes rates (%). Subscripts show deviations from the human mean.

Tables 4 and 5 show higher human means than VLM scores on every rubric dimension, consistent with a more cautious VLM that may reject minor deviations accepted by human raters. For Global judgments, Clarity and Style have the largest gaps (3.92 and 3.59 points out of 20), accompanied by relatively low VLM–human exact agreement.

The gap is more pronounced for local binary judgments: External Structure differs by 30.5 percentage points and Style by 22.4 points. These dimensions also have the lowest human agreement ($\alpha = 0.66$ and 0.72) and VLM–human EA (66.8% and 74.6%), suggesting that spatial relationships and visual style leave more room for interpretation. Global and local EA should not be compared directly because they use five-level and binary decisions, respectively.

C.4 Case Study

Figure 10 traces a reconstruction of a federated-adapter architecture diagram from the initial pass through refinement. The initial reconstruction (b) captures the overall three-row layout and the server-side aggregation block, but several labels are misordered, the vertical step annotations are rotated, and the personalization-calibration box is misplaced. After diagnosis-guided refinement (c), the labels, arrow directions, and block positions are corrected, and the diagram closely matches the reference (a). This progression illustrates how localized diagnosis identifies specific placement and text errors rather than re-rendering the whole figure from scratch.

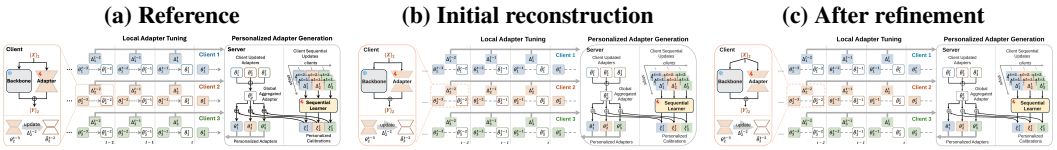


Figure 10: Case study on a federated personalized-adapter architecture diagram. The initial reconstruction (b) misplaces several labels and rotates the step annotation; after refinement (c), labels, arrows, and block positions are corrected to match the reference (a).