

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

Siwei Han^{1*} Haonian Ji^{1*} Siyang Xin^{1*} Juanquan Shi¹ Shi Qiu¹ Xinyu Ye¹
 Peng Xia¹ Jiaqi Liu¹ Zhaorun Chen² Yiyang Zhou¹ Linjie Li³ Lijuan Wang³ Huaxiu Yao^{1†}

¹UNC-Chapel Hill ²University of Chicago ³Microsoft

siweih@cs.unc.edu, huaxiu@cs.unc.edu

*Equal contribution. †Corresponding author.

Abstract

Automatically generating clear and accurate figures for research papers remains challenging, as it requires semantic understanding, precise structure, and visual aesthetics. Existing approaches struggle to balance fidelity and quality: large language model (LLM) code-based methods (e.g., SVG, Mermaid) are structured but inflexible, while image-generation models (e.g., GPT-Image-1, Nano Banana) produce hard-to-edit and often inaccurate figures. We present Paper2Figure, a dual multi-agent system with an interactive web platform for paper-to-figure generation. Generation Agents convert text into our designed FigScript language, encoding figure semantics, styles and layout. The web system renders the FigScript into an initial image, which Refinement Agents iteratively analyze to locate issues and revise the FigScript for improved logic, alignment, aesthetics and text accuracy. Crucially, users can further refine results through an intuitive web interface, ensuring full control over the final output. To evaluate Paper2Figure, we introduce Paper2Figure Bench, a benchmark comprising 100 academic figures with paired descriptions. Experiments demonstrate that Paper2Figure markedly improves accuracy by 12%, beauty by 13.5%, and completeness by 17.0% over state-of-the-art baselines in fully automatic generation without human adjustment. By combining automated generation with interactive edit, Paper2Figure bridges the gap between AI assistance and researcher control, offering a practical solution for high-quality academic figure creation.

1. Introduction

Effective visual communication is central to scientific research. Concepts are far easier to comprehend through structured diagrams than textual descriptions [7, 10]. Consequently, main figures in academic papers, such as process

diagrams, conceptual flowcharts, and system overviews, serve as indispensable tools for explanation, and dissemination. However, creating such figures remains a labor-intensive and expertise-demanding task: researchers must translate nuanced textual logic into visually coherent, aesthetically balanced, and publication-ready graphics.

Recent advances in generative AI have inspired two main directions for automatic figure generation, each with fundamental limitations. The first relies on large language models (LLMs) to produce structured code, such as Scalable Vector Graphics (SVG) [5, 6, 24, 25, 37] or markup languages like Mermaid [37]. These approaches provide semantic clarity and editability but often yield messy layouts and poor aesthetics, producing figures that are structurally correct yet visually unappealing. The second direction leverages powerful text-to-image models [16, 26, 30–32, 36, 41], including GPT-Image-1 [23] and Nano Banana (Gemini 2.5 Flash Image) [9], to synthesize richly styled figures. Despite their visual appeal, these models struggle with layout accuracy, textual rendering, and editability, making their outputs impractical for scientific use [4, 14, 17, 34]. As a result, no existing method can jointly ensure semantic precision, visual quality, and flexible structure control.

We propose **Paper2Figure**, a dual multi-agent system [11, 38] with an interactive web platform for automatic paper-to-figure generation. Paper2Figure introduces a structured representation language, **FigScript**, designed to capture figure semantics, various styles, and layout relations. Generation Agents interpret textual descriptions and generate the initial FigScript, which the web system renders into a preliminary image. Refinement Agents then analyze this image to identify inconsistencies and revise the underlying FigScript, improving logical relations, alignment, aesthetics, and text accuracy. After auto generation, users can manually fine-tune figures through the web interface or guide Refinement Agents to edit if desired. This pipeline effectively combines accurate understanding of textual in-

put, faithful and aesthetically optimized visual realization, and user-level flexibility within a unified framework.

To evaluate Paper2Figure, we present **Paper2Figure Bench**, a benchmark of 100 academic figures paired with textual descriptions. Experimental results show that Paper2Figure substantially outperforms state-of-the-art baselines in accuracy (by 12.0%), beauty (by 13.5%) and completeness (by 17.0%), demonstrating that collaborative agent design and structured figure representation together enable a new level of clarity and quality in automated scientific visualization.

Our contributions are threefold: (1) We propose Paper2Figure, a dual-agent system that unifies semantic reasoning, visual refinement, and flexible editing through a structured intermediate language, FigScript, and implement it as an interactive web platform; (2) We introduce Paper2Figure Bench, a benchmark for evaluating text-to-figure generation under realistic academic scenarios; (3) We demonstrate that Paper2Figure achieves superior accuracy, beauty and completeness compared with leading LLM- and diffusion-based baselines, advancing the automation of scientific figure creation.

2. Paper2Figure

Paper2Figure is a dual multi-agent framework with an interactive web platform for generating publication-quality scientific figures from textual descriptions, as illustrated in Figure 1. We first design an intermediate language FigScript, which encodes figure semantics, styles, and layout. Textual descriptions are first interpreted by the Generation Agents and translated into FigScript, which is then rendered into an initial image. The Refinement Agents subsequently analyze this image, identify potential issues, and revise the underlying FigScript for logical correctness, alignment, and aesthetics before producing the final refined figure. This design ensures accurate understanding of input text, faithful and visually appealing realization, and robust handling of complex layouts, while also supporting optional user adjustments through the web interface.

2.1. Paper2Figure Dual Multi-agent System

Paper2Figure employs a dual multi-agent architecture that mirrors the workflow of professional scientific illustrators. It consists of two cooperative subsystems, the *Generation Agents* and the *Refinement Agents*, which interact through the intermediate representation language FigScript. All agents are implemented using GPT-4o, ensuring consistent reasoning capability and high-quality language-vision understanding across the entire pipeline. This design enables the system to progressively construct and improve figures while maintaining consistency. Each agent specializes in a distinct aspect of figure creation, ensuring both logical accuracy and aesthetic optimization across stages. We formal-

Algorithm 1 Dual Multi-Agent Workflow

```

1: Input: User instruction  $I$ 
2: Output: Final FigScript  $S_{\text{final}}$  and rendered image  $F_{\text{final}}$ 
3: Hyperparameter: Number of refinement iterations  $N$  (default: 1)

# — Stage 1: Generation —
4:  $P \leftarrow \text{PlanAgent.run}(I) \triangleright$  Extract entities and create a high-level plan
5:  $S_{\text{draft}} \leftarrow \text{ModuleAgent.run}(P) \triangleright$  Construct visual modules in FigScript
6:  $S_{\text{initial}} \leftarrow \text{LayoutAgent.run}(S_{\text{draft}}) \triangleright$  Refine alignment, routing, and spacing
7:  $F_{\text{initial}} \leftarrow \text{render}(S_{\text{initial}}) \triangleright$  Render the initial figure from the FigScript

# — Stage 2: Refinement —
8:  $S_{\text{current}} \leftarrow S_{\text{initial}}$ 
9:  $F_{\text{current}} \leftarrow F_{\text{initial}}$ 
10: for  $i = 1$  to  $N$  do
11:    $C \leftarrow \text{CriticAgent.run}(F_{\text{current}}) \triangleright$  Identify visual issues in the current image
12:   if  $C$  is empty then
13:     break  $\triangleright$  No further refinements needed
14:   end if
15:    $R \leftarrow \text{RefineAgent.run}(C) \triangleright$  Formulate a structured revision plan
16:    $S_{\text{current}} \leftarrow \text{EditAgent.run}(S_{\text{current}}, R) \triangleright$  Execute plan and update the FigScript spec
17:    $F_{\text{current}} \leftarrow \text{render}(S_{\text{current}}) \triangleright$  Re-render the figure for the next iteration's review
18: end for
19:  $S_{\text{final}} \leftarrow S_{\text{current}}$ 
20:  $F_{\text{final}} \leftarrow F_{\text{current}}$ 
21: return  $S_{\text{final}}, F_{\text{final}}$ 

```

ize the collaborative process of our dual multi-agent system in Algorithm 1.

In the generation stage, three agents collaborate to transform textual descriptions into an executable FigScript specification. The *Plan Agent* first analyzes the input user instruction to extract entities, processes, and logical relationships, then formulates a high-level plan that defines the figure’s modular structure, flow direction, and hierarchical organization. The *Module Agent* subsequently implements this plan by constructing visual modules in FigScript, encoding nodes, edges, containers, and labels that represent the key semantic units of the description. The *Layout Agent* then refines the spatial organization by adjusting alignment, edge routing, spacing, and grouping, aiming for a compact and visually balanced composition. The outcome of this stage is a complete FigScript file that can be deterministi-

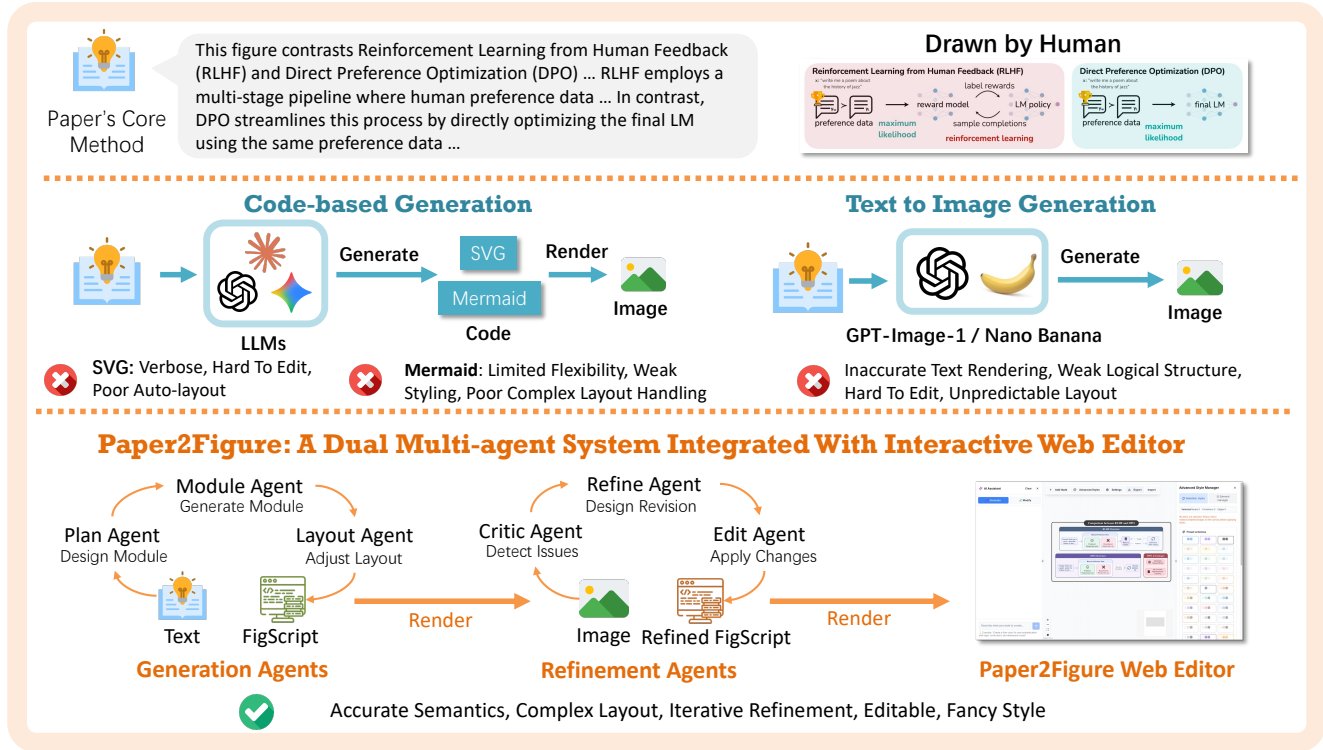


Figure 1. Overview of the Paper2Figure framework, comparing traditional code-based (SVG/Mermaid) and text-to-image methods with our dual multi-agent system and interactive web editor that collaboratively generate, refine, and edit scientific figures.

cally rendered into an initial figure image, serving as the input to the refinement process.

The refinement stage focuses on self-correction and visual enhancement. It begins with the *Critic Agent*, which examines the rendered image to detect issues such as misplaced modules, inconsistent text alignment, color imbalance, or visual clutter that reduce readability. The *Refine Agent* interprets these detected issues and formulates a structured revision plan that specifies modifications to the underlying FigScript, including positional adjustments, text corrections, layout restructuring, and style harmonization. The *Edit Agent* then executes this plan by applying the corresponding updates to FigScript and triggering a new rendering cycle. This closed feedback loop ensures that the figure converges toward both semantic precision and aesthetic quality. For efficiency, the refinement stage is executed once by default; our evaluation reports results for both the Generation-only configuration and the full Generation-plus-Refinement pipeline.

Through coordinated operation between these agents, Paper2Figure achieves accurate understanding and faithful visual realization of textual input, producing figures that are not only structurally coherent but also aesthetically optimized. The multi-agent collaboration, grounded in the structured FigScript representation and specifically

designed system support, enables the system to handle complex academic diagrams with clarity, consistency, and flexibility that existing approaches cannot achieve.

2.2. FigScript: Structured Intermediate Language

FigScript is a structured language that bridges textual semantics and graphical realization. FigScript defines each figure as a hierarchical graph composed of nodes, edges, containers, and styling attributes. It captures both the logical structure and visual layout of a figure, serving as the communication medium among agents and between the backend and the web interface.

In terms of complexity, FigScript is positioned between traditional vector graphics languages such as SVG and high-level declarative formats like Mermaid. While it inherits the precise controllability of SVG, it abstracts many low-level drawing operations into semantically meaningful constructs, allowing agents to reason about figures at the conceptual level. At the same time, unlike Mermaid’s fixed syntax and limited expressiveness, FigScript provides a rich set of adjustable parameters that enable fine-grained customization under consistent design rules.

Each element in FigScript supports extensive configurable attributes, including colors, font styles, border thickness, arrow shapes, container padding, and icon styles. The

layout system further incorporates multiple automatic arrangement algorithms that can be selected or tuned through parameters. These flexible yet constrained design primitives allow agents to balance structural accuracy and visual harmony while ensuring that the resulting figures remain editable and stylistically coherent.

2.3. Paper2Figure Web Editor

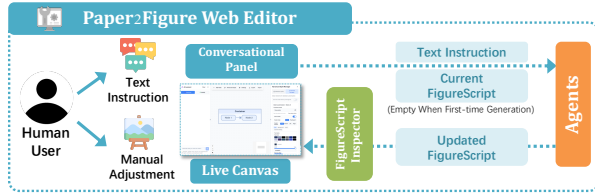


Figure 2. Paper2Figure Web Editor supports text instructions to agents for generation and editing, and allows user to manually adjust on a live canvas.

As shown in Figure 2, Paper2Figure Web Editor provides an integrated environment that connects automated agent generation with user-driven control. It visualizes the figures created by the agents and allows users to generate, modify, and refine them through a combination of natural-language instructions and direct manipulation. The editor consists of three major components: a *Conversational Panel*, a *Live Canvas*, and a *FigureScript Inspector*. Users can describe a desired figure in natural language via the Conversational Panel, which passes the instruction to the Generation Agents together with the current (possibly empty) FigScript state. The resulting FigScript is rendered in real time on the Live Canvas.

Beyond generation, users can issue language-based editing commands such as “move the preprocessing module to the left” or “rename ‘Input Data’ to ‘Raw Dataset’.” The editor forwards the command and the current FigScript to the Refinement Agents, which update the FigScript accordingly and render the modified figure. Users can also directly adjust any visual element through the Live Canvas. Through this tight integration, Paper2Figure enables accurate text-to-figure generation, agent-assisted refinement, and flexible human adjustment within a unified pipeline. While the agents provide semantic understanding and aesthetic optimization, the editor offers users transparent control over both structure and style, supporting efficient creation of clear, consistent, and visually compelling scientific figures.

3. Paper2Figure Bench

3.1. Data collection and captioning

We collected approximately five hundred research papers published on arXiv within the past two years, downloading both their PDF and source files. From these papers, we

manually selected those containing relatively complex main figures that illustrate complete methods or system pipelines. The selection was based on two key criteria: *importance*, meaning that the figure represents the core contribution of the paper, and *clarity*, referring to the completeness and readability of its visual structure. After careful screening, we retained one hundred representative figures with clear content and well-defined layouts.

Each selected figure, together with its source paper, was provided to GPT-4o for automatic caption generation through a three-step process. The model first produced a concise caption emphasizing the overall logic and core idea of the figure, then generated a detailed caption describing key components and structural relations of the figure. Finally, the two versions were merged by prompting GPT-4o to preserve the clear organization of the concise version while integrating only essential details, such as module names and key connections.

All captions were manually reviewed and standardized by annotators to ensure accuracy and consistency. Each was cross-checked with the figure and paper to verify step order, terminology, and information flow. Through this semi-automatic and human-verified process, Paper2Figure Bench achieves high semantic fidelity, logical consistency, and stylistic uniformity.

3.2. Evaluation Metrics

We evaluate all models using a rubric-based framework consisting of three equally weighted dimensions: Accuracy, Beauty, and Completeness. Accuracy comprises three sub-criteria (A1 module coverage, A2 consistency of relations and directions, and A3 alignment of terminology and symbols). Beauty includes six sub-criteria (B1 layout and grouping, B2 whitespace and margins, B3 line organization and endpoints, B4 text readability, B5 color distinction, and B6 background boundaries). Completeness contains a single sub-criterion (C) assessing the logical consistency and completeness between the caption generated from the image and the original caption. Each sub-criterion is independently scored on a three-level scale of 0% (not satisfied), 50% (partially satisfied), and 100% (fully satisfied). The score for each dimension is computed as the mean of its sub-criteria, and the overall figure score is obtained by averaging the three dimension scores and linearly rescaling the result to a 0–100 range. Critical issues such as blurred rendering, misleading arrow directions, or missing essential information result in automatic downgrading of the affected sub-criterion.

Figure 3 shows the evaluation pipeline of Paper2Figure Bench. We employ an LLM-as-a-Judge evaluation strategy using GPT-4o guided by detailed rubrics across three dimensions: Accuracy, Beauty, and Completeness. Accuracy measures the correctness of module coverage, directional

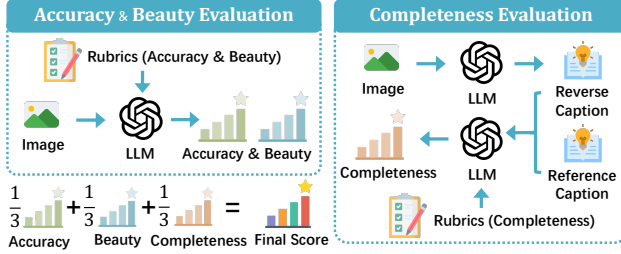


Figure 3. Evaluation pipeline of our rubric-based framework. GPT-4o scores Accuracy and Beauty directly from the image, and evaluates Completeness by generating and comparing a visually inferred caption with the reference one. The three scores are equally weighted to produce the final result.

relationships, and terminology consistency. Beauty evaluates visual quality through layout organization, spacing, alignment, text readability, color differentiation, and boundary clarity. Both dimensions are assessed directly in a single evaluation round where GPT-4o scores all sub-criteria based on explicit rubrics. Completeness requires a two-round evaluation to assess semantic sufficiency. In the first round, GPT-4o generates a caption purely from the figure’s visual content, without reference to the original description. In the second round, this generated caption is compared against the reference caption to determine whether the figure fully and correctly expresses the intended logic and information. This process detects issues such as missing steps, hallucinated content, incorrect ordering, or semantic inconsistencies. Together, this rubric-based, LLM-assisted framework provides consistent, interpretable, and reproducible assessments of semantic precision, visual quality, and informational completeness.

4. Experiment

Our experiments comprehensively evaluate the proposed Paper2Figure framework and benchmark it against multiple baseline approaches for text-to-figure generation. We aim to answer the following key questions: (1) How does Paper2Figure’s auto generation perform compared with existing code-based and image-based baselines in terms of accuracy, beauty, and completeness? (2) What is the contribution of the Refinement Agents, and how much improvement does the full Paper2Figure system achieve over its generation-only variant? (3) How consistent are our rubric-based automatic evaluation metrics with human judgments? and (4) What insights can be drawn from case studies comparing the outputs of baselines and Paper2Figure?

To address these questions, we evaluate three categories of baseline models alongside two variants of Paper2Figure (with and without refinement), apply our proposed scoring framework to all generated figures, analyze metric–human

agreement, and conduct detailed qualitative comparisons through representative case studies.

4.1. Experiment Setup

Baseline Models. We evaluate three categories of baseline models, covering both code-based and image-based generation paradigms. (1) The first group consists of LLMs that directly generate **Scalable Vector Graphics (SVG)** code from text descriptions, including GPT-o3 [22], GPT-4o [20], GPT-5 [21], Claude 4.5 Sonnet [2], Claude Opus 4 [3], Gemini 2.5 Flash, and Gemini 2.5 Pro [9]. (2) The second group includes LLMs that generate figures in the **Mermaid** language format, namely GPT-5 [21] and Claude 4.5 Sonnet [2]. (3) The third group covers **text-to-image** models that synthesize figures directly as raster images, including GPT-Image-1 [23] and Nano Banana (Gemini 2.5 Flash Image) [9].

We further evaluate two variants of our proposed Paper2Figure. The first variant, Paper2Figure (w/o Refinement), uses only the Generation Agents to produce figures from text. The second, Paper2Figure (full), incorporates both Generation and Refinement Agents, forming the complete system. All methods and baselines are tested on the Paper2Figure Bench dataset described in Section 3, using the same textual inputs for fair comparison.

Evaluation Metrics. We apply the scoring strategy introduced in Section 3.2, evaluating each model across Accuracy, Beauty, and Completeness, and report all results on a unified 100-point scale to ensure consistent comparison.

Human Annotators. To further validate the reliability of the automatic metric, we randomly sampled 100 generated figures and asked two independent annotators to provide human ratings. The final human score is computed as the average of the two annotators’ evaluations. We then measure the correlation between human judgments and the automatic metrics, including our proposed metric, BERTScore, and the F1 score, demonstrating strong alignment between human and model-based evaluations.

4.2. Main Results

Figure 4 shows the quantitative results across the three evaluation dimensions. Overall, Paper2Figure consistently outperforms all baseline models in both accuracy and visual quality, achieving an average improvement of 14.1% over the strongest baseline. The baselines reveal distinct limitations across different generation paradigms. Table 1 further presents the scores of all models across each individual evaluation sub-criterion.

Large language models that generate SVG or Mermaid code display similar structural tendencies but differ in the nature of their weaknesses and overall performance. Models generating SVG code, such as GPT-5, Claude 4 Opus, and Gemini 2.5 Pro, achieve relatively strong performance,

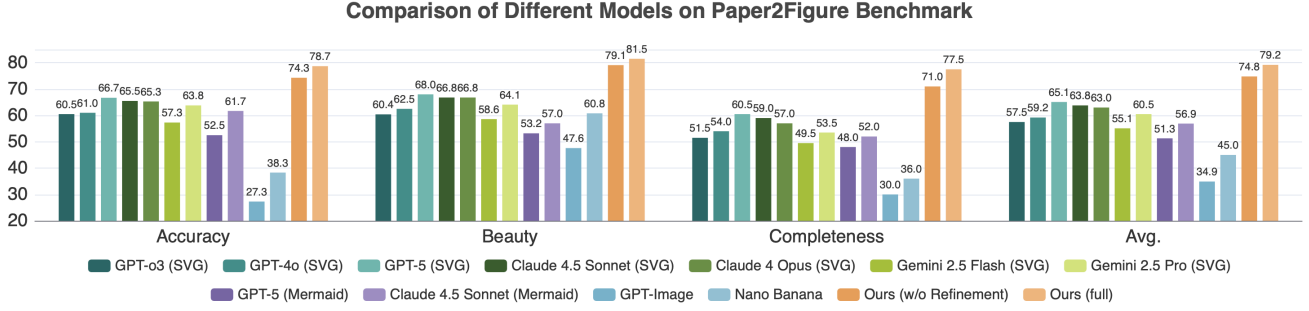


Figure 4. Comparison of different models on Paper2Figure benchmark, showing the average scores for Accuracy, Beauty, Completeness, and the overall mean. Paper2Figure(full) clearly outperforms code-generating LLMs and image-generation models across all dimensions.

Table 1. Detailed quantitative results for all models on each evaluation sub-criterion.

Method	Vis. Type	Accuracy			Beauty						Completeness C	Avg.
		A1	A2	A3	B1	B2	B3	B4	B5	B6		
Large Language Model (SVG)												
GPT-o3 [22]	SVG	78.0	52.5	51.0	63.5	65.5	51.0	52.5	78.5	51.5	51.5	57.5
GPT-4o [20]	SVG	79.0	52.5	51.5	66.5	67.0	51.5	54.0	82.5	53.5	54.0	59.2
GPT-5 [21]	SVG	81.5	58.5	60.0	72.5	71.5	60.0	57.0	89.5	57.5	60.5	65.1
Claude 4.5 Sonnet [2]	SVG	82.0	58.5	56.0	72.0	71.0	55.0	58.0	86.5	58.0	59.0	63.8
Claude Opus 4 [3]	SVG	80.5	57.5	58.0	74.0	70.5	55.0	56.0	87.0	58.0	57.0	63.0
Gemini 2.5 Flash [9]	SVG	75.5	49.0	47.5	61.5	63.5	51.0	48.0	77.5	50.0	49.5	55.1
Gemini 2.5 Pro [9]	SVG	80.5	54.5	56.5	69.0	69.0	54.0	55.0	83.0	54.5	53.5	60.5
Large Language Model (Mermaid)												
GPT-5 [21]	Mermaid	44.5	54.5	58.5	39.0	37.5	72.5	56.0	41.0	73.5	48.0	51.3
Claude 4.5 Sonnet [2]	Mermaid	62.5	58.5	64.0	42.5	43.5	72.5	57.5	52.0	74.0	52.0	56.9
Image Generation Model												
GPT-Image-1 [23]	Image	30.5	23.5	28.0	50.5	29.5	49.5	50.5	71.5	34.0	30.0	34.9
Nano Banana [9]	Image	40.5	34.5	40.0	52.5	66.5	52.5	54.0	72.5	66.5	36.0	45.0
Paper2Figure (ours)												
Paper2Figure (w/o Refinement)	FigScript	73.0	69.0	81.0	83.0	77.5	82.5	77.0	76.5	78.0	71.0	74.8
Paper2Figure (full)	FigScript	77.5	74.5	84.0	82.5	80.0	83.5	81.5	80.0	81.5	77.5	79.2

with an average Accuracy of around 65.3%, Beauty of 66.3%, and Completeness of 57.0%. Their key advantage lies in faithfully preserving textual and structural details from the input, which maintains semantic coverage. However, this fidelity often leads to visual clutter: overlapping elements, broken or misaligned lines, and excessively small text make the logic difficult to read. Floating nodes and disconnected components further disrupt coherence, resulting in figures that are information-dense but hard to interpret. In contrast, models generating Mermaid code, including GPT-5 and Claude 4.5 Sonnet, achieve notably lower overall performance, approximately 57.1% in Accuracy, 55.1% in Beauty, and 50.0% in Completeness. Their diagrams suffer from overly rigid, homogeneous layouts caused by the fixed positioning rules of the Mermaid renderer. While the structure appears clean, it lacks visual hierarchy and flexi-

bility, producing schematic and monotonous diagrams with limited color diversity and insufficient detail.

Image generation models, such as GPT-Image-1 and Nano Banana, produce visually appealing textures but fail to capture the logical structure required for scientific illustration. The generated figures often contain distorted text, spelling errors, missing modules, and disordered arrow connections, resulting in inconsistent information flow and low accuracy (32.8%). Among them, Nano Banana exhibits slightly richer visual styling. In contrast, GPT-Image-1 frequently suffers from incomplete boundaries and cropped content near the edges, further compromising diagram readability. Both models achieve limited completeness around 33.0% due to the absence of controllable layout and symbolic reasoning capabilities.

In contrast, our proposed Paper2Figure achieves the

highest performance across all dimensions and sub-criteria. It surpasses the strongest baseline by 14.1% in overall score and achieves balanced improvements in semantic precision, visual composition, and information completeness. The Paper2Figure(w/o Refinement) variant already demonstrates strong structural alignment and coherent layout, scoring 74.3% in Accuracy, 79.1% in Beauty, and 71.0% in Completeness. The full version of Paper2Figure with both Generation Agents and Refinement Agents, further enhances figure clarity and visual harmony, achieving 79.2% in overall average. Refinement Agents significantly improve module organization, text alignment, and color balance, while also yielding a moderate gain in completeness.

4.3. Agreement Between Metrics and Humans

To verify the reliability of our automatic evaluation framework, we conducted a consistency analysis between model-based metrics and human judgment. We randomly selected 100 generated figures from all evaluated systems and asked two independent annotators with relevant research experience to rate each figure based on the same rubric used in our automated evaluation. The final human score for each sample was computed as the average of the two annotators' ratings. We then compared these human scores with three automatic metrics: BERT, F1, and our proposed LLM-based scoring metric. Both BERT and F1 scores were calculated by comparing captions generated through GPT's visual inference for the figure with the reference captions.

Table 2. Comparison of different automatic scoring methods in terms of their agreement with human judgments.

Methods	Cosine $\uparrow$	Pearson $\uparrow$	Spearman $\uparrow$
BERT	0.7863	0.6337	0.5758
F1	0.7847	0.5608	0.5955
Paper2Figure Bench	0.8652	0.7345	0.7289

As shown in Table 2, our metric exhibits the strongest correlation with human preferences across all similarity measures. It achieves a cosine similarity of 0.8652, Pearson correlation of 0.7345, and Spearman correlation of 0.7289, outperforming both BERTScore and F1 by a notable margin. This result demonstrates that the LLM-as-a-judge approach, guided by detailed rubrics, effectively captures human evaluative criteria.

These findings confirm that our scoring framework not only provides interpretable and fine-grained assessments but also aligns closely with human perceptions of figure quality, supporting its reliability for large-scale automatic evaluation of text-to-figure generation systems.

4.4. Case Study

Figure 5 offers a qualitative comparison across generation paradigms, reveals clear shortcomings in the baselines. For

LLMs generating SVG code, GPT-5 preserves most textual details and largely maintains the correct RLHF-DPO logic, yet the figure contains numerous visual defects, including malformed arrowheads, text spilling outside bounding boxes, and frequent collisions between labels, borders, and connector lines. Gemini 2.5 Pro and Claude 4 Opus similarly capture the overall process structure, but Gemini leaves the Example Prompt node entirely unconnected, making its role unintelligible, while Claude displays overlapping text and frames and an overly monotonous layout. LLMs generating Mermaid code exhibit a different failure mode: GPT-5's Mermaid output is visually unappealing and overly rigid, with an unattractive color scheme and, more critically, major omissions such as failing to show how preference pairs train DPO, along with a structural error that incorrectly links the RLHF endpoint to the DPO Final LM. Claude 4.5 Sonnet contains more complete information but still suffers from drab colors and a flat, overly uniform arrangement. Image-generation models show the weakest structural fidelity: GPT-Image-1 includes distorted and misspelled text, incorrect arrow connections—including a prominent self-loop from the LM Policy back into "Reinforcement Learning"—and incomplete image boundaries, while Nano Banana, despite a seemingly polished layout, introduces severe logical inaccuracies, most notably depicting "Directly Optimize Final LM" as an input to "Reparameterize Reward Model", contradicting the DPO formulation.

In contrast, our method produces outputs that are both semantically accurate and visually coherent: the initial figure achieves complete logical coverage, balanced spacing, and harmonious color usage, and the refinement step further adds missing arrow labels, removes redundant nodes such as the unnecessary "DPO Policy Optimization" element in "DPO Advantages" container, and reorganizes the layout into a more compact and professional composition, resulting in a high-quality diagram that faithfully represents the RLHF and DPO pipelines.

It is noteworthy that part of these advantages come from Paper2Figure's built-in layout and color templates, which allow the agents to construct figures using standardized visual compositions and predefined style keywords. This design reduces the need for excessive custom element specification while still enabling fine-grained adjustments when the agents decide to override default templates. Consequently, Paper2Figure combines structural precision, aesthetic consistency, and flexible controllability, surpassing both code-based and image-based baselines.

5. Related Work

5.1. Figure Generation by LLMs

Recent years have seen rapid progress in generating visual content directly from text, driven by large language mod-

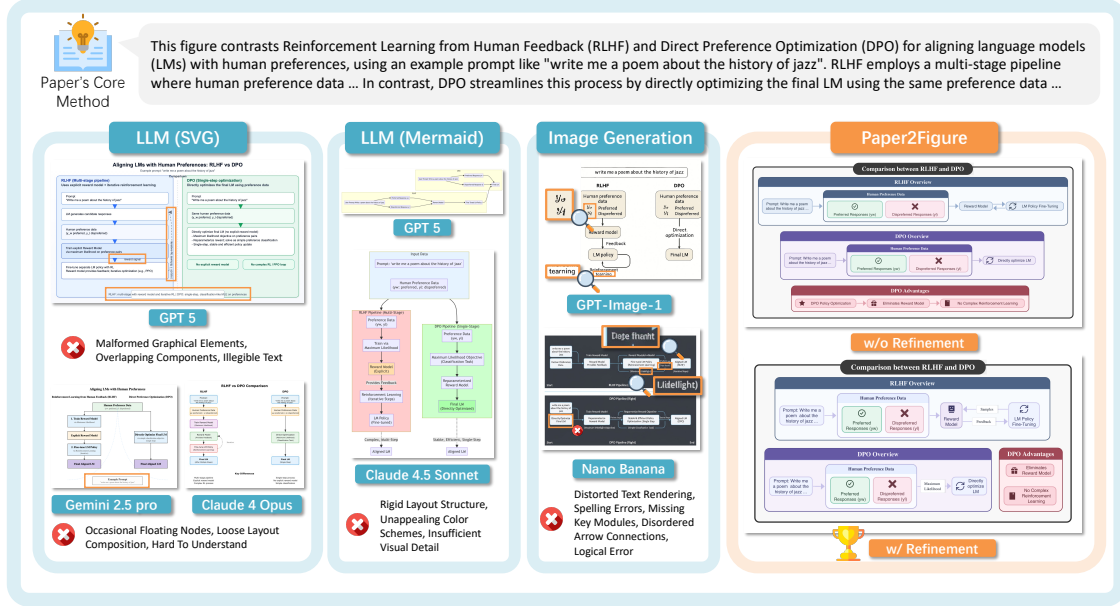


Figure 5. Case study of generated figures for the same input description. SVG-based models preserve detailed text but suffer from overlapping elements and loose layouts; Mermaid-based models produce rigid and less aesthetic results; image generation models often contain text distortions, missing modules, and incorrect logical flows. Paper2Figure achieves both structural accuracy and visual clarity, producing publication-quality figures.

els (LLMs) [1, 9, 27, 39] and multimodal generative systems. One major line of work employs LLMs to produce code-based visual representations such as Scalable Vector Graphics (SVG) or high-level markup languages like Mermaid [5, 6, 24, 25, 37]. These paradigms leverage the textual nature of structured graphics, enabling models to translate natural language into visual structures through code. SVG offers precise geometric control and editability, while Mermaid provides a higher-level abstraction for expressing logical relationships and workflows. Despite these advantages, both formats face similar challenges: generated outputs often contain semantic inconsistencies, misplaced or overlapping elements, and visually unrefined layouts [6]. The resulting figures may be syntactically correct yet fail to achieve clear or aesthetically coherent communication. Moreover, once generated, code-based figures typically require extensive manual revision to fix alignment or visual balance, limiting their practicality for end-to-end figure generation.

A second family of approaches utilizes text-to-image synthesis models [16, 26, 30–32, 36, 41], such as DALL-E 3, GPT-Image-1 [23], and Gemini 2.5 Flash Image (Nano Banana) [9], to directly render complex and visually appealing figures from natural language prompts. These models excel at stylistic diversity and visual richness but remain unreliable for scientific or technical illustrations. They frequently misrender textual labels, distort logical relations, or fail to preserve hierarchical structures described in the input [30]. Furthermore, their rasterized outputs are inher-

ently difficult to edit or refine, making them unsuitable for workflows where accuracy and modifiability are critical.

5.2. Multi-Agent Systems in Generative AI

The use of multiple cooperating agents has recently emerged as a powerful strategy for complex generative tasks [8, 28]. Frameworks such as MetaGPT [12] and related systems decompose problems into specialized roles [18, 33], like generation, critique, and revision, allowing agents to collaborate iteratively and self-correct errors. This paradigm has proven effective in domains such as software design, story writing, code synthesis [13], etc [11, 15, 35, 42], improving both reasoning depth and output consistency. However, applications of multi-agent collaboration to structured visual generation remain limited [19, 40]. In particular, the potential of such frameworks for producing scientific figures, where semantic precision, spatial coherence, and visual clarity are simultaneously required, has yet to be systematically explored.

6. Conclusion

This paper introduces Paper2Figure, a dual multi-agent system with an interactive web platform for generating high-quality scientific figures from text. Experiments on 100 academic figures show that Paper2Figure clearly surpasses both code-generating LLMs and image-generation models, producing publication-ready figures that more faithfully convey scientific semantics and structure.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 8
- [2] Anthropic. Claude sonnet 4.5 system card. <https://www.anthropic.com/claude-sonnet-4-5-system-card>, 2025. 5, 6
- [3] Anthropic. System card: Claude opus 4 & claude sonnet 4. Technical report, Anthropic, 2025. 5, 6
- [4] Eslam Mohamed Bakr, Pengzhan Sun, Xiaoqian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20041–20053, 2023. 1
- [5] Mu Cai, Zeyi Huang, Yuheng Li, Utkarsh Ojha, Haohan Wang, and Yong Jae Lee. Leveraging large language models for scalable vector graphics-driven image understanding. *arXiv preprint arXiv:2306.06094*, 2023. 1, 8
- [6] Mu Cai, Zeyi Huang, Yuheng Li, Utkarsh Ojha, Haohan Wang, and Yong Jae Lee. An investigation on llms’ visual understanding ability using svg for image-text bridging. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5377–5386. IEEE, 2025. 1, 8
- [7] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021. 1
- [8] Shuaihang Chen, Yuanxing Liu, Wei Han, Weinan Zhang, and Ting Liu. A survey on multi-generative agent system: Recent advances and new frontiers. *arXiv pre-print*, 2024. 8
- [9] Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blstein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 1, 5, 6, 8
- [10] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015. 1
- [11] Siwei Han, Peng Xia, Ruiyi Zhang, Tong Sun, Yun Li, Hongtu Zhu, and Huaxiu Yao. Mdocagent: A multi-modal multi-agent framework for document understanding. *arXiv preprint arXiv:2503.13964*, 2025. 1, 8
- [12] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2023. 8
- [13] Dong Huang, Jie M Zhang, Michael Luck, Qingwen Bu, Yuhao Qing, and Heming Cui. Agentcoder: Multi-agent-based code generation with iterative testing and optimisation. *arXiv preprint arXiv:2312.13010*, 2023. 8
- [14] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023. 1
- [15] Haonian Ji, Shi Qiu, Siyang Xin, Siwei Han, Zhaorun Chen, Hongyi Wang, Dake Zhang, and Huaxiu Yao. From edvisbench to eduvisagent: A benchmark and multi-agent framework for pedagogical visualization. *arXiv preprint arXiv:2505.16832*, 2025. 8
- [16] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023. 1, 8
- [17] Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Teufel, Marco Bellagente, et al. Holistic evaluation of text-to-image models. *Advances in Neural Information Processing Systems*, 36:69981–70011, 2023. 1
- [18] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for” mind” exploration of large language model society. *Advances in Neural Information Processing Systems*, 36: 51991–52008, 2023. 8
- [19] Ishani Mondal, Zongxia Li, Yufang Hou, Anandhavelu Natarajan, Aparna Garimella, and Jordan Boyd-Graber. Scidoc2diagrammer-maf: Towards generation of scientific diagrams from documents guided by multi-aspect feedback refinement. *arXiv pre-print*, 2024. 8
- [20] OpenAI. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 5, 6
- [21] OpenAI. Gpt-5 system card. Technical report, OpenAI, 2025. 5, 6
- [22] OpenAI. O3 and o4-mini system card. Technical report, OpenAI, 2025. 5, 6
- [23] OpenAI. Introducing 4o image generation, 2025. 1, 5, 6, 8
- [24] Juan A Rodriguez, Shubham Agarwal, Issam H Laradji, Pau Rodriguez, David Vazquez, Christopher Pal, and Marco Pedersoli. Starvector: Generating scalable vector graphics code from images. *arXiv preprint arXiv:2312.11556*, 2023. 1, 8
- [25] 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 Computer Vision and Pattern Recognition Conference*, pages 16175–16186, 2025. 1, 8
- [26] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 1, 8

- [27] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 8
- [28] Raphael Shu, Nilaksh Das, Michelle Yuan, Monica Sunkara, and Yi Zhang. Towards effective genai multi-agent collaboration: Design and evaluation for enterprise applications. *arXiv pre-print*, 2024. 8
- [29] Shubhankar Singh, Purvi Chaurasia, Yerram Varun, Pranshu Pandya, Vatsal Gupta, Vivek Gupta, and Dan Roth. Flowvqa: Mapping multimodal logic in visual question answering with flowcharts. *arXiv preprint arXiv:2406.19237*, 2024. 5
- [30] Jingxuan Wei, Cheng Tan, Qi Chen, Gaowei Wu, Siyuan Li, Zhangyang Gao, Linzhuang Sun, Bihui Yu, and Ruifeng Guo. From words to structured visuals: A benchmark and framework for text-to-diagram generation and editing. In *CVPR (2025)*, 2025. 1, 8
- [31] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12966–12977, 2025.
- [32] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 1, 8
- [33] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024. 8
- [34] Peng Xia, Siwei Han, Shi Qiu, Yiyang Zhou, Zhaoyang Wang, Wenhao Zheng, Zhaorun Chen, Chenhang Cui, Mingyu Ding, Linjie Li, et al. Mmie: Massive multimodal interleaved comprehension benchmark for large vision-language models. *arXiv preprint arXiv:2410.10139*, 2024. 1
- [35] Peng Xia, Jinglu Wang, Yibo Peng, Kaide Zeng, Xian Wu, Xiangru Tang, Hongtu Zhu, Yun Li, Shujie Liu, Yan Lu, et al. Mmedagent-rl: Optimizing multi-agent collaboration for multimodal medical reasoning. *arXiv preprint arXiv:2506.00555*, 2025. 8
- [36] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024. 1, 8
- [37] Ximing Xing, Juncheng Hu, Guotao Liang, Jing Zhang, Dong Xu, and Qian Yu. Empowering llms to understand and generate complex vector graphics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19487–19497, 2025. 1, 8
- [38] Zelai Xu, Zhexuan Xu, Xiangmin Yi, Huining Yuan, Xinlei Chen, Yi Wu, Chao Yu, and Yu Wang. Vs-bench: Evaluating vlms for strategic reasoning and decision-making in multi-agent environments. *arXiv preprint arXiv:2506.02387*, 2025. 1
- [39] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 8
- [40] Abhay Zala, Han Lin, Jaemin Cho, and Mohit Bansal. Diagrammergpt: Generating open-domain, open-platform diagrams via llm planning. In *COLM (2024)*, 2024. 8
- [41] Xinjie Zhang, Jintao Guo, Shanshan Zhao, Minghao Fu, Lunhao Duan, Jiakui Hu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, et al. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*, 2025. 1, 8
- [42] Yiyang Zhou, Yangfan He, Yaofeng Su, Siwei Han, Joel Jang, Gedas Bertasius, Mohit Bansal, and Huaxiu Yao. Reagent-v: A reward-driven multi-agent framework for video understanding. *arXiv preprint arXiv:2506.01300*, 2025. 8