Paper2Figure: A Multi-Agent Collaborative System for Figure Generation Towards Academic Research Paper

Supplementary Material

A. Generation Prompts for Baselines

We provide the generation prompts used for the baseline below:

SVG Diagram Generation

System Prompt: You are a diagramming assistant. Output only a single (`<svg>...</svg>`) with the diagram. Do not include explanations or code fences.

User Prompt: {caption}
Convert this text into a flowchart. Output only the SVG code.

Mermaid Diagram Generation

System Prompt: You are an expert at creating Mermaid diagrams. Given a technical caption describing an architecture or system, generate a clear and accurate Mermaid diagram. Only output the Mermaid code without any explanation or markdown code blocks. Start directly with the diagram type (e.g., `graph TD`, `flowchart LR`, etc.)

User Prompt: Create a Mermaid diagram for the following description: {caption}

Image Generation

User Prompt: {caption}
Convert this text into a flowchart.

B. Evaluation Metrics Details

B.1. Rubric: Accuracy

A1. Coverage (are all main modules/stages listed in the caption shown?)

Level 3 – Complete coverage. Every main module/stage explicitly mentioned in the caption appears in the figure; no two distinct modules are merged; no extra modules contradict the caption; names align one-to-one between caption and figure.
Level 2 – Partial coverage. All main modules/stages are present, but readers may need to check the caption to confirm the mapping. Auxiliary pieces or examples may be absent, or the names in the figure differ from the caption while still maintaining a clear one-to-one correspondence
Level 1 – Missing/misleading. At least one main module/stage mentioned in the caption is missing, or two distinct modules are merged into a single element in a way that changes the structure, leading to misunderstanding of the overall structure.

A2. Relations and Direction (are connections/flow correct?)

Level 3 – Clear and complete. All key relations (data flow, calls, parallel/merge, containment, residual connections, etc.) are clearly expressed with arrows or connectors; directions are unambiguous; there are no dangling endpoints or other graphical elements that could be interpreted as incomplete links.
Level 2 – Readable but incomplete. The overall sequence or dependency is visible, but at least one relation needs to be checked against the caption to confirm its exact meaning; some connectors may be underspecified but do not directly contradict the caption.
Level 1 – Wrong/missing. Key relations are incorrect or omitted, or serial vs. parallel structure is confused. Any wrong or contradictory arrow direction, missing arrow that creates ambiguity, line running through text that causes uncertainty, or dangling endpoint suggesting an unfinished connection must be scored as Level 1.

A3. Terminology and Symbols Align With Caption (labels, spelling, abbreviations, math symbols)

Level 3 – Aligned and consistent. Labels, abbreviations, and math symbols in the figure match those in the caption; each concept uses a single, consistent name; there are no spelling mistakes in key terms; and case conventions are consistent when they carry meaning. None of the downgrade conditions for Level 1 or 0 apply.
Level 2 – Mappable. Different wording, alternative phrasing, or unexplained abbreviations are present but can still be mapped one-to-one and unambiguously to the terms in the caption. Wording may differ, but as long as the mapping remains clear and reliable, the Level should be 2 rather than 3.
Level 1 – Misaligned/misleading. Any mismatch that breaks reliable mapping between caption and figure (e.g., the same name used for different things, different names used for the same thing, or an incorrect symbol), or a mis-

spelling that prevents alignment, must be scored as Level 1. Labels that contradict the caption or confuse meaning also fall into this category.

B.2. Rubric: Beauty

B1. Layout and Grouping (hierarchy, sections)

Level 3 – Clear hierarchy. Functional grouping and stage/section boundaries are clearly present; the reading order is obvious. The layout is not a plain top-to-bottom stack when the caption implies multiple stages or modules.

Level 2 – Some hierarchy. Some grouping exists, but boundaries are weak or rely mainly on color without clear container edges or separators. The overall structure is still somewhat interpretable, though less explicit than at Level 3.

Level 1 – No hierarchy. Modules are merely stacked with little or no visible grouping, and relationships between parts are hard to judge. A pure single-column, ungrouped vertical stack when the caption indicates multiple modules or stages must also be scored as Level 1.

B2. Spacing and Edge Proximity (crowding, touching, occlusion)

Level 3 – Adequate whitespace. Elements have sufficient padding; no objects touch the borders of the figure; there is no overlap; and text is neither compressed nor occluded.

Level 2 – Compact but readable. Spacing is tight in places but all elements remain distinct; there is no overlap or edge contact that impairs legibility, and the figure is overall still readable.

Level 1 – Crowded/occluded. Crowding, overlap, or contact with the figure borders causes reading issues. Any element touching the edges, overlapping with another element, or text being partially occluded must be scored as Level 1.

B3. Line Organization (crossings, routing, endpoints)

Level 3 – Simple and clear. Endpoints land precisely on shapes; arrows are present and point correctly; lines avoid running through text; crossings use bends, bridges, or branch marks to avoid ambiguity. The routing is easy to follow and none of the severe problems listed for Level 1 occur.

Level 2 – Clear but somewhat messy. Crossings or unmarked bends exist and the layout is somewhat cluttered, but destinations remain traceable and unambiguous. There are no lines running through text, no dangling or clearly misaligned endpoints, and no missing or contradictory arrows; otherwise the case must be scored as Level 1.

Level 1 – Unclear routing. Routing is unclear: crossings lack branch marks, lines run across text, or destinations are ambiguous. Any line through text, dangling or misaligned endpoint, or missing/contradictory arrow direction also should be scored as Level 1.

B4. Text and Titles Readability (fonts, contrast, hierarchy)

Level 3 – Clear. Fonts render cleanly; size is adequate; foreground/background contrast is sufficient; titles and content are not cut off; and the visual hierarchy between titles and body text is evident.

Level 2 – Readable with a pause. Some small or low-contrast text, or proximity to lines and other shapes, causes minor effort to read, but all key labels remain legible and can be interpreted without guessing.

Level 1 – Unreadable. Key labels are fuzzy, too small, covered, or cut off, or contrast is insufficient so that text blends into the background. Blurry or occluded text in multiple places, or any critical label that cannot be reliably read, must be scored as Level 1.

B5. Color (contrast and functional differentiation)

Level 3 – Harmonious and clearly differentiated. The palette provides functional differentiation: different modules use clearly different colors, and each module keeps an internally consistent scheme that supports clarity. Color differences between modules are obvious and layered, foreground/background contrast does not hinder readability, and the overall appearance is harmonious.

Level 2 – Some interference. Color contrast is not very strong; some differences between modules exist but are not immediately obvious, requiring a brief pause to interpret. The palette may be simple or somewhat monotone, but modules can still be distinguished and internal colors within a module do not seriously disrupt reading. If modules cannot be reliably distinguished by color, or internal colors within a module are inconsistent to the point of confusion, the Level must be 1 instead.

Level 1 – Chaotic or uncoordinated. Colors are overly complex or poorly organized, with no clear differentiation of modules; different modules use the same or very similar colors so that roles cannot be clearly distinguished; or foreground/background contrast is insufficient and reading is affected. Palettes that are overly chaotic with no clear module differentiation, or so monotone that modules cannot be separated, as well as internally inconsistent colors within a module that obstruct understanding, must all be scored as Level 1.

B6. Containers and Grouping Visuals (boundaries, titles, alignment)

Level 3 – Clear and consistent. Each group or stage has a closed boundary; edges align cleanly; titles do not cover

content; and container styles (borders, fills, corners) are consistent across similar groups.

Level 2 – Minor flaws. Small alignment or closure issues exist (e.g., slight gaps or misalignments), but the extents of each container remain easy to judge and titles only minimally interfere with content.

Level 1 – Messy. Containers overlap one another, run through text or nodes, or have open or unclear boundaries such that their extents are hard to interpret. Boundaries that cross or cover internal text/nodes, or open containers where the intended limits are ambiguous, must be scored as Level 1.

Evaluation

System Prompt: You are a diagram scoring assistant. Apply the following rubrics to a given image: {Rubric A and B}. Start from Level 1 and increase only with clearly visible, direct evidence in the image. Prefer conservative scoring when uncertain; do not average issues or compensate across items. Caption is not a crutch: do not credit elements that appear only in the caption; the caption may only disambiguate labels/relations that are already visible.

User Prompt: Caption: {caption}.
Image to evaluate: {image}.
Respond ONLY with a JSON object of the form

$$\{"level" : [A_1, A_2, A_3, B_1, B_2, B_3, B_4, B_5, B_6]\}$$

and nothing else. Do not include any explanation, natural language, or extra keys.

B.3. Rubric: Completeness

Level 3. Completely correct and fully consistent with the reference caption.

Level 2. Mostly correct with minor omissions or slightly imprecise wording compared to the reference caption.

Level 1. Partially correct, loosely related, or containing contradictions with the reference caption; or completely wrong/not inferable.

We evaluate completeness via a two-step reverse-generation procedure. First, given a diagram, we ask the model to produce a textual description that summarizes what can be inferred from the diagram alone (modules, relations, assumptions, and key details). This yields a reverse-generated description of the image. Second, we compare this description with the original refined caption by feeding both texts, the model is instructed to judge how fully the diagram-based description recovers the content of the original caption and to output a single completeness level.

Evaluation

System Prompt: You should write detailed figure captions for technical images and diagrams. Use ONLY the provided image. Describe visible elements precisely: objects, labels, icons, arrows/connectors (direction), containers/grouping, and containment relationships. Preserve technical terminology and notation exactly as shown; use inline LaTeX-style forms if visible. Do NOT speculate or add information not clearly visible. No citations or meta commentary. Do NOT include a figure number. Output the caption paragraph only.

User Prompt: Write the caption of the image: {image}.

Evaluation

System Prompt: You are a strict grader. You should evaluate correctness and consistency of the candidate caption relative to the reference caption. Using the following rubric to evaluate: {Rubric C}.

User Prompt: Please compare the candidate caption: {candidate caption} with the reference caption: {reference caption}. Respond ONLY with a JSON object of the form

$$\{"level" : C\}$$

and nothing else. Do not include any explanation, natural language, or extra keys.

C. Paper2Figure Bench Construction

To construct high-quality textual descriptions for each figure in Paper2Figure Bench, we employed GPT-4o in a three-stage caption generation pipeline.

First, the model was prompted to produce a concise caption that focuses on the main idea and overall logical flow of the figure.

Second, the model generated a detailed caption that enumerates the key modules, components, and structural relations visible in the visual layout.

Finally, both versions were merged using a refinement prompt that preserves the organizational clarity of the concise caption while selectively incorporating only the essential details from the detailed one.

This procedure ensures that the resulting caption is semantically accurate, visually grounded, and stylistically consistent across all samples.

Concise Caption Prompt

You are given the full PDF of a scientific paper and a figure image extracted from that paper.
1. Locate the exact caption in the PDF that belongs to this figure.
2. Carefully read the caption and the nearby context that explains the figure.
3. Produce a **2-3 sentence concise caption** that explains what the figure conveys, focusing on the core idea and overall logical flow.
Return only the refined caption text.
{pdf} {image}

Detailed Caption Prompt

You are given the full PDF of a scientific paper and a figure image extracted from that paper.
1. Locate the exact caption in the PDF that belongs to this figure.
2. Carefully read the caption and the nearby context that explains the figure.
3. Produce a **detailed caption** that explains all key components and structural relations visible in the figure.
Return only the refined caption text.
{pdf} {image}

Merged Caption Prompt

You will receive two captions for the same scientific figure:
• Caption V1 is concise and captures the overall logical flow.
• Caption V2 is detailed but may include excessive specifics.
Write a **4-5 sentence refined caption** that preserves the structure and logical flow of V1 while integrating only the essential supporting details from V2 (such as key modules or major interactions). Avoid unnecessary graphical specifics, keep the wording precise, and ensure the caption is self-contained for readers who can see the figure but not the original paper.
Return only the refined caption text.
Caption V1: {concise caption}
Caption V2: {detailed caption}

Method	Vis.	Acc. ↑	Bea. ↑	Comp. ↑	Avg. ↑
Open-Source VLM (SVG Baseline)					
Qwen3-VL-8B-Thinking	SVG	41.0	44.0	35.0	40.0
Qwen3-VL-30B-A3B-Thinking	SVG	54.0	53.0	49.0	52.0
Qwen3-VL-235B-A22B-Thinking	SVG	56.0	55.0	52.0	54.3
GLM-4.6V	SVG	55.5	55.0	52.5	54.3
Paper2Figure (ours) w/ Open-Source VLM Backbones					
Paper2Figure (w/o Ref., Qwen3-VL-8B)	FigScript	55.0	56.0	54.0	55.0
Paper2Figure (**full**, Qwen3-VL-8B)	FigScript	59.0	60.5	60.0	59.8
Paper2Figure (w/o Ref., Qwen3-VL-30B)	FigScript	62.0	63.0	61.0	62.0
Paper2Figure (**full**, Qwen3-VL-30B)	FigScript	66.0	67.0	66.0	66.3
Paper2Figure (w/o Ref., Qwen3-VL-235B)	FigScript	64.0	65.0	63.0	64.0
Paper2Figure (**full**, Qwen3-VL-235B)	FigScript	69.0	69.5	68.0	68.8
Paper2Figure (w/o Ref., GLM-4.6V)	FigScript	63.5	64.5	63.0	63.7
Paper2Figure (**full**, GLM-4.6V)	FigScript	68.5	69.0	68.0	68.5

Table 3. **Results with open-source VLM backbones.** For each backbone, the SVG direct-generation result serves as the baseline. Acc./Bea./Comp. are category-wise means; Avg. is the mean of the three.

D. Additional Experiments

D.1. Open-Source VLM Backbones

To evaluate whether the performance gains of Paper2Figure stem from the multi-agent design and FigScript representation rather than reliance on a specific proprietary model, we conduct additional experiments with four representative open-source VLMs as backbone: Qwen3-VL-8B-Thinking, Qwen3-VL-30B-A3B-Thinking, Qwen3-VL-235B-A22B-Thinking, and GLM-4.6V. We compare each open-source backbone under the SVG direct-generation baseline and under our full Paper2Figure system.

As shown in Table 3, Paper2Figure with open-source backbones consistently and substantially outperforms the corresponding open-source SVG baselines across all evaluation dimensions. This indicates that the performance gains of Paper2Figure primarily arise from the dual multi-agent design and the FigScript representation rather than from any specific proprietary model. While GPT-4o achieves the best absolute performance, open-source models remain practical and effective alternatives, demonstrating the model-agnostic nature of our framework.

D.2. Runtime and Cost Analysis

We conduct a token-based cost and runtime analysis on Paper2Figure Bench, using the same user prompt for all methods to ensure fair comparison. Runtime is approximated by the total token count (input + output) relative to the SVG baseline. Table 4 presents the results.

Paper2Figure (full) achieves a 33.8% improvement in Avg. score over the GPT-4o SVG baseline at 1.90× the cost and 3.62× the relative runtime overhead. The generation-only variant (Paper2Figure w/o Ref.) already attains a 26.4% gain at only 1.22× the cost, offering an efficient alternative when latency or budget is constrained.

Method	Model	In	Out	FigScript out	Cost ($)	vs. SVG	RT (rel.)	Avg.
SVG baseline	GPT-4o	96	2985	–	0.0301	1.00×	1.00×	59.2
Mermaid baseline	GPT-5	104	326	–	0.0034	0.12×	0.14×	51.3
Image gen. baseline	GPT-Image-1	112	1607 (img)	–	0.0648	2.15×	–	34.9
Paper2Figure (w/o Ref.)	GPT-4o	3128	2874	968	0.0366	1.22×	1.95×	74.8
Paper2Figure (**full**)	GPT-4o	7228	3922	1174	0.0573	1.90×	3.62×	79.2

Table 4. **Token-based cost and runtime proxy.** Runtime (rel.) is computed as (In+Out) normalized by (In+Out) of the SVG baseline. Cost is estimated from token counts using published API pricing.

D.3. Human-in-the-Loop Refinement

Paper2Figure supports human-in-the-loop refinement through two mechanisms: (1) natural-language instructions issued via the web interface to trigger another agent-based refinement pass, and (2) direct in-browser editing of individual elements. To quantify the benefit of interactive control, we evaluate a human-assisted setting in which a human provides refinement instructions, the agents execute the revisions, and the human applies final local adjustments. This setting achieves an Avg. score of 83.0, representing a +4.8% improvement over the fully automatic Paper2Figure (full) variant (79.2). The result demonstrates that the web interface enables users to efficiently steer and fine-tune figures to better match their intent, while the automated pipeline already provides a strong starting point.

D.4. Cross-Domain Generalization

Paper2Figure Bench is constructed from 100 Computer Science papers, focusing on structured pipeline figures for controlled evaluation of semantic faithfulness and editability. To validate that the approach generalizes beyond this domain, we randomly sample 100 cases from FlowVQA [29], a multi-category flowchart benchmark spanning domains outside CS. Since FlowVQA is not designed for direct figure quality assessment, we apply the same Paper2Figure Bench evaluation metrics for fair comparison.

Under the same GPT-4o setting, the GPT-4o SVG baseline achieves an Avg. score of 54.0, while Paper2Figure (w/o Refinement) and Paper2Figure (full) achieve 76.5 and 79.4, respectively, corresponding to gains of +41.7% and +47.0% over the SVG baseline. These results confirm that Paper2Figure generalizes across domains, as its performance advantage stems from structural reasoning and FigScript representation rather than CS-specific knowledge.

D.5. Sensitivity to Instruction Granularity

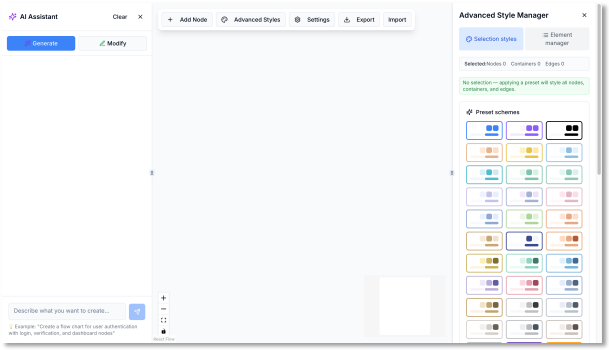
We analyze how Paper2Figure responds to varying levels of instruction detail by testing three prompt types: (1) *low-structure*: a short high-level paragraph describing the method (abstract-style); (2) *default*: the caption-style instruction used in Paper2Figure Bench; and (3) *high-structure*: a detailed specification with explicit module lists, ordering, grouping, and layout constraints. Performance across the three settings is 78.7 / 79.2 / 79.6 (Avg.), indicating that the system does not require highly detailed specifications. Coarse descriptions already yield strong drafts, and full specifications provide only marginal additional gains.

E. More details about Paper2Figure

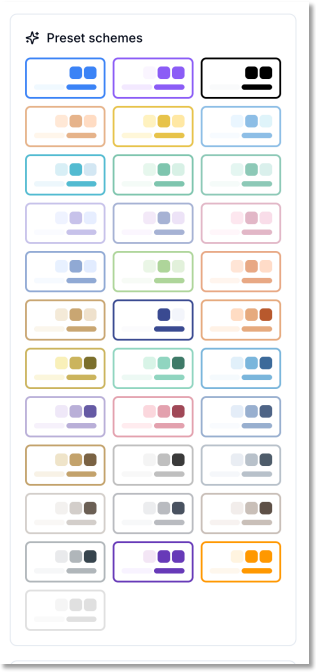
For reproducibility, we clarify the design principles of our agent system. The exact prompting templates used for the Generation and Refinement Agents are not released, as they contain implementation details tied to our internal workflow. However, the functional roles and interaction patterns of all agents are fully described in Section 2.1, and the system guarantees that all agent outputs conform to the FigScript specification introduced in the main text.

FigScript is a Mermaid-inspired declarative language for scientific diagrams, extending the Mermaid syntax with richer controls over node geometry, spacing, hierarchical grouping, arrow styles, color schemes, and multiple layout algorithms. These extensions enable FigScript to encode both the semantic structure and the visual design choices necessary for producing coherent and publication-quality figures. Renderer executability is enforced at 100% through an automated error-feedback loop: any render error, together with the original script, is passed back to the Layout and Edit Agents to fix iteratively until the script renders without errors, ensuring that all FigScript outputs produced by Paper2Figure are valid and fully renderable. Currently, FigScript is optimized for pipeline-style and structured conceptual diagrams; figures requiring image-like elements (e.g., photographs, pixel-precise plots) are not natively supported, and we plan to integrate an image-generation module and extend layout support to multi-panel and non-graph structures in future work.

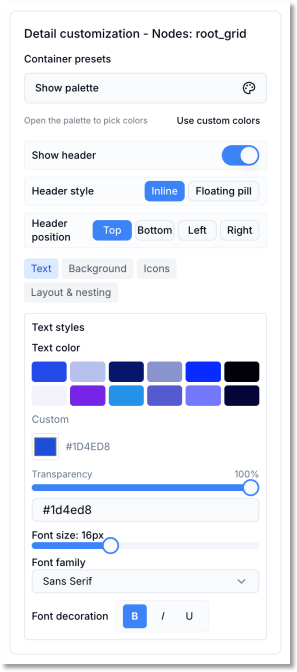
As shown in Figure 6, the Paper2Figure Web Editor provides an interactive environment that allows users to inspect, adjust, and refine figures generated by the agent system. The main workspace (a) supports real-time rendering of FigScript specifications, enabling users to directly manipulate nodes, containers, and layout structures. To promote visual consistency and reduce the burden of manual styling, the editor offers a rich set of color presets (b) that can be applied to modules or the entire figure. In addition to preset schemes, users may access fine-grained customization controls (c), including font families, text colors, container shapes, header styles, spacing, and other presentation parameters. These capabilities allow users to either rely on standardized templates for rapid creation or override individual settings for detailed tuning, making the editor both efficient and highly flexible for producing publication-quality figures.



a. Paper2Figure Web Editor



b. Color Presets



c. Element Editor

Figure 6. Interface of the Paper2Figure Web Editor. (a) The main editor workspace; (b) built–in color presets for consistent visual styling; (c) fine-grained element customization for nodes, containers, and text.