ANIMATEBANANA: TRANSFORMING PROCESS DIAGRAMS INTO ANIMATED NARRATIVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Diagrams are often used to visually communicate scientific processes. However, conveying process dynamics and explanatory flow from a static diagram is challenging. Although animated narratives improve comprehension and engagement, creating them from a diagram is a manual and time-consuming process. We present **ANIMATEBANANA**, a framework for automatically transforming process-oriented scientific diagrams into animated narratives. Given a diagram and a user-specified presentation style, ANIMATEBANANA first constructs an editable, animation-aware programmatic representation of the diagram. Specialized multimodal agents progressively reason over the source diagram and structured intermediate representations to determine presentation structure, narration, and animation. Critic-mediated feedback is used to refine intermediate outputs. The resulting narratives are faithful to the source diagram, controllable, and deployable across multiple presentation formats (PDF, SVG, GIF, MP4, and PPTX). To support this new task, we introduce **AnimateBench**, a curated benchmark of 5,000 process-oriented diagrams with verified animated narratives spanning diverse diagram families, presentation styles, and levels of structural complexity, including expert-derived reference narratives where available. We further introduce task-specific metrics for evaluating final narratives. Automated evaluation and controlled user studies show that ANIMATEBANANA substantially outperforms competing baselines across animation validity and narrative quality. Ablations further support explicit intermediate representations, source diagram grounding, and early-stage critic feedback as important design choices for reliable animated narrative generation.

1 INTRODUCTION

Process-oriented scientific diagrams (Wong, 2011) are used to communicate neural architectures, scientific workflows, algorithms, and engineering systems by visually encoding process components and information flow. Unlike general-purpose plots or illustrations, the underlying concepts are often dynamic. As a result, readers must infer information flow, temporal progression, and causal relationships from a static image. Even with accompanying text, viewers must mentally reconstruct the underlying process and its dynamics.

To compensate, researchers, educators, and industry practitioners routinely animate diagrams during presentations. Prior work in multimedia learning has shown that suitably designed animated presentation strategies can improve comprehension and knowledge retention while reducing cognitive load (Mayer & Moreno, 2003; Rey et al., 2019; Xie et al., 2017; Schneider et al., 2023). However, creating animated narratives remains largely manual. Authors must decide what to present, when to present, how to guide the viewer's attention, and what to explain at each stage. Producing high-quality animated narratives requires substantial effort and specialized authoring tools.

Recent advances in multimodal foundation models have enabled impressive progress in scientific figure generation, editing, and executable graphics generation. Diagram-to-graphics-code systems recover editable programmatic representations from raster images (Rodriguez et al., 2025a; Belouadi et al., 2024b). Other systems generate or modify publication-quality scientific illustrations (Zhu et al., 2026a; Li et al., 2026b; Zhu et al., 2026b; Shao et al., 2026). However, generating an animated narrative requires more than an editable representation of the diagram. It requires recovering the

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.

diagram structure and identifying coherent presentation units. It also requires determining their order and coordinating the visual progression with accompanying text narration. Together, these requirements define a new task: *scientific diagram-conditioned animated narrative generation*.

We introduce ANIMATEBANANA, a modular framework of specialized multimodal agents that transforms process-oriented scientific diagrams into animated narratives. The generated narrative is conditioned on a user-specified presentation style. Rather than generating the narrative in a single step, ANIMATEBANANA progressively reasons over explicit intermediate representations (Fig. 1). It first generates an editable, animation-aware programmatic representation with element identities, structural grouping and metadata constructed specifically for downstream sequencing and animation. Subsequent agents plan the presentation sequence and narration before generating the final animation. Critic-guided feedback is used to verify and refine intermediate representations throughout the pipeline. The source diagram provides additional visual grounding where required. When available, ANIMATEBANANA exploits auxiliary context (e.g. source document, figure caption) to improve narration.

The resulting animated narratives are faithful and editable, with explicit control over animation style. The narratives can be rendered into multiple deployment formats (SVG, GIF, MP4, PPTX), allowing them to be reused across papers, presentations and online media without manual re-authoring for each medium. ANIMATEBANANA can also produce narratives that render natively inside standard PDF viewers, enabling animated visual experiences within conventional PDF documents.

To support this emerging task, we introduce **AnimateBench**, the first benchmark for process-oriented scientific diagram animation and narration. AnimateBench consists of 5,000 curated process-oriented diagrams with verified animated narratives. The bench also includes 500 human-authored presentations from related oral talk videos. We introduce task-specific metrics for evaluating the final animated narratives. The evaluations show that ANIMATEBANANA generates high-quality animated narratives and outperforms multiple baselines. Stage-specific ablations show the importance of ANIMATEBANANA's structured intermediate representations, source-diagram grounding and early-stage critic feedback. User studies further show that ANIMATEBANANA's animated narratives are highly preferred by human evaluators.

The process diagram illustrating ANIMATEBANANA's process flow is itself animated using our framework. This animation and other supporting material can be viewed at our anonymized website https://animatebanana.github.io/animatebanana_supplementary/

2 RELATED WORK

Diagram-to-graphics-code and programmatic representations: Recent work has explored programmatic representations (e.g., SVG, TikZ) from raster images (Rodriguez et al., 2025a;b; Li et al., 2026a; Belouadi et al., 2024a;b), including SVG generation and animation for icons, simple illustrations, and chemical structures (Wang et al., 2026b). More recent approaches incorporate explicit reasoning, iterative rendering, and visual feedback to improve quality of vector graphics (Wang et al., 2026a; Liang et al., 2026a) and to obtain high-fidelity document representations (Raghuveer et al., 2026). In contrast to these approaches, ANIMATEBANANA goes beyond visually faithful and editable reconstruction. The diagram representation is constructed specifically with addressable elements and structural grouping for downstream narrative planning and animation.

Scientific diagram generation and understanding: Multimodal foundation models have recently been applied to scientific visual communication, including systems for generating posters (Pang et al., 2025), presentation slides (Zheng et al., 2025; Ge et al., 2025) and videos (Shi et al., 2025). Several recent agentic systems generate and iteratively refine editable scientific figures (Zhu et al., 2026a; Li et al., 2026b; Zhu et al., 2026b; Huang et al., 2026; Shao et al., 2026; He et al., 2026a). Complementary efforts have explored explicit intermediate representations for diagram understanding (Xu & Forbus, 2025), representation learning (Wang et al., 2021), structural and layout recovery (Zhang et al., 2024; Shi et al., 2023), and benchmarks for scientific-diagram and vector-graphic understanding (Zhang et al., 2026; Ortega & Gómez-Pérez, 2026; Bo et al., 2026; Wang et al., 2026b). Recent work has also studied scientific image synthesis through programmatic generation, using explicit understanding and planning stages to improve structural and scientific correctness (Lin

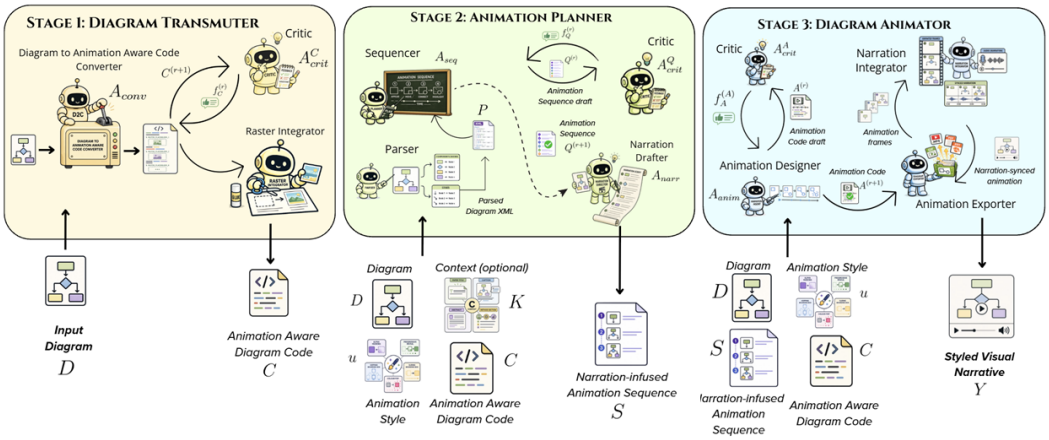


Figure 1: A High-Level Overview of the ANIMATEBANANA pipeline

et al., 2026). However, these resources do not address the presentation structure, temporal progression, and narration required for diagram-conditioned animated narrative generation.

Animated narratives and agentic generation: A complementary line of work studies motion generation from single images (Shirakawa et al., 2025), narrated data videos (Shen et al., 2024; 2025; Shao et al., 2025), and narration-centric authoring (Wang et al., 2025). Some recent systems generate editable animations over structured vector graphics (Tseng et al., 2025; Liu et al., 2025; Levy et al., 2026; Liang et al., 2026b). Other systems generate narrated presentations for papers (Shi et al., 2025; Chen et al., 2026), code explainer videos (Chen et al., 2026) and afford greater freedom in constructing the visual presentation. Recent agentic visual-generation systems such as GenAgent (Jiang et al., 2026) and SceneOrchestra (He et al., 2026b) focus on static image and 3D scene generation. Agent frameworks such as Re^4 (Cheng et al., 2026) and ToolTree (Yang et al., 2026) demonstrate the value of structured reasoning and feedback without task-specific model training. ANIMATEBANANA uses a plan–execute–critique workflow with representations and transformations specialized for diagram-conditioned animated narrative generation.

3 ANIMATEBANANA

Given a static process-oriented diagram D, a user-specified animation style u, and optional auxiliary context K, our goal is to generate an animated narrative $Y = \mathcal{F}(D, u; K)$. By default, $K = \varnothing$. When available, K may include the figure caption or relevant source document text. The animation style u specifies how the visual progression is realized. The style choices include *Progressive Reveal*, which incrementally reveals elements from a blank canvas; *Hopping Bounding Box* and *Sliding Bounding Box*, which discretely or continuously move a focus box across successive elements; *Color Pop*, which progressively restores color to elements of an initially grayed-out diagram; and *Alpha Masking*, which progressively unmasks elements of an initially masked diagram. Each style may optionally be augmented with pan–zoom effect over the region currently being presented.

Rather than generating the narrative directly, ANIMATEBANANA uses a multi-stage agentic framework that progressively constructs representations tailored to animation and narration. As illustrated in Figure 1, the framework comprises three stages. The *Diagram Transmuter* converts D into animation-aware diagram code C, where visual elements become addressable and organized for downstream animation. The *Animation Planner* reasons over C with visual grounding from D, optionally using K, to obtain a narration-infused sequence S that encodes presentation order and grouping at the desired granularity. The *Diagram Animator* utilizes S along with code C and style u to obtain the final animated narrative Y. Throughout the process, specialized multimodal agents construct and iteratively refine these representations through critic-guided feedback.

3.1 STAGE-I: DIAGRAM TRANSMUTER

As shown in Figure 1, the *Diagram Transmuter* converts the input diagram D into animation-aware diagram SVG code C. It is important to note that the Diagram Transmuter is not a generic diagram-to-code converter. It constructs C specifically for downstream narrative generation. The generated code also incorporates persistent markers and animation-related metadata that support subsequent presentation planning, narration, and animation.

The Converter agent $\mathcal{A}_{\text{conv}}$ generates an initial code representation $C^{(0)} = \mathcal{A}_{\text{conv}}(D, \phi, \phi)$, assigning persistent semantic identifiers to animatable elements such as nodes, connecting arrows, text labels, and composite elements. These identifiers provide explicit handles for animation attributes (e.g., visibility, timing) without altering the visual organization of D. Since C is used throughout the subsequent stages, errors in its content can propagate to animation planning. To address this, the Critic agent $\mathcal{A}^{C}_{\text{crit}}$ evaluates the rendered representation of $C^{(r)}$ against D using a visual fidelity rubric based on semantic content, containment, layout, style, and relative proportions. Here, r represents refinement round index. The Critic assigns a score in $[0, 1]$, with 0.7 as the pass threshold, and produces feedback $f_C^{(r)} = \mathcal{A}^{C}_{\text{crit}}(D, C^{(r)})$. The Converter then revises the representation as $C^{(r+1)} = \mathcal{A}_{\text{conv}}(D, C^{(r)}, f_C^{(r)})$. Refinement continues until rendering satisfies the visual fidelity threshold or a maximum number of rounds (3) is reached.

Scientific diagrams often contain photographs, icons and other visually complex regions that are poorly suited to direct programmatic reconstruction. During code generation, these regions are represented by placeholders with a short identifier label. The Raster Integrator uses the source diagram and the list of placeholder identifiers to propose bounding boxes for each placeholder. The Raster Integrator then extracts the corresponding regions from diagram D and inserts them into C, preserving their position and identity.

3.2 STAGE II: ANIMATION PLANNER

Although diagram code C from Stage I makes diagram elements individually addressable, it does not specify how they should be presented. It is the animation style u that determines how diagram elements are grouped and ordered at the required level of granularity. Some styles (***Progressive Reveal, Alpha Masking and Color Pop***) animate every element, edge and text label in the diagram. ***Hopping Box, Sliding Box*** affect only block-level diagram elements and not arrows and text labels.

The Parser (Figure 1) converts the animation-aware diagram code C into a structured XML representation P, using the source diagram D for visual grounding and the semantic identifiers and grouping introduced by the preceding module. C retains the information required for faithful rendering and animation. P abstracts away rendering-specific details but preserves diagram entities, their connections, and hierarchical relationships. Hierarchy is particularly relevant to presentation granularity, since a module may be introduced as a whole or through its constituent elements at different points in the narrative. Given the parsed XML representation P and Diagram D, the Sequencer agent $\mathcal{A}_{\text{seq}}$ constructs an initial presentation sequence $Q^{(0)} = (q_1, \ldots, q_T)$, where each q_t contains one or more elements to be introduced at step t. The Sequencer must choose among alternative groupings and traversal orders while accounting for structural dependencies and hierarchy. For example, connected elements may be introduced together. But a composite module may also be presented before its internal components.

At refinement round r, the Sequence Critic $\mathcal{A}^{Q}_{\text{crit}}$ evaluates $Q^{(r)}$ for coverage and presentation coherence to produce feedback $f_Q^{(r)} = \mathcal{A}^{Q}_{\text{crit}}(P, Q^{(r)}, D)$. The Sequencer revises its proposal as $Q^{(r+1)} = \mathcal{A}_{\text{seq}}(P, Q^{(r)}, f_Q^{(r)}, D)$ until the sequence is validated or the refinement limit (2 rounds) is reached. The style u is used to select the correct subset of elements (e.g. blocks, arrows, text labels) to be introduced for each member q_t of the validated sequence.

Finally, the Narration Drafter A_{narr} examines the animation sequence and the diagram D to determine the explanation at each stage so that the narration is synchronized with the evolving visual state. It generates narration text n_t for each presentation step q_t to obtain $S = ((q_1, n_1), \ldots, (q_T, n_T))$. When available, auxiliary context K, such as the figure caption or source document, provides addi-

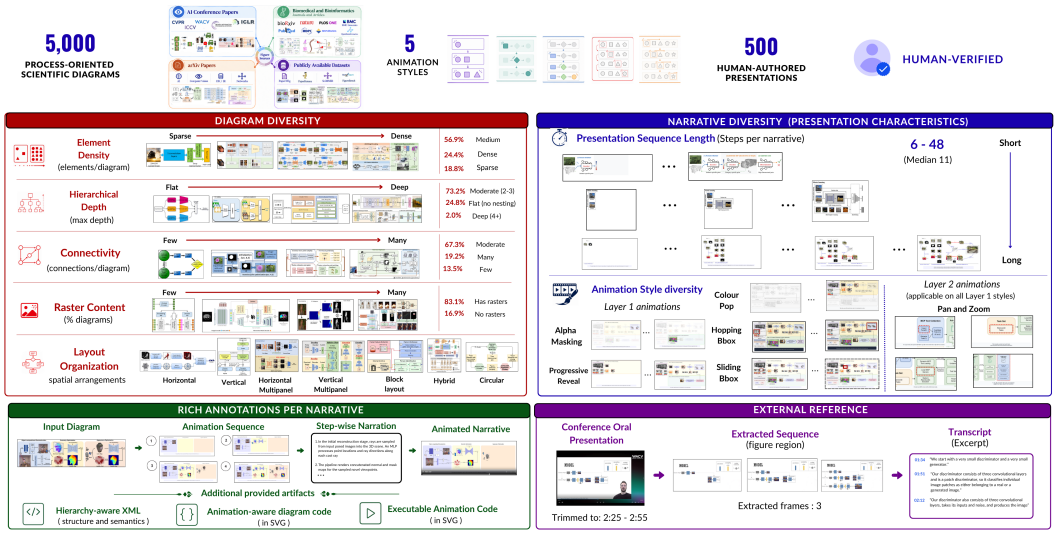


Figure 2: A high-level overview of AnimateBench

tional semantic information for this decision, particularly when an element's role is not evident from the diagram alone.

3.3 Stage III: Diagram Animator

The *Diagram Animator* converts the narration-infused sequence S into the final animated narrative Y. Given S, the source image D, the animation-aware code C, and a user-specified style u, the Animation Designer agent $\mathcal{A}_{\text{anim}}$ generates an initial executable animation SVG code $A^{(0)} = \mathcal{A}_{\text{anim}}(D, C, S, u, \phi, \phi, \phi)$. The generated code is rendered into frames, and each frame is evaluated by the Animation Critic $\mathcal{A}^{A}_{\text{crit}}$ against the animation sequence and source diagram. The Critic verifies that each focus element becomes active at its designated state and follows the prescribed traversal order. It also checks that no element is introduced prematurely. In parallel, the Critic performs structural checks on the animation code to ensure that the animation style u is properly implemented, element IDs are preserved without modification, and that all raster references remain unchanged. At refinement round r, the Critic reports violations of these visual and temporal constraints as feedback $f_A^{(r)} = \mathcal{A}^{A}_{\text{crit}}(D, S, A^{(r)}, M^{(r)})$. The critic memory $M^{(r)}$, shown in Figure 1, records feedback and attempted corrections from previous rounds. The Animation Designer uses both the current feedback and this history to determine a revised solution, producing $A^{(r+1)} = \mathcal{A}_{\text{anim}}(D, C, S, u, A^{(r)}, f_A^{(r)}, M^{(r)})$. Refinement continues until the animation is validated or the maximum number of rounds (3 rounds) is reached. Once validated, A is rendered into the visual states required by the target presentation style.

The Animation Exporter converts these states into the requested deployment format (PDF, SVG, GIF, MP4, or PPTX). In parallel, the Narration Integrator associates the step-level narration in S with the corresponding visual states, using speech synthesis if audio narration is requested. The final narrative Y retains the presentation and narration decisions encoded in S while adapting the decisions to the target style and output format.

4 AnimateBench: Benchmarking Scientific Diagram Animation

We introduce **AnimateBench**, a benchmark of 5,000 process-oriented scientific diagrams spanning five animation styles. For this, we first curate process-oriented diagrams from existing diagram datasets (Zhu et al., 2026a; Wu et al., 2024; Rodríguez et al., 2023; Zhu et al., 2026b) and independently source scientific papers spanning computer vision, machine learning, information retrieval, bioinformatics, computer networking and related areas. We focus on figures that communicate pipelines, architectures, workflows, and other structured processes. The selection is guided by diversity in element density, connectivity, hierarchy, layout and raster content (see Figure 2).

Constructing animation annotations at this scale manually is prohibitively expensive. We therefore use ANIMATEBANANA with Gemini-3.5 Flash to generate initial candidate narratives. This bootstrap model is weaker than the backbone used in our main experiments and is used only to initialize the annotation process. Human verifiers inspect each candidate narrative. They revise intermediate representations as needed to correct errors in coverage, sequencing or animation-style. The resulting verified narratives form the reference collection in AnimateBench. Prior to benchmark creation, a separate development set of 300 diagrams, excluded from AnimateBench, was used to refine ANIMATEBANANA modules' configuration and finalize the quality assessment protocol for human verifiers.

We also collect the corresponding conference presentation videos when available. We temporally localize the portion in which the figure is explained and spatially crop the video to the diagram region, retaining the accompanying transcript. The resulting presentation sequence serves as a nominal human-authored reference for the narrative in the benchmark counterpart. In addition to the animated narrative, AnimateBench includes metadata such as animation style and available source context (e.g. figure caption, method section) for each diagram. We also retain hierarchy-aware XML to assist evaluation.

5 EVALUATION METRICS

Evaluating an animated scientific diagram requires separating basic validity from presentation quality. For instance, an animation that substantially alters the source diagram or violates style should not receive a high score merely because its presentation sequence appears coherent. We therefore use a two-stage evaluation. Generated animations must first satisfy visual fidelity and animation-style compliance criteria. Subsequently, we assess the quality of their visual progression and narration. Human-authored presentations are used as reference material and in the user study but not as targets for automated evaluation since presentation structure and granularity reflect author-specific communicative intent rather than a unique ground truth.

Validity: We use two video-level validity criteria. **Visual Fidelity (VF)** evaluates whether the animated output preserves the source diagram's content, layout, geometry, proportions, typography, and visual appearance while remaining temporally stable. **Animation Style Compliance (ASC)** evaluates whether the requested animation style is applied consistently according to its defining behavior. Each criterion receives a four-level ordinal judgment, with the upper two levels constituting a pass and the lower two a failure. For an animation Y, we define $G(Y) = \mathbb{1}[\mathrm{VF}(Y)\ \text{passes}]\ \mathbb{1}[\mathrm{ASC}(Y)\ \text{passes}]$ such that $G(Y) = 1$ only when both validity requirements are satisfied.

Narrative quality: For valid animations, we evaluate each presentation step along three complementary dimensions. **Selection Sensibility (SS)** assesses whether elements selected at the current step form an appropriate and structurally coherent focus. **Granularity and Pacing (GP)** assesses whether the amount and complexity of information introduced at each step form an appropriate presentation unit. **Narration Alignment (NA)** assesses whether the accompanying narration is temporally aligned with the active visual content and provides a coherent, contextually grounded explanation. Each metric is judged at every presentation step using a four-level ordinal scale. Let $r_t^m \in \{1, 2, 3, 4\}$ denote the ordinal judgment for metric $m \in \{\mathrm{SS}, \mathrm{GP}, \mathrm{NA}\}$ at presentation step t, where 4 denotes the highest-quality judgment. For an animation containing N evaluated steps, we measure the normalized ordinal distance from the highest-quality judgment as $d_m = \frac{1}{3N} \sum_{t=1}^{N} |4 - r_t^m|$, and define the corresponding quality score as $Q_m = 1 - d_m$. The overall narrative-quality score is $Q = (Q_{\mathrm{SS}} + Q_{\mathrm{GP}} + Q_{\mathrm{NA}})/3$, and the final **Animated Narrative Score (AN)** is $\mathrm{AN}(Y) = G(Y)Q(Y)$, with $\mathrm{AN} \in [0, 1]$. AN is averaged over all outputs, with invalid outputs contributing zero. We report the VF and ASC pass rates, the three narrative-quality components, and AN. Details including judging rubrics and aggregation examples are provided in the supplementary material.

6 EXPERIMENTS AND DISCUSSION

Experimental setup: We evaluate ANIMATEBANANA on AnimateBench through automatic evaluation, ablations, and a user study. Baselines are evaluated either *end-to-end*, using their final gen-

Table 1: **Results on ANIMATEBENCH for animated-narrative generation. Animation Validity (blue):** VF=Visual Fidelity, ASC=Animation Style Compliance. **Narrative Quality (green):** SS=Selection Sensibility, GP=Granularity and Pacing, NA=Narration Alignment. **Overall (red):** AN=Animated Narrative Score. SS, GP, NA, AN are all validity-gated. All metrics lie in $[0, 1]$. Results for intermediate representations are provided in the supplementary material.

		Animation Validity		Narrative Quality			Overall
Method	**Params**	**VF↑**	**ASC↑**	**SS↑**	**GP↑**	**NA↑**	**AN↑**
Closed-source models (Diagram-to-Animation)							
Gemini 3.1 Pro	–	0.173	0.155	0.132	0.149	0.112	0.131
Opus 5 on Claude Code	–	0.155	0.107	0.084	0.081	0.073	0.079
GPT-5.6 Sol	–	0.191	0.073	0.068	0.069	0.058	0.065
Gemini-3.5 Flash	–	0.164	0.148	0.121	0.129	0.087	0.112
Gemini-3.7 Flash	–	0.171	0.152	0.127	0.134	0.092	0.118
ANIMATEBANANA (with *Gemini-3.7 Flash*)	–	**0.967**	**0.934**	**0.856**	**0.894**	**0.880**	**0.877**
ANIMATEBANANA *backbone ablations*							
Qwen3.8-27B	27B	0.679	0.423	0.350	0.393	0.358	0.367
Qwen3-VL-32B-Instruct	32B	0.027	0.014	0.011	0.012	0.012	0.012
Gemma 4	31B	0.238	0.142	0.102	0.130	0.105	0.110
GLM-4.6V	106B	0.014	0.010	0.009	0.010	0.007	0.007
Qwen3-VL-235B-A22B	235B	0.016	0.012	0.010	0.013	0.011	0.011
DeepSeek-V4 Pro	1.6T	0.066	0.039	0.033	0.037	0.025	0.030
Kimi K3	2.8T	0.326	0.140	0.126	0.137	0.133	0.132
ANIMATEBANANA *stage-specific ablations*							
Stage I w/o Critic	–	0.956	0.857	0.778	0.816	0.807	0.801
Stage II w/o Diagram Image Input	–	0.945	0.846	0.768	0.807	0.783	0.786
Sequencer (Stage II) w/o XML Input	–	0.923	0.824	0.744	0.780	0.760	0.761
Sequencer (Stage II) w/o Critic	–	0.967	0.911	0.825	0.867	0.853	0.848
Narration Drafter (Stage II) w/o Context	–	0.967	0.934	0.856	0.889	0.822	0.856
Animation Designer (Stage III) w/o Diagram Image Input	–	0.956	0.879	0.794	0.837	0.830	0.820
Animation Designer (Stage III) w/o Critic	–	0.967	0.934	0.848	0.890	0.880	0.873
ANIMATEBANANA (w/o Stage I, II and III critics)	–	0.956	0.843	0.769	0.809	0.795	0.791

erated animation, or *pipeline-integrated*, by replacing the corresponding ANIMATEBANANA agent backbone. All methods are therefore comparable on final animation metrics. Intermediate metrics are reported only when the corresponding artifacts are available. We use Gemini-3.7-Flash as the backbone for ANIMATEBANANA in all applicable pipeline stages. Unless otherwise stated, methods receive the same source diagram and available auxiliary context. For judging, we use Kimi-2.6 with fixed evaluation rubrics. **Evaluation protocol:** We report the metrics from Sec. 5 on full ANIMATEBENCH. The main evaluation compares final animated narratives across methods. We additionally evaluate backbone and stage-specific ablations (Table 1). Finally, our user study complements the automatic metrics with human judgments of animation and narrative quality, system preference, the utility of auxiliary context, and the effect of human verification in AnimateBench. The complete study protocol and related details are provided in the supplementary material.

6.1 RESULTS

Overall performance and direct generation: ANIMATEBANANA achieves high animation validity (0.967 VF, 0.934 ASC) and narrative quality (0.856 SS, 0.894 GP, 0.88 NA), yielding an overall AN of 0.877 (Table 1). Direct diagram-to-animation generation performs substantially worse, with the strongest closed-source baseline reaching only 0.131 AN. Notably, ANIMATEBANANA with Qwen3.8-27B reaches 0.367 AN, exceeding all direct baselines despite using a smaller backbone. This indicates that model scale alone is insufficient. Progressively tackling the task through ANIMATEBANANA's structured intermediate stages is critical for style-aligned narrative generation.

Representations and source grounding: Removing the XML input to the Sequencer reduces AN from 0.877 to 0.761, showing the importance of explicit diagram structure for presentation planning. Removing the source diagram from Stage II and the Animation Designer lowers AN to 0.786 and 0.82, respectively, indicating that structured representations and source-diagram grounding provide complementary information. Removing auxiliary context lowers AN from 0.877 to 0.856 and also reduces NA, supporting its role in narrative generation.

Table 2: **Human evaluation of animation quality.** Mean ± standard deviation over 25 evaluated diagrams per system, with each diagram first averaged over its participant ratings. GP is not human-rated and is fixed to 1 for valid animations when computing $\mathrm{AN}_{\mathrm{human}}$.

System	VF	ASC	SS	NA	$\mathrm{AN}_{\mathrm{human}}$
AnimateBanana	**.929±.22**	**.909±.24**	**.837±.25**	**.820±.24**	**.855±.24**
AnimateBanana (Qwen3.8-27B)	.431±.46	.421±.46	.340±.40	.368±.43	.376±.42
Gemini 3.1 Pro	.305±.42	.275±.41	.238±.36	.234±.35	.249±.37

Critic feedback: Removing all critics lowers AN from 0.877 to 0.791. Removing the Stage I critic alone causes a substantial drop to 0.801, whereas removing the Animation Designer (Stage III) critic has little effect (0.873). This suggests that critic feedback is more important in early stages where errors can affect subsequent downstream representations.

Robustness to judge choice: To assess whether evaluation is sensitive to the choice of judge, we re-score 50 stratified outputs with Claude-Sonnet-5 and GPT-5.6-Sol using the same rubrics. Their AN rankings agree strongly with Kimi-2.6, with Kendall's $\tau = 0.82$ and 0.88, respectively. Agreement across all three judges is also high, with Krippendorff's $\alpha = 0.95$ (95% CI $[0.88, 1.00]$).

Effect of animation style on AN: We generated all five animation styles for 100 diagrams. Mean AN ranged from 0.916 to 0.962 across styles, and the mean within-diagram standard deviation was 0.033. This suggests that AN is not overly affected by animation style.

6.2 User Study

Study design: We complement automatic evaluation with a user study involving 38 participants and 93 randomly selected source diagrams spanning different animation styles and diagram types. Each diagram–system received approximately five human ratings on average. The quality evaluation uses 25 diagrams per system. Similar to automated assessment, participants first judge visual fidelity (VF) and animation-style compliance (ASC) of individual animated narratives. Conditional on passing both criteria, they rate selection sensibility (SS) and narration–animation alignment (NA) on a 0–10 scale. Animations that fail either criterion are scored 0. All scores are normalized to $[0, 1]$. We report mean (± std) scores over diagrams for AnimateBanana, best open-source backbone (Qwen3.8-27B), and best-performing closed-source model (Gemini 3.1 Pro). Since animations were played at a self-selected pace, we do not evaluate Granularity and Pacing (GP) but set it to 1 for valid animations when computing $\mathrm{AN}_{\mathrm{human}}$. In the *Ranking-Based Evaluation*, participants compare competing systems and select the animation they would be most likely to use themselves. We report the overall win rate and pairwise preferences between AnimateBanana and each alternative. We also conduct two targeted evaluations with a selective cohort to assess the utility of auxiliary context and effect of human verification in AnimateBench. All participants evaluate the animated narratives with the source diagram available throughout the trial.

Human and automatic ratings: As shown in Table 2, AnimateBanana achieves 0.929 VF, 0.909 ASC, 0.837 SS, and 0.820 NA, resulting in an $\mathrm{AN}_{\mathrm{human}}$ of 0.855. Across the three systems, human evaluation preserves the system ordering observed under automatic evaluation (c.f. Table 1), although absolute scores differ between the two evaluation protocols. Inter-rater agreement is Krippendorff's $\alpha = 0.62$ for $\mathrm{AN}_{\mathrm{human}}$. Human and automatic ratings show a Spearman correlation of $\rho = 0.66$ for AN on $n = 75$ matched diagram–system outputs. Human and automatic judgments agree on joint animation validity (G) in 78.4% of outputs ($\kappa = 0.57$).

User preference: Figure 3 summarizes ranking-based evaluation. AnimateBanana achieves a win rate of 61.6%, compared with 11.8% for Qwen3.8-27B and 26.5% for human-authored presentations. AnimateBanana is preferred over Qwen3.8-27B in 79.1% of pairwise comparisons. The preference over Qwen3.8-27B agrees with the large difference observed in both automatic and human quality scores. Among trials in which AnimateBanana advanced to second round, it was preferred to human-authored presentation in 77.8% of comparisons. When two participants evaluated the same diagram, they selected the same tournament winner in 71% of pairs.

Context and verification: Narratives generated with auxiliary context are preferred in 67.6% of context-utility trials, compared with 15.5% for the without-context condition and 16.9% with no

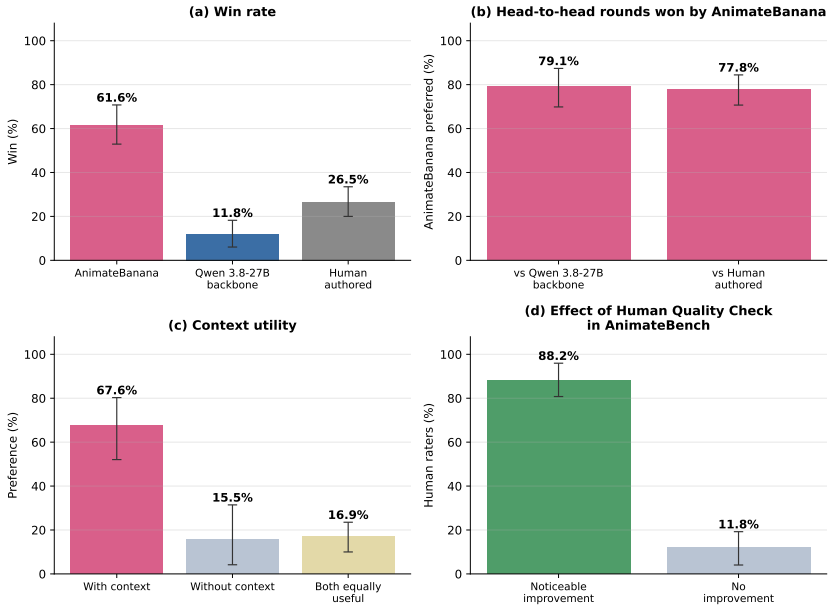


Figure 3: **Human evaluation.** (a) Tournament preference across ANIMATEBANANA, Qwen3.8-27B, and human-authored animations. (b) Pairwise tournament results. Comparison with the human-authored animation occurs when ANIMATEBANANA wins round 1. (c) Preference with and without context. (d) Effect of human verification on narrative quality. Error bars show 95% participant-clustered bootstrap confidence intervals.

preference. This agrees with the automatic ablation, where removing context reduces AN from 0.877 to 0.856. The two evaluations therefore support the use of caption and paper context beyond information available in the source diagram alone. For AnimateBench, participants judged human verification to produce a noticeable improvement in 88.2% of trials.

6.3 LIMITATIONS

Occasional failures arise on structurally dense diagrams. Closely packed elements and complex edge patterns can cause minor replication or sequencing errors, particularly when containment and directional cues compete. A few style-compliance failures occur with alpha masking, where inactive elements may become overly faint. Also, our current study focuses on process-oriented scientific diagrams. Extension to less structured illustrations remains an interesting direction. Finally, our user study evaluates animation quality rather than downstream learning outcomes. However, such assessment is not the focus of the paper and would require a dedicated task-based study.

7 CONCLUSION

We introduce *animated narrative generation for scientific process diagrams* and ANIMATEBANANA, a framework that explicitly represents diagram structure before planning and animation. Experiments show that ANIMATEBANANA's staged approach substantially outperforms direct diagram-to-animation generation. Ablations highlight the importance of source-diagram grounding and early-stage refinement. Human evaluation shows a strong preference for ANIMATEBANANA. We also introduce **AnimateBench** and task-specific metrics to support evaluation of this new task. Faithful, controllable animation can extend scientific diagrams beyond their static form. ANIMATEBANANA brings this capability closer to routine use in papers, presentations, tutorials, and other forms of scientific communication.

ETHICS STATEMENT

ANIMATEBANANA generates animated and narrated interpretations of scientific diagrams, and its outputs may therefore introduce incorrect emphasis, ordering, or narration even when the source diagram is preserved visually. Such outputs should not be treated as authoritative scientific explanations without human verification. We mitigate this risk by explicitly grounding generation in the source diagram, evaluating visual fidelity and narrative quality, and retaining editable intermediate representations that allow authors to inspect and revise the generated narrative before use. The system may also lower the cost of producing persuasive but misleading scientific visualizations. We therefore present it as an assistive authoring tool rather than a substitute for expert review. Our user study involves human participants and is conducted in accordance with applicable IRB procedures. Participation is voluntary and reported results are aggregated without identifying participants.

AI USE STATEMENT

Generative AI tools were used to revise manuscript text for clarity and readability, generate some smaller diagrams within the main figures, assist with reference formatting and code editing, and create the anonymized companion website containing supplementary resources. All AI-assisted text and artifacts were reviewed by the authors, and code was tested as part of the implementation and experimental pipeline. We did not use generative AI tools to develop the research methodology or experiments, formulate hypotheses or mathematical claims, implement the proposed method, or interpret experimental results. We take responsibility for the final content of this work, including text and artifacts produced with the aid of generative AI.

REFERENCES

Jonas Belouadi, Anne Lauscher, and Steffen Eger. AutomaTikZ: Text-guided synthesis of scientific vector graphics with TikZ. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=v3K5TVP8kZ. pg 2.

Jonas Belouadi, Simone Paolo Ponzetto, and Steffen Eger. Detikzify: Synthesizing graphics programs for scientific figures and sketches with tikz. In *NeurIPS*, volume 37, pp. 85074–85108, 2024b. pgs 1 and 2.

Weihao Bo, Shan Zhang, Yanpeng Sun, Jie Liu, Yongke Yao, Jinhao Du, Wei He, Kai Zou, Zechao Li, and Jingdong Wang. Diagram-MMU: A multi-modal benchmark for scientific diagrams. In *European Conference on Computer Vision (ECCV)*, 2026. pg 2.

Yanzhe Chen, Kevin Qinghong Lin, and Mike Zheng Shou. Code2video: A code-centric paradigm for educational video creation. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. pg 3.

Ao Cheng, Lei Zhang, and Guowei He. Re^4: Scientific computing agent with rewriting, resolution, review and revision. In *ICLR*, 2026. URL https://openreview.net/forum?id=PMdr3fQpBE. pg 3.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, et al. Autopresent: Designing structured visuals from scratch. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2902–2911. IEEE, 2025. pg 2.

Qijia He, Xunmei Liu, Hammaad Memon, Ziang Li, Zixian Ma, Jaemin Cho, Jason Ren, Daniel S Weld, and Ranjay Krishna. Vfig: Vectorizing complex figures in svg with vision-language models, 2026a. URL https://arxiv.org/abs/2603.24575. pg 2.

Yun He, Kelin Yu, and Matthias Zwicker. SceneOrchestra: Efficient agentic 3d scene synthesis via full tool-call trajectory generation. In *European Conference on Computer Vision*, 2026b. pg 3.

Siyuan Huang, Yifan Zhou, Yutong Gao, Zi Yin, Juyang Bai, Xinxin Liu, Rama Chellappa, Chun Pong Lau, Cheng Peng, Sayan Nag, and Shraman Pramanick. Scifig: Towards automating editable figure generation for scientific papers, 2026. URL https://arxiv.org/abs/2601.04390. pg 2.

Kaixun Jiang, Yuzheng Wang, Junjie Zhou, Pandeng Li, Zhihang Liu, Chen-Wei Xie, Zhaoyu Chen, Yun Zheng, and Wenqiang Zhang. GenAgent: Scaling text-to-image generation via agentic multimodal reasoning. In *European Conference on Computer Vision*, 2026. pg 3.

Matan Levy, Ran Margolin, Bar Cavia, Dvir Samuel, Yael Pritch, Shmuel Peleg, Alex Rav Acha, Ariel Shamir, and Dani Lischinski. Livesvg: Zero-shot svg animation via video generation. *arXiv preprint arXiv:2605.30174*, 2026. pg 3.

Sifan Li, Yujun Cai, Hongkai Chen, and Yiwei Wang. Geosvg-rl: Geometry-aware reinforcement learning for layout-constrained text-to-svg diagram generation. *arXiv preprint arXiv:2605.25447*, 2026a. pg 2.

Zhuoling Li, Jiarui Zhang, Ping Hu, Jason Kuen, Jiuxiang Gu, Hossein Rahmani, and Jun Liu. Automatic method illustration generation for ai scientific papers via drawing middleware creation, evolution, and orchestration. In *European Conference on Computer Vision (ECCV)*, 2026b. pgs 1 and 2.

Guotao Liang, Zhangcheng Wang, Juncheng Hu, Haitao Zhou, Ziteng Xue, Jing Zhang, Dong Xu, and Qian Yu. Render-in-the-loop: Vector graphics generation via visual self-feedback. In *European Conference on Computer Vision (ECCV)*, 2026a. pg 2.

Guotao Liang, Zhangcheng Wang, Chuang Wang, Juncheng Hu, Haitao Zhou, Junhua Liu, Jing Zhang, Dong Xu, and Qian Yu. VAnim: Rendering-aware sparse state modeling for structure-preserving vector animation. In *International Conference on Machine Learning*, 2026b. pg 3.

Honglin Lin, Chonghan Qin, Zheng Liu, Qizhi Pei, Yu Li, Zhanping Zhong, Xin Gao, Yanfeng Wang, Conghui He, and Lijun Wu. Scientific image synthesis: Benchmarking, methodologies, and downstream utility. In *European Conference on Computer Vision (ECCV)*, 2026. pg 2.

Vivian Liu, Rubaiat Habib Kazi, Li-Yi Wei, Matthew Fisher, Timothy Langlois, Seth Walker, and Lydia Chilton. Logomotion: Visually-grounded code synthesis for creating and editing animation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 2025. doi: 10.1145/3706598.3714155. pg 3.

Richard E. Mayer and Roxana Moreno. Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist*, 38(1):43–52, 2003. pg 1.

Raul Ortega and José Manuel Gómez-Pérez. From terminology to diagrams: Visual-instruction generation for scientific diagram understanding. In *(COLM)*, 2026. pg 2.

Wei Pang, Kevin Qinghong Lin, Xiangru Jian, Xi He, and Philip H. S. Torr. Paper2poster: Towards multimodal poster automation from scientific papers. In *Advances in Neural Information Processing Systems*, volume 38, 2025. pg 2.

R. Raghuveer, Anirudh Srinivasan, Venkata Kesav Venna, S. Sreevatsa, Aryan Jain, Sahithi Kukkala, and Ravi Kiran Sarvadevabhatla. REPLICA: An agentic framework for visually faithful document reconstruction. In *ICDAR*, 2026. pg 2.

Günter Daniel Rey, Maik Beege, Steve Nebel, and Maria Wirzberger. A meta-analysis of the segmenting effect. *Edu. Psy. Review*, 31:389–419, 2019. pg 1.

Joan A. Rodríguez, David Vázquez, Issam H. Laradji, Marco Pedersoli, and Pau Rodríguez. Ocr-vqgan: Taming text-within-image generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3689–3698, 2023. pg 5.

Juan A. Rodriguez, Abhay Puri, Shubham Agarwal, Issam H. Laradji, Sai Rajeswar, David Vazquez, Christopher Pal, and Marco Pedersoli. Starvector: Generating scalable vector graphics code from images and text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 29691–29693, 2025a. doi: 10.1609/aaai.v39i28.35369. pgs 1 and 2.

Juan A. Rodriguez, Haotian Zhang, Abhay Puri, Aarash Feizi, Rishav Pramanik, Pascal Wichmann, Arnab Mondal, Mohammad Reza Samsami, Rabiul Awal, Perouz Taslakian, Spandana Gella, Sai Rajeswar, David Vazquez, Christopher Pal, and Marco Pedersoli. Rendering-aware reinforcement learning for vector graphics generation. In *NeurIPS*, 2025b. pg 2.

Sascha Schneider, Felix Krieglstein, Maik Beege, and Günter Daniel Rey. Successful learning with whiteboard animations – a question of their procedural character or narrative embedding? *Heliyon*, 9(2):e13229, 2023. doi: 10.1016/j.heliyon.2023.e13229. pg 1.

Chenyang Shao, Jiahe Liu, Fengli Xu, and Yong Li. Livefigure: Generating editable scientific illustration with vlm agents. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, Seoul, South Korea, 2026. PMLR. URL https://proceedings.mlr.press/v306/. pgs 1 and 2.

Zekai Shao, Leixian Shen, Haotian Li, Yi Shan, Huamin Qu, Yun Wang, and Siming Chen. Narrative player: Reviving data narratives with visuals. *IEEE Transactions on Visualization and Computer Graphics*, 31(10):6781–6795, 2025. doi: 10.1109/TVCG.2025.3530512. pg 3.

Leixian Shen, Yizhi Zhang, Haidong Zhang, and Yun Wang. Data player: Automatic generation of data videos with narration-animation interplay. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):109–119, 2024. doi: 10.1109/TVCG.2023.3327197. pg 3.

Leixian Shen, Haotian Li, Yun Wang, Tianqi Luo, Yuyu Luo, and Huamin Qu. Data playwright: Authoring data videos with annotated narration. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):5884–5897, 2025. doi: 10.1109/TVCG.2024.3477926. pg 3.

Danqing Shi, Weiwei Cui, Danqing Huang, Haidong Zhang, and Nan Cao. Reverse-engineering information presentations: Recovering hierarchical grouping from layouts of visual elements. *Visual Intelligence*, 1(1):9, 2023. doi: 10.1007/s44267-023-00010-1. pg 2.

Jingwei Shi, Zeyu Zhang, Biao Wu, Yanjie Liang, Meng Fang, Ling Chen, and Yang Zhao. PresentAgent: Multimodal agent for presentation video generation. In Ivan Habernal, Peter Schulam, and Jörg Tiedemann (eds.), *EMNLP*, pp. 760–773, November 2025. ISBN 979-8-89176-334-0. URL https://aclanthology.org/2025.emnlp-demos.58/. pgs 2 and 3.

Takahiro Shirakawa, Tomoyuki Suzuki, Takuto Narumoto, and Daichi Haraguchi. Mg-gen: Single image to motion graphics generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 3554–3563, October 2025. pg 3.

Tiffany Tseng, Ruijia Cheng, and Jeffrey Nichols. Keyframer: Empowering animation design using large language models. In *IEEE Symposium on Visual Languages and Human-Centric Computing*, 2025. pg 3.

Haomin Wang, Qi Wei, Qianli Ma, Shengyuan Ding, Jinhui Yin, Kai Chen, and Hongjie Zhang. Reliable reasoning in SVG-LLMs via multi-task multi-reward reinforcement learning. In *European Conference on Computer Vision (ECCV)*, 2026a. pg 2.

Haomin Wang, Jinhui Yin, Qi Wei, Wenguang Zeng, Lixin Gu, Shenglong Ye, Zhangwei Gao, Yaohui Wang, Yanting Zhang, Yuanqi Li, Yanwen Guo, Wenhai Wang, Kai Chen, Yu Qiao, and Hongjie Zhang. InternSVG: Towards unified SVG tasks with multimodal large language models. In *The Fourteenth International Conference on Learning Representations*, 2026b. pg 2.

Shaowei Wang, Lingling Zhang, Xuan Luo, Yi Yang, Xin Hu, and Jun Liu. RL-CSDia: Representation learning of computer science diagrams. *arXiv:2103.05900*, 2021. pg 2.

Yun Wang, Leixian Shen, Zhengxin You, Xinhuan Shu, Bongshin Lee, John Thompson, Haidong Zhang, and Dongmei Zhang. Wonderflow: Narration-centric design of animated data videos. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):4638–4654, 2025. pg 3.

Bang Wong. The overview figure. *Nature methods*, 8(5):365–366, 2011. pg 1.

Siwei Wu, Yizhi Li, Kang Zhu, Ge Zhang, Yiming Liang, Kaijing Ma, Chenghao Xiao, Haoran Zhang, Bohao Yang, Wenhu Chen, Wenhao Huang, Noura Al Moubayed, Jie Fu, and Chenghua Lin. SciMMIR: Benchmarking scientific multi-modal information retrieval. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *ACL (Findings)*, August 2024. pg 5.

Heping Xie, Fuxing Wang, Yanbin Hao, Jiaxue Chen, Jing An, Yuxin Wang, and Huashan Liu. The more total cognitive load is reduced by cues, the better retention and transfer of multimedia learning: A meta-analysis and two meta-regression analyses. *PLOS ONE*, 12:e0183884, 08 2017. doi: 10.1371/journal.pone.0183884. pg 1.

Wangcheng Xu and Kenneth Forbus. Understanding diagrams with explicit intermediate visual representation. In *12th Annual Conference on Advances in Cognitive Systems (ACS)*, 2025. pg 2.

Shuo Yang, Caren Han, Yihao Ding, Shuhe Wang, and Eduard H Hovy. Tooltree: Efficient llm tool planning via dual-feedback monte carlo tree search and bidirectional pruning. In *ICLR*, pp. 30750–30772, 2026. pg 3.

Tingwen Zhang, Ling Yue, Zhen Xu, and Shaowu Pan. DiagramBank: A large-scale dataset of diagram design exemplars with paper metadata for retrieval-augmented generation. *arXiv preprint arXiv:2604.20857*, 2026. pg 2.

Xinyu Zhang, Lingling Zhang, Xin Hu, Jun Liu, Shaowei Wang, and Qianying Wang. Alignment relation is what you need for diagram parsing. *IEEE Transactions on Image Processing*, 33: 2131–2144, 2024. doi: 10.1109/TIP.2024.3374511. pg 2.

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. PPTAgent: Generating and evaluating presentations beyond text-to-slides. In *EMNLP*, pp. 14402–14418. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.728. pg 2.

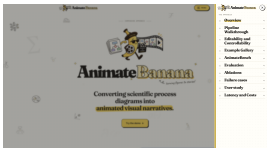
Dawei Zhu, Rui Meng, Yale Song, Xiyu Wei, Sujian Li, Tomas Pfister, and Jinsung Yoon. Paperbanana: Automating academic illustration for AI scientists. In *(ICML)*, 2026a. URL https://openreview.net/forum?id=FneePKFVHT. pgs 1, 2, and 5.

Minjun Zhu, Zhen Lin, Yixuan Weng, Panzhong Lu, Qiujie Xie, Yifan Wei, Sifan Liu, QiYao Sun, and Yue Zhang. AutoFigure: Generating and refining publication-ready scientific illustrations. In *International Conference on Learning Representations*, 2026b. pgs 1, 2, and 5.

8 APPENDIX

An anonymized website has been created for all the supplementary material, including animations and other media. This website can be accessed at https://animatebanana.github.io/animatebanana_supplementary/

Clicking on 'Menu' in top right corner of the webpage shows the navigation pane as shown in the screenshot below.



Note: The above link is in line with the following QA present as part of FAQ in the official Author Guidelines at https://iclr.cc/Conferences/2027/AuthorGuidelines

> **Q: Can I submit a link for demonstration (for example, including a video)?**
> Yes, such a link can be submitted as long as it is completely anonymous...

To ensure internet connectivity does not pose an issue, a condensed version of website contents have also been collected into a folder and included as a zip file - see Supplementary Material attached separately as part of the submission.