

ManimAgent: Self-Evolving Multimodal Agents for Visual Education

Wenjia Jiang^{1,6,*}, Zongyuan Cai^{2,*}, Yuanhang Shao², Chenru Wang³, Boyan Han²,
Zhixue Song⁴, Keyu Chen⁵, Shengwei An³, Xu Yang², Zhou Yang^{1,6,†}

¹University of Alberta ²Southeast University ³Virginia Tech

⁴Emotion Machine Inc. ⁵Vivavia Inc. ⁶Alberta Machine Intelligence Institute (Amii)

[†]Canada CIFAR AI Chair <https://manimagent.github.io/>

Abstract

Multi-round reflection lets agents built on large language models recover from failures within a single task, but lessons learned across many reflection rounds on one task are discarded before the next begins. We study this gap on a code-generation task: from a scientific paper section, the agent writes executable MANIM code. We present MANIMAGENT, a self-evolving multimodal agent that carries reflection experience across tasks through a dual-channel Episodic Memory Bank grown entirely from its own task stream, with no weight updates and no human seeds. After each animation converges, a vision–language model scores the rendered keyframes; the resulting signals populate a positive channel $\mathcal{M}^+$ that stores success rationales as soft Reference Examples, and a negative channel $\mathcal{M}^-$ that stores VLM-screened failure patterns as hard Known Pitfalls. On a fixed-probe evaluation against no-memory, matched-budget retrieval-augmented generation, and shuffled-memory baselines, blind human Pass@1 rises and reflection rounds fall as memory size grows. We will release the code, frozen memory snapshots, and the task stream.

1 Introduction

“Memory is not an instrument for exploring the past but for exploring the future.”
— (Endel Tulving)

Large language models and their agentic wrappers have become capable enough to handle complex, multi-stage code generation tasks (Chen et al., 2021; Yao et al., 2023; Shinn et al., 2023). A recent line of work applies this capability to visual education through MANIM (The Manim Community Developers, 2026), a Python animation library that programmatically renders mathematical and pedagogical scenes from declarative scripts. MANIM was originally developed for the 3Blue1Brown YouTube channel and is now community-maintained. Concretely, a multi-agent system reads a section of a scientific paper and

*Equal contribution.

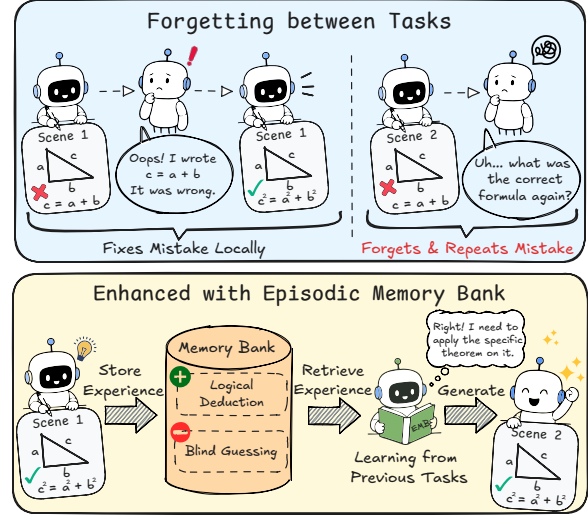


Figure 1: **MANIMAGENT closes the cross-task forgetting gap that reflection alone leaves open.** (Top) A reflection-only agent fixes its mistake within Task 1 but, without cross-task memory, repeats it on Task 2. (Bottom) MANIMAGENT stores each attempt in a dual-channel Episodic Memory Bank (EMB): positive entries (logical deductions) and negative entries (blind guesses), retrieved on Task 2 so it succeeds first try.

writes MANIM code that renders a short teaching animation, with promising results on isolated tasks (Chen et al., 2026; Samarth P et al., 2025; Joshi et al., 2026).

However, these systems are forgetful. A state-of-the-art reflection pipeline asked to animate the Fourier transform may spend many reflection rounds, and thousands of tokens, learning that a formula must be left-aligned, that a 0.3 s transition is too fast, or that a specific $\text{\LaTeX}$ macro silently breaks under MANIM; the animation is delivered, and every one of those lessons is thrown away. When the same system is later asked to explain a probability distribution, it makes the same layout mistakes and pays the full reflection cost again. Reflection-based agents (Madaan et al., 2023; Shinn et al., 2023) improve repeated trials through feedback, but do not evaluate whether visual-programming experience transfers across

disjoint tasks. We call this structural failure cross-task forgetting.

Intuitively, a competent human educator would not start from zero on every new topic. An instructor who has explained Fourier transforms many times carries two kinds of accumulated experience: positive templates of how the time-frequency picture should be staged, and negative lessons about the formula and the diagram overlapping until the audience loses the thread. The first kind tells the educator what to do; the second kind tells them what to avoid. Cross-task forgetting, in this light, is the absence of either kind of memory across episodes. Motivated by this analogy, we extend the standard reflection pipeline with an explicit cross-task memory of the same two flavours. Concretely, we present MANIMAGENT, a self-evolving multi-agent system that narrows the inter-task gap by adding two components to a standard reflection pipeline: a vision-language model (VLM) acting as a structured reward source over rendered keyframes, and a dual-channel Episodic Memory Bank (EMB) grown entirely from the task stream of the system itself, with no human seeds and no weight updates.

The first component is a strong VLM, drawing on LLM/VLM-as-a-judge evaluation work (Liu et al., 2023; Zheng et al., 2023; Lee et al., 2024), used as a structured reward source. Given the rendered keyframes, the source teaching text, and the storyboard, the VLM returns a multi-axis critique covering logical flow, layout and occlusion, and pedagogical accuracy. This critique plays two roles inside MANIMAGENT: it drives the per-task reflection loop in which the system iteratively revises the generated code, and it gates the memory-write step that decides what is added to the EMB at the end of each task.

The second component is the Episodic Memory Bank itself, a dual-channel external store that the system populates as it processes a stream of paper-section tasks. After each task converges, two complementary signals are distilled. The positive channel $\mathcal{M}^+$ records a natural-language success rationale attached to every high-scoring scene, acting as a soft global template for what good looks like on related tasks. The negative channel $\mathcal{M}^-$ records a structured VLM-screened failure pattern for every improving FAILURE→SUCCESS transition in the reflection trace, acting as a hard exclusion list that rules out fatal mistakes the positive channel alone cannot surface (the operational VLM-score

write gate is detailed in §3.3). On subsequent tasks, both channels are retrieved by task embedding and injected into the Coder prompt as Reference Examples and Known Pitfalls.

To validate this design, we grow the EMB on a memory-building stream and then freeze snapshots of different sizes. Each snapshot is evaluated on the same disjoint probe tasks in read-only mode, so probe outputs cannot be written back to memory and task order cannot explain differences between conditions (§4.3). Because the same VLM both scores rendered keyframes and gates memory writes, using VLM scores as the headline metric would risk evaluator leakage (the reward model is also the judge, so the system could appear to improve simply by drifting toward the preferences of the VLM). We therefore use blind human judgement of first-attempt videos together with reflection rounds as the headline evidence, and report VLM scores only as auxiliary diagnostics (§4.5).

This work makes three contributions. We propose MANIMAGENT, a self-evolving multi-agent system that narrows the inter-task gap with a VLM reward source and a dual-channel EMB, with no weight updates (§3). Two LLM distiller agents convert each reflection trace into a free-form success rationale ($\mathcal{M}^+$) and a structured failure lesson ($\mathcal{M}^-$), gated by VLM-scored improvement (§3.3). We will also release a JSON-indexed paper-section animation dataset with scene-role labels and an output-level human-scoring protocol, and evaluate on it via a fixed-probe snapshot design that isolates cross-task memory from task-order effects (§4.3, §4.1).

2 Related Work

Retrieval-augmented generation and episodic memory in LLM agents. Retrieval-augmented generation retrieves external evidence into the prompt of a model instead of storing all task knowledge in parameters (Lewis et al., 2020). For code generation, systems such as DocPrompting and RepoCoder retrieve documentation, API references, or repository context to improve generation quality (Zhou et al., 2023; Zhang et al., 2023). Episodic-memory agents such as MemGPT, Generative Agents, and ExpeL write experiences back to persistent stores during or after execution (Packer et al., 2023; Park et al., 2023; Zhao et al., 2024). We share the retrieval interface with these lines and adopt a matched-budget Manim-code RAG corpus

(§4.2) as a strong baseline. However, our memory is self-generated from VLM-scored animation traces rather than from static documents or unconstrained conversation logs; it is written only under explicit operational write gates; and it is split into positive and negative channels so that reference examples and known pitfalls can play different roles in the Coder prompt. Appendix E makes this contrast explicit.

In-context example configuration. Multimodal in-context learning studies show that demonstration selection, quality, and composition can substantially affect model behaviour. Yang et al. (Yang et al., 2023) study image selection and caption configuration for image captioning, while Li et al. (Li et al., 2024) extend demonstration retrieval and configuration to visual question answering. MANIMAGENT is adjacent to this literature because retrieved experiences likewise provide prompt-time guidance, but it is not formulated as a multimodal ICL method: its examples are distilled from the agent’s own historical trajectories, admitted through VLM/reward-gated memory writes, and separated into positive Reference Examples and negative Known Pitfalls.

Lifelong learning and skill libraries. Voyager is the closest prior work in spirit: an LLM-driven agent in MINECRAFT accumulates an external skill library that subsequent tasks can call (Wang et al., 2023). Related web-agent work also uses large multimodal models to act across long-horizon interfaces (He et al., 2024). MANIMAGENT differs in three ways. First, the reward signal is not a sparse rule-based success bit, but a continuous, multi-axis visual judgement from a VLM. Second, the memory is dual-channel: positive rationales and VLM-screened failure patterns are distilled with explicit VLM-score improvement gating (§3.3), whereas skill-library agents mainly preserve successful solutions. Third, the EMB is grown entirely from the task stream of the system itself, with no seed primitives, human-curated examples, or external video corpus. Recent Manim-generation work also explores agentic inference strategies for animation generation (Silva et al., 2026); our focus is complementary, on cross-task memory and fixed-probe evaluation. Appendix D (Table 3) summarises these distinctions, and Appendix C extends the related-work discussion to programmatic animation pipelines and reflection-based agents.

3 Method

MANIMAGENT produces a short MANIM teaching animation from a paper section (text, scene role, domain tag) via a standard reflection pipeline of six agents that storyboard, code, render, diagnose crashes, score keyframes with a VLM, and revise. To this pipeline we add a self-grown, VLM-gated, dual-channel Episodic Memory Bank (EMB) and a pair of LLM distillers: the positive channel $\mathcal{M}^+$ stores success rationales injected as soft Reference Examples, and the negative channel $\mathcal{M}^-$ stores VLM-screened failure patterns injected as hard Known Pitfalls. Both channels are distilled at consolidation time, so the EMB holds transferable lessons rather than literal before/after diffs. Across the task stream, the system reads from the EMB before generating and writes to it after converging, with no weight updates and no human seeds (Figure 2).

3.1 Setup and Notation

A task is a paper section $\tau = (s, r, d)$, where s is the section text, d is its domain tag, and the scene role r takes one of BACKGROUND, METHOD, EXPERIMENT, or CONCLUSION. The STORYBOARD Σ decomposes τ into ordered scene specifications $\{\sigma_i\}_{i=1}^n$; each σ_i carries a scene name, the claim to be conveyed, the evidence to use, and the final takeaway, which downstream agents are required to preserve. Within a scene, t indexes reflection iterations inside whichever loop is active (text-reflection or visual-reflection, defined in §3.2); $c_i^{(t)}$ is the candidate MANIM code at iteration t , $K_i^{(t)}$ are $n_K=4$ keyframes sampled from a successful render, and $u(t) \in [0, 100]$ is the aggregate VLM score on those keyframes (defined only after a render succeeds, since text-reflection iterations may crash before rendering). The execution trace of a scene records the tuple $(c_i^{(t)}, \text{render result}, u(t) \text{ if available, diagnostic})$ at every iteration; consolidation reads from this trace. A frozen sentence encoder $\phi(\cdot)$ embeds retrieval queries.

The external memory is a single store $\mathcal{M} = \mathcal{M}^+ \cup \mathcal{M}^-$ backed by one SQLite table and one retrieval interface, but indexed by two per-polarity Faiss IndexFlatIP structures so that a query against $\mathcal{M}^+$ never returns a record drawn from $\mathcal{M}^-$ even at high recall. Records share a context header $(s_\sigma, r_\sigma, d_\sigma, \text{paper_id}, \text{section_id})$, used to form the retrieval query, and a provenance

box limits, short-circuit and tie-break rules, and distillation length caps are listed in Appendix B.

3.3 Dual-Channel Memory Design

Two choices define the EMB: maintaining two channels rather than one, and gating each channel with a different write rule. We motivate both; the full field-level schema is deferred to Appendix J (Table 11).

Why two channels, not one. $\mathcal{M}^+$ records what a good solution looks like; $\mathcal{M}^-$ records what to avoid. A positive-only skill-library architecture (as in Voyager-style agents (Wang et al., 2023)) has no mechanism for hard exclusion: a known-bad pattern would have to be encoded indirectly, by hoping that positive exemplars outvote it at retrieval time. Keeping the channels separate lets us inject $\mathcal{M}^+$ as a soft prior and $\mathcal{M}^-$ as a hard rule, mirroring the shape of cue a human educator carries: positive templates plus explicit pitfalls.

Positive-channel write rule. A scene whose best-of- N rendition c^* scores $u^* \geq \theta_{\text{high}} = 85$ is written to $\mathcal{M}^+$ as a success rationale plus the chosen code, score, and frame hash. We deliberately set $\theta_{\text{high}} < \theta_{\text{conv}}$, so that every scene which triggers the auto-pass also clears the write bar, and a scene that narrowly converged in the last visual-loop iteration is still remembered. The threshold makes $\mathcal{M}^+$ a high-precision soft prior: every retrieved exemplar has cleared an explicit quality bar.

Two kinds of failure transition. The execution trace defined in §3.1 contains two structurally different improvement signals, which we distil separately. Text transitions are crash→success pairs at adjacent text-reflection iterations: $c_i^{(t-1)}$ raises an error of category κ , $c_i^{(t)}$ renders cleanly. Since there is no continuous score across a crash, we accept every such transition and use the error category plus traceback tail as the diagnostic. Visual transitions are $v_{\text{rev}} \rightarrow v_{\text{rev}} + 1$ pairs at adjacent visual-reflection iterations: both render cleanly, both receive aggregate VLM scores, and we require $u(t) - u(t-1) \geq \Delta_{\text{fail}} = 5$ on the 0–100 scale, using the revision_instruction as the diagnostic. The text-margin requirement is degenerate (a crash has no numeric score); the visual margin filters out noise from a stochastic reviewer.

Lessons are LLM-distilled, not literal diffs. For each surviving failure transition, a separate LES-

SON DISTILLER agent consumes the before/after code and the diagnostic and emits the six $\mathcal{M}^-$ body fields; for success records, a RATIONALE WRITER agent consumes the chosen code and emits a high-level rationale ρ . We use LLM distillation rather than literal diffs because raw before/after code is dominated by scene-specific tokens (variable names, numerical constants, camera angles) that would mislead retrieval; the distillation step lifts the lesson ℓ to a transferable description. Field-level length caps are listed in Appendix B.

Provenance, deduplication, and embedder pinning. The dedup key on the SQLite store combines run id, scene id, polarity, extraction source, and transition ordinal. The ordinal lets $v_0 \rightarrow v_1$ and $v_1 \rightarrow v_2$ remain distinct records rather than the later overwriting the earlier, so a long reflection trace contributes multiple lessons when it should. The embedder identity (model, dim, version) is pinned in an on-disk spec file at first creation and refused on reopen if a different encoder is requested, so EMB snapshots cannot be silently re-embedded between writes and reads, a precondition for the fixed-probe protocol of §4.3, which evaluates the same snapshot multiple times. At evaluation time, an `-emb-readonly` flag nulls out the post-task consolidation step so that the reflection traces of the probe set cannot leak into the snapshot under test.

No capacity, no eviction. We do not currently apply capacity bounds, eviction policies, or paraphrase-level deduplication: $\mathcal{M}^+$ and $\mathcal{M}^-$ grow monotonically across the memory-building stream. `hit_count` and `last_used` columns are recorded in provenance for future consolidation work but are not consumed by the current retrieval interface. All numerical hyperparameters introduced above are summarised in Appendix P (Table 14); the conceptual contrast between EMB and standard retrieval-augmented generation is deferred to Appendix E.

4 Experiments

To validate the effectiveness of MANIMAGENT, we conduct experiments on our paper-section animation dataset with frozen EMB snapshots, matched-budget RAG and Random-EMB baselines, and a blind human-scoring protocol. The central question is whether a self-grown Episodic Memory Bank (EMB) makes a VLM-reflection animation agent improve across tasks without any model-weight up-

Condition / Variant	Intervention or note	Human Pass@1 $\uparrow$	Reflection rounds $\downarrow$	Human quality $\uparrow$
Headline conditions — fixed-probe evaluation under blind human scoring				
VLM reflection only	Standard reflection without retrieval	68.5	12.2	3.41
Manim-code RAG	Static curated Manim corpus, matched retrieval	83.3	11.4	3.65
Random EMB	Shuffled EMB content, matched count and budget	69.6	11.3	3.42
EMB@0	Empty self-grown memory	62.0	12.2	3.26
EMB@50	50-entry self-grown EMB snapshot	66.7	11.3	3.31
EMB@100	100-entry self-grown EMB snapshot	75.0	10.8	3.43
EMB@200 (= Full EMB)	200-entry EMB snapshot, both channels gated	84.9	6.5	3.88
(a) Memory channels — which polarity carries the signal?				
No $\mathcal{M}^+$ (success)	Retrieve only failure memories	70.0	12.0	3.40
No $\mathcal{M}^-$ (failure)	Retrieve only success memories	80.9	11.6	3.52
No memory	Disable both memory channels	61.8	12.2	3.20
(b) Memory content and retrieval — what gets stored and looked up?				
Rationale-only $\mathcal{M}^+$	Drop code excerpt from positive injection	75.0	10.5	3.50
Ungated $\mathcal{M}^-$	Store every diagnostic, no improvement gate	81.8	8.4	3.64
Top-1 retrieval	One neighbour per channel ($k_+ = k_- = 1$)	81.0	9.9	3.84
(c) Reviewer and static-corpus controls — gains beyond reward and retrieval per se				
Weak VLM	Step-3.7-Flash replaces the GPT-5.5 reviewer	80.0	8.6	3.42
RAG (doc only)	Manim docs only, matched budget	82.1	11.5	3.57

Table 1: **Headline conditions and ablations on the fixed-probe set share a single human-scoring protocol.** The top block reports the seven headline conditions used in §4.3; EMB@200 (= Full EMB) is best on all three metrics and the only condition that improves over Manim-code RAG on all three headline metrics. The lower three blocks ablate Full EMB: (a) isolates the contribution of each polarity, (b) stresses the consolidation pipeline (rationale vs. code, gate stringency, retrieval depth), and (c) separates EMB gains from reviewer strength and from a static documentation-only retrieval corpus. RAG (doc + code) is identical to the headline “Manim-code RAG” row and not repeated.

dates. Because the same VLM diagnoses rendered keyframes and gates memory writes, we do not use VLM scores as the primary evidence. Instead, the headline metrics are blind human judgements on first-attempt outputs and the reflection rounds of the system; ablations on channel isolation, write gates, retrieval depth, and VLM–human agreement address mechanism and reward reliability.

4.1 Evaluation Task Stream

Our evaluation instrument is a released, indexed paper-section dataset. Each task asks the system to read a local unit of a scientific paper and generate a short MANIM animation for that unit. The release separates tasks at the paper level into three roles: **memory-building** (39 tasks) grows the EMB before snapshots are frozen; **fixed-probe** (33 tasks) evaluates each frozen snapshot in read-only mode and supplies the headline metrics; **cross-test** (40 tasks) is a held-out exploratory split used only for the qualitative inspection of retrieved memories in Appendix N. Model-visible inputs are restricted to the paper title, abstract, target-unit excerpt, required prior context, prior objects, scene

role, domain, and source type; evaluation-only fields (key claims, reference scene plans, rubrics, condition identifiers, fatal flags) and output-level human scores are kept outside the model-input subtree and never written to the EMB. A quarantined debugging holdout, the field-isolation discipline, and the release validator are detailed in Appendix F and Appendix G.

Starting from an empty memory bank, we process the memory-building stream and freeze snapshots at

$$\mathcal{K} = \{0, 50, 100, 200\}.$$

During probe evaluation, the snapshot is used only for retrieval; no probe output is written back to memory. Thus, differences between EMB@0 and EMB@ K on the same probe tasks measure accumulated cross-task memory rather than test-time learning. The task scope is local paper-section animation; full-paper video generation is left outside the headline experiment to avoid narrative and long-video confounds.

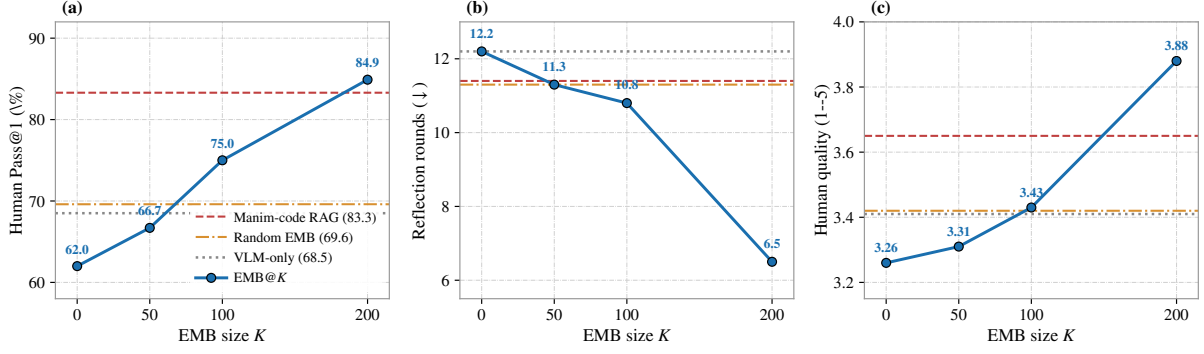


Figure 3: **The descriptive point estimates improve monotonically with EMB size.** Solid line with markers tracks the frozen EMB snapshot at $K \in \{0, 50, 100, 200\}$ on each of the three headline metrics; horizontal references are Manim-code RAG (dashed red), Random EMB (dash-dot orange), and VLM-only (dotted grey). For Human Pass@1 and Human quality, the EMB curve is above the Manim-code RAG reference at $K=200$; for reflection rounds (lower is better), the corresponding point estimate is substantially lower.

4.2 Experimental Setup

Configurations. We compare seven conditions on the same fixed-probe set: VLM reflection only, an empty EMB (EMB@0), frozen EMB snapshots at $K \in \{50, 100, 200\}$, a matched-budget Manim-code RAG baseline, and Random EMB. Per-condition interventions are summarised in Table 1; full operational descriptions and the Coder prompt template that turns retrieved records into the Reference Examples and Known Pitfalls slots are deferred to Appendix B and Appendix K.

Models. All text agents and the multimodal VLM reviewer use a single GPT-5.5 family endpoint; only the reviewer receives rendered keyframes. Decoding settings are fixed across conditions. The Weak-VLM ablation replaces the reviewer with Step-3.7-Flash. Renderer version, retrieval hyperparameters, and per-channel truncation budgets are held constant and listed in Table 14.

Baselines. Our main comparison is **Manim-code RAG**, which extends VLM reflection with retrieval from a fixed, curated corpus of public Manim documentation and community examples through the same encoder, index, and truncation budget as the EMB; the corpus is in-domain, the retrieval machinery is identical, and it has no exposure to our tasks, so an EMB–RAG gap is evidence for the self-grown dual-channel content of the EMB rather than for retrieval per se (Appendix E; a documentation-only variant appears in Table 1). Comparisons with generic LLM-agent frameworks and the closest paper-to-animation systems (Code2Video (Chen et al., 2026), Manimator (Samarth P et al., 2025), LLM2Manim (Joshi

et al., 2026), Paper2Video (Zhu et al., 2025)) are discussed in Appendix C; we do not benchmark them because their output modality and termination criteria do not map to a single MANIM scene.

Human evaluation. 54 blind raters supply a binary usability decision and five 1–5 Likert scores (Likert, 1932) per first-attempt video across 25 audited fixed-probe tasks. Agreement is low: Fleiss’ $\kappa = -0.01$ for binary usability, with 59.4% full agreement (Fleiss, 1971), and the five Likert-dimension ICCs are 0.24, 0.00, 0.10, 0.33, and -0.09 (Shrout and Fleiss, 1979). We retain all blind ratings without exclusion, majority voting, or post hoc adjudication. Recruitment, protocol, and agreement details are in Appendix H.

Metrics. We report **Human Pass@1**, the fraction of first-attempt videos judged by blind raters as usable paper-section animations; **reflection rounds**, the sum of text-reviewer retries and VLM-driven visual revisions; and **Human quality**, the mean of five 1–5 human rating dimensions: visual design, key-claim coverage, visual robustness, animation flow, and first-attempt usability. VLM scores are auxiliary diagnostics only.

4.3 Main Results: First-Attempt Generation

For each frozen snapshot, we run the identical fixed-probe set with the same model, renderer, reflection budget, decoding parameters, and human-scoring protocol. The fixed-probe design controls for task order, test-time learning, and prompt-length effects; the Manim-code RAG and Random EMB baselines make the retrieval comparison stricter. The headline block of Table 1 reports the seven evaluated

conditions and Figure 3 visualises the EMB snapshot trends against the three reference baselines.

The descriptive point estimates for EMB@ K move monotonically on all three metrics (Figure 3, Table 1): Pass@1 rises 62.0→84.9, quality 3.26→3.88, and reflection rounds drop 12.2→6.5 from EMB@0 to EMB@200. Relative to Manim-code RAG, EMB@200 has slightly higher point estimates on both human metrics and a substantially lower reflection-round estimate (6.5 vs. 11.4); Random EMB rules out generic memory-format-context explanations.

Figure 4 gives a compact qualitative example on one BERT fixed-probe run: visual reflection turns first renderable scenes with weak layout hierarchy into clearer final frames. Appendix M shows the interpolated online curve, and Appendix O gives the complete qualitative trace table and larger before/after montage.

4.4 Mechanism Analysis

The main results establish that the EMB helps; this section asks which design choices drive the gain. We run six ablation and control variants on the same fixed probe (one per row of Table 1), each removing or replacing a component of the dual-channel memory or the consolidation pipeline: channel isolation (dropping $\mathcal{M}^+$ or $\mathcal{M}^-$ individually), write-gate removal (keeping every failed-attempt diagnostic with no minimum-improvement gate), shallow retrieval ($k_+=k_-=1$), rationale-only $\mathcal{M}^+$ (dropping the code excerpt from positive injection), matched-token RAG controls (documentation-only, with documentation-plus-code represented by the headline Manim-code RAG row), and a weak-VLM substitution (Step-3.7-Flash as reviewer). Each variant uses the same human-scoring protocol; the per-variant mechanism rationale is in Appendix L.

The channels play differentiated roles: removing $\mathcal{M}^+$ collapses Pass@1 and quality (positive exemplars carry first-attempt benefit), while removing $\mathcal{M}^-$ preserves quality but inflates reflection rounds (failure memories cut repair cost); the gated full EMB beats ungated $\mathcal{M}^-$ and Top-1 retrieval, limiting context-padding explanations.

4.5 Reward Reliability, Cost, and Scalability

The VLM that drives revision and gates writes is a noisy internal signal; VLM-human agreement on the paired probe subset ($n=37$ videos) is weak (Pearson $r= -0.17$, Spearman $\rho= -0.20$, Co-

hen’s $\kappa= -0.07$ (Cohen, 1960); Appendix I). Per-task cost is an estimated 36–38K LLM and 20–23K VLM tokens over 19–31 min, and Figure 5 shows EMB@200 on the cost-quality Pareto frontier under a comparable retrieval budget, with higher wall-clock time than Manim-code RAG; per-condition cost and operational limits are deferred to Appendix P and the Limitations section.

Scaling beyond EMB@200. To examine how the system behaves as the memory bank grows beyond the range used in the main human evaluation, we further evaluate EMB snapshots containing 250, 300, 350, and 400 entries. All four settings use the same 33 fixed-probe tasks, model configuration, retrieval strategy, and generation pipeline, so the EMB size remains the primary variable. Table 2 reports token consumption, reflection rounds, and the internal VLM score for these additional settings.

EMB size	Token consumption	Reflection rounds	VLM score
EMB@250	49K	6.8485	86.6500
EMB@300	48K	6.7273	86.2929
EMB@350	48K	6.5758	86.4389
EMB@400	50K	6.8788	86.6202

Table 2: **Scaling diagnostics beyond EMB@200 on the fixed-probe set.** The additional settings use the same tasks, model configuration, retrieval strategy, and generation pipeline. VLM score is an internal diagnostic; these settings were not independently re-evaluated by human raters.

The results indicate that the scaling trend largely plateaus beyond 250 entries. Across EMB@250–400, the VLM score remains within a narrow range of 86.29–86.65 and does not improve monotonically as the memory bank expands. EMB@400 obtains a score of 86.62, close to 86.65 at EMB@250, suggesting that the small variations across these settings reflect saturation rather than continued gains. Token consumption remains between 48K and 50K, while the average number of reflection rounds stays between 6.58 and 6.88.

This stability is consistent with the fixed-depth retrieval design. For each scene, the system injects only the retrieved top- $k_+ = 2$ positive and top- $k_- = 3$ negative memories into the Coder prompt, rather than placing the entire EMB in the model context. Increasing the total EMB size therefore expands the candidate pool for vector retrieval without causing the prompt length or per-task token consumption to grow linearly. The current imple-

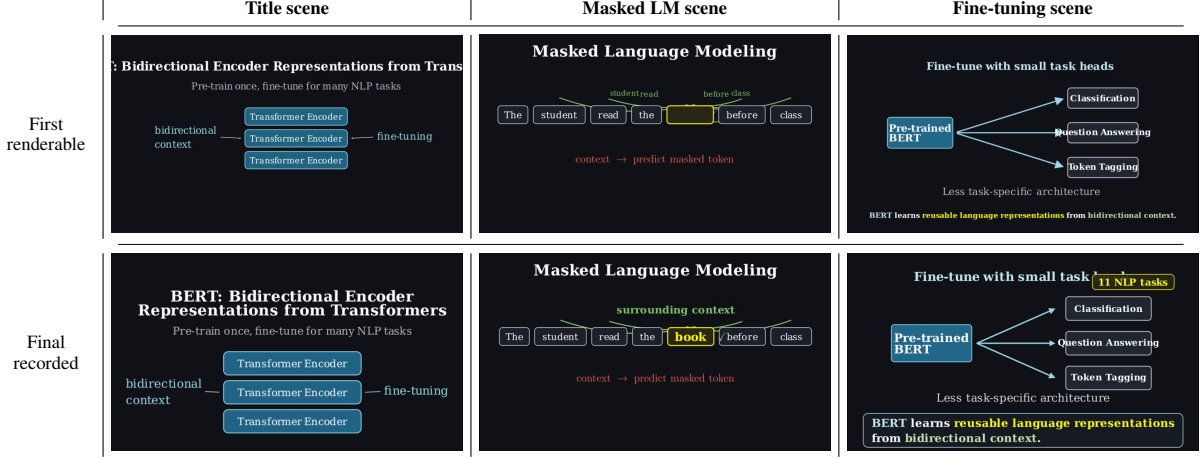


Figure 4: **A compact qualitative example traces three scene revisions for one BERT fixed-probe task.** The main paper keeps the qualitative experiment compact by showing three representative scene revisions as before/after thumbnails. The title scene becomes balanced around the encoder stack, the masked-language-modeling scene exposes surrounding context and token prediction, and the fine-tuning scene clarifies the downstream-task hierarchy. Appendix O reports the detailed trace outcomes, VLM axes, and larger before/after grid.

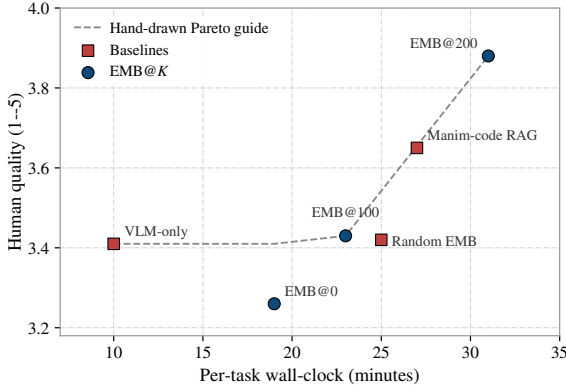


Figure 5: **EMB@200 sits on the cost-quality Pareto frontier.** Each marker is a condition; x is per-task wall-clock from Table 13, y is Human quality from Table 1. EMB@200 trades 4 additional minutes over Manim-code RAG for a +0.23 gain on Human quality while EMB@0/RAG/Random cluster around quality ≈ 3.4 . Point estimates from Table 13 and Table 1.

mentation uses a fixed-dimensional vector index, whose storage and retrieval overhead remains small relative to code generation, Manim rendering, and VLM review within the evaluated range.

We do not repeat human evaluation for these additional settings. Accordingly, the VLM scores in Table 2 are used only to characterize the internal scaling trend and should not be interpreted as independent evidence of human-perceived quality improvements. Overall, larger EMBs remain operationally stable but provide limited additional benefit at the scale considered here.

5 Conclusion

The dual-channel EMB of MANIMAGENT converts per-task reflection traces into cross-task experience without weight updates. A fixed-probe snapshot protocol shows monotonic gains across EMB snapshots, with the mature EMB@200 snapshot matching Manim-code RAG on human metrics while substantially reducing reflection rounds. Ablations link these gains to the two channels rather than to retrieval or prompt-length artefacts. Artifacts will be released upon acceptance.

Limitations

Evaluation scope. MANIMAGENT is evaluated only with MANIM across five English-language scientific domains; transfer to other renderers or languages, and effects on student learning, remain untested.

Scale and evaluator dependence. We validate the EMB only through 400 entries and do not extrapolate its stable trend to production-scale or indefinitely growing memories. Memory maintenance, robustness to model upgrades, and prompt sensitivity remain open. Because VLM-human agreement is weak, VLM scores are used only for repair and memory writing, not as headline quality evidence (Appendices P and I).

Baseline and contamination. Controls omit long-context retrieval, and split isolation does not remove within-stream near-duplicates (Appendices E and F).

Ethics and Artifact Release Statement

Human evaluation involved 54 anonymous, uncompensated volunteers and collected no direct identifiers or sensitive content. The study was not conducted under a formal institutional framework. Raw ratings were temporarily stored with author-only access and deleted within 14 days. The release excludes paper PDFs and full text and preserves applicable licenses, notices, and source attribution for code, documentation, videos, and EMB snapshots. Further recruitment, data-handling, and artifact-release details appear in Appendices H and G.

Acknowledgments

The project is supported in part by research grants from Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; and the National Science and Engineering Research Council (NSERC).

References

- Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziem, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. 2026. [GEPA: Reflective Prompt Evolution Can Outperform Reinforcement Learning](#). In *International Conference on Learning Representations*, volume 2026, pages 8479–8565.
- Yuzheng Cai, Siqi Cai, Yuchen Shi, Zihan Xu, Lichao Chen, Yulei Qin, Xiaoyu Tan, Gang Li, Zongyi Li, Haojia Lin, Yong Mao, Ke Li, and Xing Sun. 2025. [Training-Free Group Relative Policy Optimization](#). Preprint, arXiv:2510.08191.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. [Evaluating Large Language Models Trained on Code](#). arXiv preprint arXiv:2107.03374.
- Yanzhe Chen, Kevin Qinghong Lin, and Mike Zheng Shou. 2026. [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.
- Jacob Cohen. 1960. [A Coefficient of Agreement for Nominal Scales](#). *Educational and Psychological Measurement*, 20(1):37–46.
- Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2026. [The Faiss Library](#). *IEEE Transactions on Big Data*, 12(2):346–361.
- Joseph L Fleiss. 1971. [Measuring Nominal Scale Agreement among Many Raters](#). *Psychological Bulletin*, 76(5):378–382.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. [CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing](#). In *International Conference on Learning Representations*, volume 2024, pages 57734–57811.
- Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. [WebVoyager: Building an end-to-end web agent with large multimodal models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890, Bangkok, Thailand. Association for Computational Linguistics.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. [Billion-Scale Similarity Search with GPUs](#). *IEEE Transactions on Big Data*, 7(3):535–547.
- Aastha Joshi, Hongyi Ke, Meet Gajjar, Aaron Christian, Qi Wang, and Jun Chen. 2026. [LLM2Manim: Pedagogy-Aware AI Generation of STEM Animations](#). arXiv preprint arXiv:2604.05266.
- Seongyun Lee, Seungone Kim, Sue Park, Geewook Kim, and Minjoon Seo. 2024. [Prometheus-Vision: Vision-language model as a judge for fine-grained evaluation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11286–11315, Bangkok, Thailand. Association for Computational Linguistics.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Li Li, Jiawei Peng, Huiyi Chen, Chongyang Gao, and Xu Yang. 2024. How to configure good in-context sequence for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26710–26720.
- Rensis Likert. 1932. A Technique for the Measurement of Attitudes. *Archives of Psychology*, 22(140):1–55.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. [G-Eval: NLG evaluation using GPT-4 with better human alignment](#). In *Proceedings of the 2023 Conference*

- on *Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-Refine: Iterative Refinement with Self-Feedback](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc.
- Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. [Generative Agents: Interactive Simulacra of Human Behavior](#). In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST ’23, pages 1–22, New York, NY, USA. Association for Computing Machinery.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Samarth P, Vyoman Jain, Shiva Golugula, and Motamarri Sai Sathvik. 2025. Manimator: Transforming Research Papers into Visual Explanations. *arXiv preprint arXiv:2507.14306*.
- Sentence Transformers. 2021. all-MiniLM-L6-v2 model card. <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>. Accessed 2026-05-26.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: language agents with verbal reinforcement learning](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.
- Patrick E Shrout and Joseph L Fleiss. 1979. [Intraclass Correlations: Uses in Assessing Rater Reliability](#). *Psychological Bulletin*, 86(2):420–428.
- Ravidu Suen Rammuni Silva, Ahmad Lotfi, Isibor Kennedy Ihianle, Golnaz Shahtahmassebi, and Jordan J Bird. 2026. Training and Agentic Inference Strategies for LLM-based Manim Animation Generation. *arXiv preprint arXiv:2604.18364*.
- The Manim Community Developers. 2026. [Manim – Mathematical Animation Framework](#).
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint arXiv:2305.16291*.
- Xu Yang, Yongliang Wu, Mingzhuo Yang, Haokun Chen, and Xin Geng. 2023. [Exploring diverse in-context configurations for image captioning](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 40924–40943. Curran Associates, Inc.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. [ReAct: Synergizing Reasoning and Acting in Language Models](#). In *The Eleventh International Conference on Learning Representations*.
- Fengji Zhang, Bei Chen, Yue Zhang, Jacky Keung, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, and Weizhu Chen. 2023. [RepoCoder: Repository-level code completion through iterative retrieval and generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2471–2484, Singapore. Association for Computational Linguistics.
- Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. [ExPeL: LLM Agents Are Experiential Learners](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19632–19642.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.
- Shuyan Zhou, Uri Alon, Frank F. Xu, Zhiruo Wang, Zhengbao Jiang, and Graham Neubig. 2023. [DocPrompting: Generating Code by Retrieving the Docs](#). In *The Eleventh International Conference on Learning Representations*.
- Zeyu Zhu, Kevin Qinghong Lin, and Mike Zheng Shou. 2025. [Paper2Video: Automatic Video Generation from Scientific Papers](#). In *NeurIPS 2025 Workshop on Scaling Environments for Agents*.

A Self-Evolving Loop Pseudocode

Algorithm 1 states the per-task procedure described in §3.2. Symbols carried over from the main text: a task $\tau = (s, r, d)$ comprises section text s , scene role $r \in \{\text{BACKGROUND, METHOD, EXPERIMENT, CONCLUSION}\}$, and domain tag d (§3.1); the two retrieval sizes are pinned at $k_+ = 2$ and $k_- = 3$ (§3.3). The render result returned by R exposes a fault

category and a Python traceback, both consumed by LESSONDISTILLER when distilling text-channel pitfalls. VLM additionally emits rubric axes and per-axis issues; these are persisted to the run log for analysis but are not read elsewhere by the loop.

B Implementation Details

This appendix collects per-condition descriptions for the experimental setup in §4.2, together with renderer-sandbox configuration, length caps, and short-circuit rules referenced from §3.2 and §3.3 but kept out of the body for space. All numbers are pinned in run manifests.

Per-condition descriptions. VLM reflection only uses the same renderer, VLM reviewer, and visual-revision loop as MANIMAGENT but disables retrieval and memory writing. **EMB@0** keeps the same prompt structure with empty Reference Examples and Known Pitfalls slots, isolating prompt-structure effects from retrieved content. **EMB@K** retrieves from a frozen self-grown EMB snapshot containing K consolidated records, each produced by the best-of- N delivery rule of §3.2 and the LLM distillation rules of §3.3. **Random EMB** preserves record count, format, and token budget but shuffles records across tasks, breaking content-query alignment.

Reflection-round accounting. For task τ , we define $RR(\tau) = \sum_{i \in \tau} (n_{i,\text{text}}^{\text{retry}} + n_{i,\text{visual}}^{\text{revision}})$. The first term counts retries actually performed after a failed static check or render; the second counts revisions actually performed in response to VLM feedback after a successful render. Initial code generation is excluded. We report the mean of $RR(\tau)$ over all 33 fixed-probe tasks. A scene skipped before a successful render contributes only its executed text retries and zero visual revisions.

Renderer sandbox and static checker. The RENDERER executes each candidate as a subprocess under POSIX rlimit (wall 180 s, CPU 120 s, RAM 4 GB) with the Cairo backend. Before invocation, the static checker rejects forbidden imports (os, subprocess, socket, ...), forbidden calls (open, eval, exec), and missing structure (the script must declare a Scene subclass matching $\sigma_i.\text{name}$ and call `self.play` at least twice). On failure the renderer returns a typed result whose category is python, latex,

manim_runtime, timeout, or unknown, together with a traceback tail and any TeX log excerpt.

The text REVIEWER produces ≤ 60 -word hints; the LESSONDISTILLER is capped at ≤ 1500 output tokens.

$\mathcal{M}^-$ field length caps. Trigger ≤ 400 chars, root cause ≤ 400 , fix recipe ≤ 400 , code fragments ≤ 800 chars each, free-form diagnostic ≤ 1000 chars. The RATIONALEWRITER output stored in $\mathcal{M}^+$ is capped at ≤ 400 chars. Retrieval-time truncations match Table 14.

Short-circuit and tie-break rules. The text-reflection loop exits when $t=T_{\text{text}}=2$, on the REVIEWER’s give_up decision, or when the same error category recurs twice (preventing reflection from looping inside the same kind of bug). The visual-reflection loop exits when $t=T_{\text{vis}}=2$ or on pass/fail; an auto-pass override fires when $u(t) \geq \theta_{\text{conv}} = 90$ regardless of the verdict of the VLM. Best-of- N delivery ships $c^* = \arg \max_t u(t)$ across renderable candidates; ties are broken by earliest t . These rules are enacted by Algorithm 1 in Appendix A.

C Extended Related Work

This section extends §2 with programmatic animation, reflection-based agents, and parameter-free inference-time optimization.

Programmatic animation and visual education. Programmatic animation systems use executable graphics code as an intermediate representation for visual explanations. MANIM provides the rendering substrate for mathematical animations (The Manim Community Developers, 2026), and recent LLM-based systems have begun to generate such animations or videos automatically. Code2Video generates educational videos from code (Chen et al., 2026); Manimator converts research papers into visual explanations (Samarth P et al., 2025); LLM2Manim studies pedagogy-aware generation of STEM animations (Joshi et al., 2026); and Paper2Video broadens the setting to video generation from scientific papers (Zhu et al., 2025). These systems establish the feasibility of paper- or code-conditioned visual explanation, but they primarily treat each input as an isolated generation problem. MANIMAGENT targets the same output medium, but asks whether the agent can retain transferable visual-programming lessons across tasks.

Algorithm 1 The self-evolving loop combines per-scene text and visual reflection, best-of- N delivery, and two distillation paths within a dual-channel EMB.

Require: Task stream $(\tau_1, \tau_2, \dots)$; agents Σ (Storyboarder), Coder, Reviewer, VLM, VisualReviser, RationaleWriter, LessonDistiller; Renderer R with static check; sentence encoder ϕ ; thresholds θ_{conv} , θ_{high} , Δ_{fail} ; budgets T_{text} , T_{vis} .

```

1:  $\mathcal{M} \leftarrow \mathcal{M}^+ \cup \mathcal{M}^- \leftarrow \emptyset$ 
2: for each task  $\tau = (s, r, d)$  in the stream do
3:    $\{\sigma_i\}_{i=1}^n \leftarrow \Sigma(\tau)$  ▷ decompose into scenes
4:   for each scene  $\sigma_i$  do
5:      $e_i \leftarrow \phi([s; r]); E^+ \leftarrow \text{top-}k_+(\mathcal{M}^+, e_i); E^- \leftarrow \text{top-}k_-(\mathcal{M}^-, e_i)$ 
6:      $c \leftarrow \text{Coder}(\sigma_i, E^+, E^-); \text{ textTrace} \leftarrow []$ 
7:     for  $t_{\text{tx}} = 0, \dots, T_{\text{text}}$  do ▷ text-reflection: crash  $\rightarrow$  retry
8:        $\text{result} \leftarrow R(c); \text{ append } (c, \text{result}) \text{ to textTrace}$  ▷ static check, then sandbox render
9:       if result is success then
10:         break
11:       end if
12:        $(\text{hint}, \text{dec}) \leftarrow \text{Reviewer}(\text{result})$ 
13:       if  $\text{dec} = \text{give\_up}$  or same error category twice then
14:         break
15:       end if
16:        $c \leftarrow \text{Coder}(\sigma_i, E^+, E^-, \text{hint}, \text{result})$  ▷ regenerate from scratch
17:     end for
18:     if result is not success then
19:       continue ▷ scene skipped; no usable transition exists
20:     end if
21:      $\text{cands} \leftarrow []; \text{ visTrace} \leftarrow []$ 
22:     for  $t_{\text{vs}} = 0, \dots, T_{\text{vis}}$  do ▷ visual-reflection: renderable  $\rightarrow$  revise
23:        $K \leftarrow R.\text{sample\_keyframes}(c, n_K = 4)$ 
24:        $(\text{axes}, u, \text{issues}, \text{instr}, \text{dec}) \leftarrow \text{VLM}(s, \sigma_i, K)$ 
25:        $\text{append } (c, u, \text{instr}) \text{ to visTrace}; \text{ append } (c, u) \text{ to cands}$ 
26:       if  $u \geq \theta_{\text{conv}}$  then
27:          $\text{dec} \leftarrow \text{pass}$  ▷ auto-pass override
28:       end if
29:       if  $\text{dec} \in \{\text{pass}, \text{fail}\}$  then
30:         break
31:       end if
32:        $c \leftarrow \text{VisualReviser}(c, \text{instr})$  ▷ not the Coder
33:     end for
34:      $(c^*, u^*) \leftarrow \arg \max_{(c, u) \in \text{cands}} u$  ▷ best-of- $N$ 
35:     if  $u^* \geq \theta_{\text{high}}$  then
36:        $\rho \leftarrow \text{RationaleWriter}(s, \sigma_i, c^*)$ 
37:        $\mathcal{M}^+ \leftarrow \mathcal{M}^+ \cup \{(e_i, \sigma_i, \rho, c^*, u^*)\}$ 
38:     end if
39:     for each text transition  $(c^{(t-1)}, c^{(t)})$  in textTrace with  $c^{(t-1)}: \text{err}, c^{(t)}: \text{ok}$  do
40:        $\ell \leftarrow \text{LessonDistiller}(c^{(t-1)}, c^{(t)}, \text{category}, \text{traceback})$ 
41:        $\mathcal{M}^- \leftarrow \mathcal{M}^- \cup \{(e_i, \sigma_i, \ell)\}$ 
42:     end for
43:     for each visual transition  $(c^{(t-1)}, c^{(t)})$  in visTrace with  $u(t) - u(t-1) \geq \Delta_{\text{fail}}$  do
44:        $\ell \leftarrow \text{LessonDistiller}(c^{(t-1)}, c^{(t)}, \text{instr}^{(t-1)})$ 
45:        $\mathcal{M}^- \leftarrow \mathcal{M}^- \cup \{(e_i, \sigma_i, \ell)\}$ 
46:     end for
47:   end for
48: end for
49: return  $\mathcal{M}$ 

```

Reflection and parameter-free optimization.

Self-Refine (Madaan et al., 2023), Reflexion (Shinn et al., 2023), and CRITIC (Gou et al., 2024) revise from feedback within a task, while ReAct connects reasoning with tool feedback (Yao et al., 2023). GEPA (Agrawal et al., 2026) and Training-Free GRPO (Cai et al., 2025) optimize textual control signals from rollouts without weight updates. MANIMAGENT follows the intra-task reflection pattern but uses EMB to distil positive and negative

experiences into a frozen cross-task store retrieved on disjoint probes. These directions are complementary rather than matched retrieval baselines; combining them is future work.

D Comparison with Prior Systems

Table 3 contrasts MANIMAGENT with prior animation-from-paper pipelines (Animator, Code2Video, LLM2Manim) and the closest lifelong-learning agent (Voyager) along four axes:

reflection style, VLM-based judging, cross-task memory, and fixed-probe evaluation. The prior animation pipelines reflect within a task but never carry information across tasks; Voyager carries cross-task memory as a single library of successful executable skills. MANIMAGENT is the only system that combines multi-agent visual reflection, a structured multi-axis VLM judge, a self-grown dual-channel memory, and a fixed-probe snapshot evaluation as the primary metric.

E EMB vs. RAG and Retrieval Terminology

The EMB uses retrieval, but it is not standard retrieval-augmented generation (RAG) with a different name. Table 4 contrasts the two on five axes: corpus, growth, write rule, content, and prompt role. The corpus of the EMB is self-generated task traces; it grows after each converged task under VLM-gated write rules; it stores both success rationales and VLM-screened failure patterns; and it injects them into two distinct prompt slots. A static Manim-code RAG corpus can still strengthen first drafts, which is why §4.2 includes it as a baseline; the fixed-probe experiment then tests whether mature EMB snapshots outperform that static corpus under matched token budget, prompt length, retrieval depth, encoder, and index, so an EMB–RAG gap is evidence for trajectory-derived cross-task memory rather than retrieval alone.

Retrieval terminology. The EMB lookup is implemented as a retrieval layer over a Faiss IndexFlatIP. We call this a retrieval layer in the codebase, an engineering term for the nearest-neighbour lookup component, and it should not be confused with the static Manim-code RAG baseline, which retrieves from a fixed external corpus rather than from a self-grown, polarity-tagged memory bank. Both use the same encoder (all-MiniLM-L6-v2, 384-dimensional embeddings) and the same index type (Faiss IndexFlatIP with cosine similarity), so any retrieval-quality gap reflects differences in retrieved content rather than retrieval machinery.

Long-context baseline scope. The matched-budget controls do not retrieve raw historical code, complete reflection trajectories, or reviewer feedback. A fair long-context comparison would need to hold the source tasks, retriever, candidate count, and final token budget fixed while varying only whether historical information is injected as raw

trajectories or distilled positive and negative experiences. We leave this comparison to future work.

F Dataset Validation

The released dataset is accompanied by an automated validator. It enforces zero paper overlap across the memory-building, fixed-probe, and cross-test splits; zero overlap between model-input and evaluation-only fields; valid source-text hydration for all headline tasks; distribution match across split strata; contamination and license metadata; snapshot plausibility for $K \in \{0, 50, 100, 200\}$; and the release policy excluding raw full-paper text and embedded human scores.

Strict paper-readiness validation additionally requires the human-scores file to be non-empty and should be run after output-level annotations and manifests have been collected. The dataset is release-ready before this final result-validation pass.

Within-stream duplication. Paper-level isolation prevents overlap between the memory-building and probe sets but does not rule out semantically near-duplicate tasks within the memory-building stream. The current validator records contamination metadata but does not perform paraphrase-level deduplication; this remains future work.

G Released Dataset Summary

Table 5 summarises the released artefacts, and Figure 6 visualises the relative split sizes: the three headline subsets (memory-building 39, fixed-probe 33, cross-test 40) are deliberately small relative to the quarantined holdout of 195 papers, which is excluded from every reported number. The release includes dataset metadata, headline tasks, quarantined holdout tasks, and paper metadata. Raw full-paper text, draft annotations, example experiment manifests, and output-level human scores are excluded from the model-visible release. The release manifest states licensing terms for system-generated code, videos, and EMB snapshots and preserves original notices and source attribution for Manim documentation, tutorials, and code examples.

G.1 Representative Fixed-Probe Examples

Table 6 gives representative records sampled from the human-evaluation candidate pool in the fixed-probe split. The selection covers all five dataset domains, all four scene roles, and all three difficulty

System	Reflection	VLM judge	Cross-task memory	Fixed-probe test
Manimator (Samarth P et al., 2025)	–	–	–	–
Code2Video (Chen et al., 2026)	multi-agent	VLM critic	–	–
LLM2Manim (Joshi et al., 2026)	compile-error	–	–	–
Voyager (Wang et al., 2023)	–	–	self-grown executable skills	partial
MANIMAGENT	multi-agent, visual	structured, multi-axis	dual-channel, self-grown	primary metric

Table 3: **MANIMAGENT combines self-grown dual-channel memory with fixed-probe evaluation.** Prior systems either lack cross-task memory, retain a single channel of successful executable skills, or do not evaluate with held-out snapshots.

Axis	Standard RAG	EMB (ours)
Corpus	fixed docs/tutorials/code	self-generated task traces
Growth	static during evaluation	grows after converged tasks
Write rule	none	VLM-gated success and failure writes
Content	human-written references	success rationales plus VLM-screened failure patterns
Prompt role	one context slot	Reference Examples and Known Pitfalls

Table 4: **EMB differs from static RAG in source, growth, write rule, content, and prompt role.**

levels. At inference time the system receives the paper title, target-unit text and anchor, domain, and scene role in the model input; human ratings and output annotations remain evaluation-only. Table 6 lists representative fixed-probe tasks for coverage; Table 7 shows a schema-complete example record.

G.2 Example Dataset Record

Table 7 gives an illustrative, shortened record whose field groups match the benchmark and private evaluation-sidecar schema. Long section excerpts are abbreviated and no raw full-paper text is reproduced. Evaluation-only annotations are structurally shown to make isolation explicit; they are never presented to the generation system or written to the EMB.

H Human Annotation Protocol

The main quality metrics are based on blind human scoring of first-attempt videos. Human scores are stored separately from the model-visible task file and are never embedded in task records or shown to any system. The release layout deliberately keeps collected output-level annotations outside the model-visible task collection.

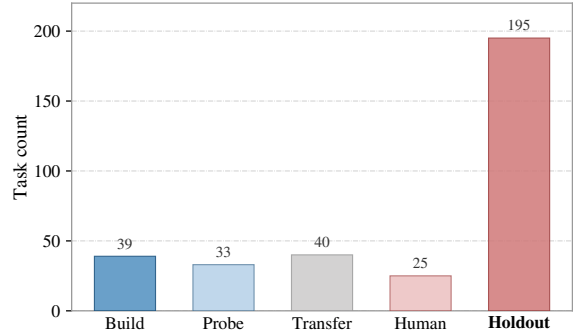


Figure 6: **Dataset split sizes are deliberately small for the three reported subsets.** The memory-building, fixed-probe, cross-test, and human-scoring subsets support construction and evaluation; the quarantined hold-out is excluded from headline claims.

Raters. 54 raters take part in the blind human evaluation, and each evaluated video receives two or three independent ratings. Raters are recruited from graduate programmes in computer science, applied mathematics, physics, and economics. No rater has prior exposure to MANIM code generation or to the specific papers in the evaluation stream. 77% identify CS/AI as their primary discipline, 11% economics/social science, 5% mathematics/statistics, and 5% physics/quantum. 42% report frequently reading research papers; 64% have occasional prior experience evaluating technical diagrams or animations, while 33% have regular experience. Before main annotation, each rater completes a one-hour calibration session on 5 held-out calibration examples covering the full quality range.

Ethics and data retention. Raters are anonymous, uncompensated volunteers. We collect no names, contact information, or other direct identifiers. They assess only non-sensitive scientific animations. Raw ratings are stored temporarily on an author-only rented server and deleted within 14 days. The study is not conducted under a formal institutional framework.

Split	Size	Released fields	Role
Memory-building split	39	Identifiers, target-unit anchors, metadata, scene roles	Grows the EMB before snapshot freezing
Fixed-probe split	33	Identifiers, target-unit anchors, metadata, evaluation-only rubrics	Evaluates every frozen EMB snapshot in read-only mode
Cross-test split	40	Identifiers, target-unit anchors, domain tags, metadata	Exploratory transfer probe
Human-scoring subset	25	Candidate task identifiers; output-level ratings are stored separately after annotation	Blind Human Pass@1 and Human quality
Quarantined hold-out	195	debug metadata only (177 fallback, 18 promoted)	Excluded from all headline claims

Table 5: **The released dataset separates construction, evaluation, and exploratory transfer.** The benchmark separates memory construction, fixed-snapshot evaluation, and exploratory transfer testing; human scores remain separate from model-visible inputs.

Task identifier	Domain	Scene role	Difficulty	Paper / target-unit anchor
2305_18290_section_experiments	CS	Experiment	Easy	Direct Preference Optimization / Experiments
1609_07132_section_method	Math	Method	Medium	Optimization Methods for Large-Scale Machine Learning / Method
1207_7235_section_results	Physics	Experiment	Hard	Quantum Entanglement and Topological Order / Results
quant-ph_0508027_section_introduction	Quantum	Background	Easy	A One Way Quantum Computer / Introduction
1706_01427_section_discussion	Economics	Conclusion	Hard	Machine Learning: An Applied Econometric Approach / Discussion

Table 6: **Five representative fixed-probe tasks anchor the human-evaluation pool.** The examples are taken from the released benchmark and illustrate domain, scene-role, and difficulty coverage without reproducing full paper text.

Background questionnaire. Before rating any videos, each rater completes a three-item background form: (i) primary discipline (computer science/AI, mathematics/statistics, physics/quantum, economics/social science, or other/no specific background); (ii) frequency of reading research papers or technical literature (rarely/never, occasionally, or frequently); and (iii) prior experience evaluating technical diagrams, animations, or other visualizations (none, occasional viewing/evaluation, or frequent creation/use/evaluation). These responses characterize the rater pool and are not used as model inputs.

Inputs shown to raters. For each sample, raters see the paper title, target-unit title and text, scene role, key claims, reference scene plan, task-specific rubric, and generated first-attempt video. The reference scene plan is guidance, not a required script. Raters do not see the system condition, EMB size, VLM score, prompt, code, revision trace, or any

filename that reveals the condition.

Sampling plan. Human-scoring tasks are sampled from successfully hydrated fixed-probe tasks after target annotations have been human-audited. We sample 25 tasks, stratified by scene role (BACKGROUND, METHOD, EXPERIMENT, CONCLUSION), domain (five domains), and difficulty (three bins: number of distinct visual elements, number of animation stages, presence of equations). Each video receives three independent ratings. The main-table plan covers 7 conditions: VLM reflection only, Manim-code RAG, Random EMB, EMB@0, EMB@50, EMB@100, and EMB@200. Applied to 25 tasks, this yields 175 evaluated videos. Raters may evaluate videos from multiple conditions. Output-level ratings for all conditions are intentionally kept outside the model-visible dataset.

Questionnaire and rating dimensions. For each video, the form first asks the binary Human Pass@1

Visibility group	Field	Illustrative shortened value
Model-visible input	Paper title	<i>Attention Is All You Need</i>
	Abstract excerpt	Abstract field present; shortened excerpt where available
	Target-unit anchor	Introduction
	Target-unit excerpt	“Recurrent models typically factor computation along symbol positions . . .”
	Required prior context	Not required
	Prior objects	None
	Scene role	Background
	Domain	Computer science
Evaluation-only fields	Source type	Section
	Key claims	Human-audited local claims in a private evaluation sidecar
	Reference scene plan	Expected pedagogical scene progression
	Task-specific rubric	Alignment, robustness, coverage, flow, and usability criteria
	VLM diagnostic rubric	Logic-flow, layout/occlusion, and accuracy checks
	Condition identifiers	Hidden system condition and snapshot identifier
Metadata / release	Fatal issue flags	Closed-set annotation options used by blind raters
	Split	Fixed-probe split
	Paper / section identifier	1706.03762 / Introduction
	Licence metadata	arXiv non-exclusive distribution licence; recorded source URL
	Contamination metadata	Pre-cutoff publication stratum; cutoff reference 2023-10-01
Human outputs	Quarantine metadata	Hydration status valid; included in headline probe
	Ratings sidecar	Human Pass@1 votes, Likert ratings, flags, and optional notes are stored separately from model-visible records.

Table 7: **This illustrative benchmark record covers the full schema.** The example shows model-visible inputs, evaluation-only fields, and release metadata while shortening textual content and isolating output-level human annotations.

question, then five 1–5 Likert items (1=very poor, 3=average, 5=very good):

- **Visual design.** Are composition, colour, information hierarchy, and overall visual presentation clear and coherent?
- **Key-claim coverage.** Are the central claims visually expressed?
- **Visual robustness.** Are elements readable, free of overlap, and correctly rendered?
- **Animation flow.** Is the pacing, ordering, and transition between scenes appropriate?
- **First-attempt usability.** Could this video be used as a paper-section animation without major structural revision?

Table 8 summarises aggregation of these ratings. Human quality is the mean of the five dimensions, equally weighted and averaged across raters. The dimensions are correlated but capture distinct aspects: visual design and visual robustness concern visual presentation; key-claim coverage concerns content coverage; animation flow concerns temporal presentation; and first-attempt usability is a holistic readiness judgement. Figure 7 shows

the per-dimension Likert distribution for the three anchor conditions (EMB@0, Manim-code RAG, EMB@200); the dimension-level story matches the headline mean, with EMB@200 lifting both the median and the lower-quartile floor over the two baselines.

Human Pass@1. Raters answer a binary usability question: “Is this first-attempt video usable as a paper-section animation without major repair?” A major repair is defined as any change requiring structural reorganization of the scene plan or a complete rewrite of the animation code. Human Pass@1 is computed from all collected binary judgements within each condition. All submitted blind ratings are retained without majority voting or post hoc adjudication.

Fatal flags. Raters additionally mark zero or more fatal issue flags from a closed set: unrelated to the target, major factual or formula error, hallucinated claim, missing central claim, severe overlap or occlusion, cropped or off-screen object, unreadable text or labels, confusing animation order, overly static presentation, excessive text, broken or unplayable output, and other. An optional free-text note records additional observations; selecting other requires such a description.

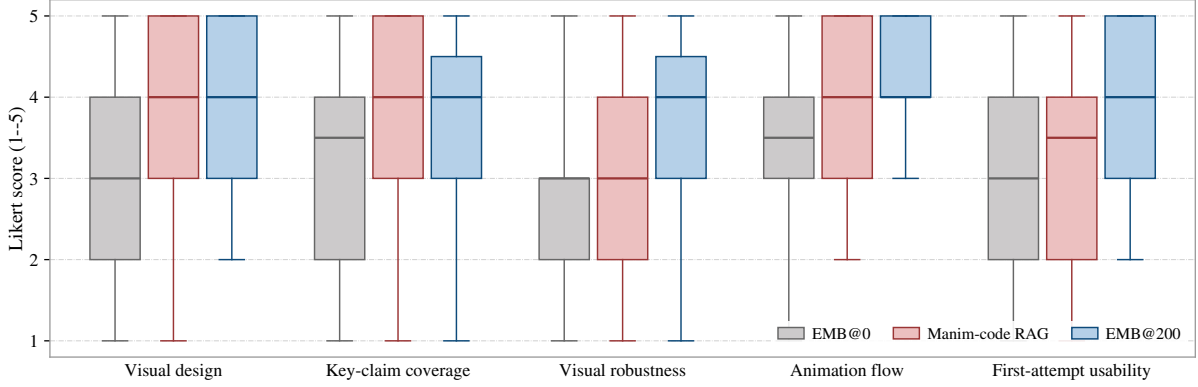


Figure 7: **Per-rater Likert scores spread across the five human-evaluation dimensions for three representative conditions.** Box distributions are derived from per-condition per-dimension rater scores (EMB@0 $n=50$, Manim-code RAG $n=54$, EMB@200 $n=53$). EMB@200 dominates EMB@0 on every dimension and lifts the lower-quartile floor relative to Manim-code RAG, indicating that the headline mean is not driven by a few high-scoring raters.

Human-scoring item	Aggregation
Usability decision	proportion of individual pass votes
Visual design	mean score
Key-claim coverage	mean score
Visual robustness	mean score
Animation flow	mean score
First-attempt usability	mean score
Human quality	mean of five dimensions
Fatal issue flags	frequency

Table 8: **Human scoring combines a binary item, five Likert dimensions, and fatal flags.** Each rating includes a binary usability judgement, dimensional ratings, and optional fatal-issue flags.

Inter-rater agreement. Fleiss’ κ for binary usability votes is -0.01 , and binary full agreement is 59.4% . Intraclass correlation (ICC) for the five Likert dimensions is low across all dimensions: $\text{visual_design} = 0.24$, $\text{key_claim_coverage} = 0.00$, $\text{visual_robustness} = 0.10$, $\text{animation_flow} = 0.33$, and $\text{first_attempt_usability} = -0.09$. All dimensions fall below common reliability thresholds ($\kappa \geq 0.4$, $\text{ICC} \geq 0.5$), reflecting the subjective nature of animation quality assessment and the disciplinary diversity of the rater pool. We retain all independent blind ratings without exclusion, majority voting, or post hoc adjudication. Because the original rater–video-level records were deleted after the 14-day retention period, condition-specific denominators and paired uncertainty intervals cannot be reliably reconstructed from the retained aggregate results.

I VLM Scoring Schema and VLM–Human Agreement

The VLM scores rendered keyframes to drive revision and to gate memory writes. Because the same VLM serves both roles, VLM scores are not used as headline quality evidence. They are reported here as auxiliary diagnostics.

VLM scoring schema. Table 9 defines the three axes used by the VLM reviewer: logical flow, layout/occlusion, and accuracy. The aggregate VLM score $u(t)$ is the unweighted mean of the three per-axis scores, each on $[0, 100]$. These scores drive visual revision (the loop terminates when $u(t) \geq \theta_{\text{conv}} = 90$ or $t = T_{\text{vis}} = 2$) and memory consolidation. The $\mathcal{M}^+$ write threshold is set to $\theta_{\text{high}} = 85$ (slightly below the convergence threshold, so that a scene can be judged good enough to remember without having reached the auto-pass bar). The $\mathcal{M}^-$ write gate requires an improvement of at least $\Delta_{\text{fail}} = 5$ between adjacent attempts. The quality conclusions of the main paper are based on human ratings, not VLM scores.

Evaluator leakage and self-confirmation risk.

The VLM that scores keyframes also gates memory writes, so systematic VLM biases could be consolidated into the EMB as self-confirming patterns. We mitigate this in three ways. (1) Headline quality evidence is always blind human judgement, stored in a separate file never shown to any system. (2) The negative-channel write gate requires a locally improving adjacent attempt pair with a margin $\Delta_{\text{fail}} = 5$, providing an operational improvement

Dimension	What the VLM evaluates
Logical Flow	Whether scene ordering follows the pedagogical narrative and animation transitions are not abrupt.
Layout / Occlusion	Whether elements overlap, exceed the canvas, or are hidden behind formulae.
Accuracy	Whether formulae are correct and the visualisation does not invite misinterpretation.
Aggregate score	Unweighted mean of the three axes; used only as an internal repair and memory-write signal.

Table 9: **Three VLM axes drive revision and memory writes.** The table defines the logical-flow, layout/occlusion, and accuracy dimensions scored on $[0, 100]$. Their unweighted mean is a fixed internal control signal, not a substitute for the reported blind human quality metric.

filter. Here, “validation” means only that a revision receives a better internal VLM evaluation than its preceding version; it does not imply human validation or rigorous causal identification. (3) The weak-VLM ablation (Table 1) quantifies sensitivity to the choice of reward model.

VLM axis independence. The three VLM axes are aggregated by unweighted mean, chosen a priori. We verify the aggregate signal against blind human ratings after collection (Table 10). Because aggregate agreement is weak, we do not use this validation to claim either axis reliability or score calibration; the axes remain internal heuristics designed to cover complementary animation failure modes.

Prompt sensitivity. Retrieval-query wording and prompt-slot formatting are fixed engineering choices that we do not ablate. The weak-VLM condition in Table 1 bounds sensitivity to the reward model but cannot eliminate reviewer blind spots shared across VLM variants.

Agreement analysis. The agreement analysis uses 37 videos for which both VLM scores and completed three-rater human annotations are available after filtering missing or failed renders. The VLM decision for each video is paired with its three human votes. We compute Pearson r and Spearman ρ between aggregate VLM score and Human quality, and Cohen’s κ between VLM pass/fail (thresholded at 90) and individual human binary usability votes. The weak, slightly negative agreement values reported in Table 10 support using blind human ratings for headline claims and treating VLM

Agreement statistic	Value
Pearson r (VLM aggregate vs. Human quality)	−0.17
Spearman ρ (VLM aggregate vs. Human quality)	−0.20
Agreement rate (VLM pass/fail vs. Human Pass@1)	0.49
Cohen’s κ (VLM pass/fail vs. Human Pass@1)	−0.07

Table 10: **VLM scores are not reliable headline evaluation evidence.** On 37 paired videos, agreement with blind human ratings is weak or slightly negative; VLM scores are therefore used only as an internal repair/write signal.

Component	$\mathcal{M}^+$ (success)	$\mathcal{M}^-$ (failure)
context (shared)	s, r, d , paper id, section id	
provenance (shared)	run id, scene id, source, ordinal, before/after scores	
body	rationale, final code, frame hash	trigger, root cause, fix recipe, anti/good example, diagnostic
inject as	Reference Examples (soft)	Known Pitfalls (hard)
write condition	best candidate meets positive-memory score threshold (85)	render succeeds after error, or visual score improves by at least 5 points
distilled by	Rationale ($T = 0.3$)	Writer Lesson ($T = 0$)

Table 11: **The EMB separates shared context from channel-specific bodies and writes.** Indexing stays channel-agnostic (one SQLite table, one retrieval interface), while two per-polarity Faiss indices preserve the soft-template / hard-exclusion split. Lessons and rationales are produced by dedicated LLM distillers rather than stored as literal before/after diffs.

scores only as an internal repair/write signal.

J EMB Schema

Table 11 reports the field-level schema of the dual-channel EMB referenced in §3.3: which fields are shared across polarities, which differ, and how each channel is written and injected.

K Coder Prompt Template

The Coder prompt is assembled per scene by concatenating four blocks: fixed system instructions, the top- k_+ retrieved positive memories formatted as Reference Examples, the top- k_- retrieved negative memories formatted as Known Pitfalls, and the scene plan from the STORYBOARDER. Retrieval depths ($k_+ = 2, k_- = 3$) and length caps match

Table 14, and per-channel content is drawn from the fields of Table 11. When either channel returns fewer than the requested k entries, the corresponding block is rendered empty, which recovers the EMB@0 prompt structure exactly.

The Reference-Examples block is a soft template: the Coder may deviate when the retrieved code does not fit the current scene. The Known-Pitfalls block is treated as a hard exclusion list to be avoided verbatim.

Coder LLM: GPT-5.5 (see Table 14).

System Prompt

You are an expert Manim Community Edition v0.20 engineer. You write **complete, runnable** Python files that render with `manim render -ql` on a headless Linux box.

Output requirement (strict). Return **exactly one** Python code block fenced with `“python ...”`. The block must define **one** class that subclasses `Scene` whose name **exactly matches** the name field of the requested scene; do not include `if __name__ == “__main__”`. No extra prose, no explanations, no second code block.

Mandatory imports / setup. Use `from manim import *`. Do not import `manim.opengl` or use `-renderer=opengl` features. The renderer is `cairo`.

Hard constraints. Violations prevent rendering.

- The target host has **no display server**. Avoid OpenGL-only Mobjects (`OpenGLMobject`, `ThreeDScene` GPU effects).
- Use simple LaTeX. Allowed packages: `amsmath`, `amssymb`, `mathtools`. Avoid exotic macros.
- No external image files. No fonts beyond what Manim ships with. No network calls.
- Keep total scene duration close to `duration_hint` ($\pm 20\%$), and cap mobject count: prefer ≤ 30 simultaneously visible mobjects.
- Use `self.play(...)` for transitions and `self.wait(t)` between transitions.

Recommended patterns.

- Use `MathTex(r"a^2 + b^2 = c^2")` (raw string) for equations and `Tex(r"...")` for prose with LaTeX.

- For colored emphasis, pass token lists into `MathTex` with the `color` keyword, or use `.set_color_by_tex(...)`.
- Group with `VGroup(...)` and arrange via `.arrange(DOWN, buff=0.5)`; position via `.to_edge(UP)`, `.shift(...)`, or `.next_to(...)`.
- Default animations: `Write`, `Create`, `FadeIn`, `FadeOut`, `Transform`, `ReplacementTransform`, `Indicate`. Default to plain `Scene`.

Self-check before responding. Class name matches the requested `scene.name`; every LaTeX backslash is escaped (raw strings); a final `self.wait(...)` is present; no OpenGL* or image / sound / file IO calls.

If a previous attempt is shown with an error, **fix the specific error indicated**; do not change unrelated parts of the code.

Reference Examples and Known Pitfalls. You may receive two extra sections in the user message. *Reference Examples* are past scenes that scored highly on the visual rubric; treat them as guidance, not as boilerplate. *Known Pitfalls* are write-gated failure→success transitions from previous runs; each lists a trigger pattern, root cause, an anti-example, and a good example, and the anti-example pattern **must be avoided** in the output. Both sections are optional and may be empty.

User Prompt Template

```
Scene to render:
{SCENE_PLAN}
Reference Examples (may be empty):
{REFERENCE_EXAMPLES}
Known Pitfalls (may be empty):
{KNOWN_PITFALLS}
```

The placeholders are filled per scene: `{SCENE_PLAN}` is the storyboarder output (name, claim, evidence, takeaway, `duration_hint`); `{REFERENCE_EXAMPLES}` holds $k_+=2$ entries from $\mathcal{M}^+$ (rationale plus code excerpt); `{KNOWN_PITFALLS}` holds $k_-=3$ entries from $\mathcal{M}^-$ (trigger, root cause, fix recipe, anti-/good example pair). Field schemas follow Table 11.

L Supplementary Ablations

This appendix provides the per-variant mechanism rationale and additional interpretation for the ablation results reported in §4.4. All variants are evaluated on the same fixed probe in read-only mode.

Channel isolation. Removing $\mathcal{M}^+$ or $\mathcal{M}^-$ individually tests whether the two channels are complementary: if so, single-channel variants should fall between the full EMB and the no-memory baseline rather than match either extreme. Removing both recovers the no-memory baseline (Table 1).

Write gates. Positive memories require a score of at least $\theta_{\text{high}} = 85$; failure memories require either a repaired render crash or a visual-score increase of at least $\Delta_{\text{fail}} = 5$. The ungated failure-memory variant tests whether the gate is load-bearing: if storing every failed-attempt diagnostic preserves quality, the gate is decorative; if Pass@1 degrades, the improvement-margin requirement is doing real work in keeping $\mathcal{M}^-$ free of random VLM noise.

Retrieval depth. Top-1 retrieval tests whether one nearest neighbour per channel suffices, so gains that survive top-1 retrieval cannot be attributed to context-padding effects. Random EMB (Table 1) preserves record count, format, and token budget while breaking content–query alignment, isolating content from format.

Rationale-only success memory. This variant removes final code excerpts from success-memory retrieval while retaining the LLM-distilled rationale, testing whether the positive-channel benefit reduces to copying past code rather than to the higher-level lesson encoded in the rationale.

Matched-token RAG baselines. We compare retrieval from Manim documentation alone with retrieval from both documentation and code under the same budget, encoder, and index used for EMB. The sweep characterises what kind of retrievable content suffices and isolates the EMB gain from any static-corpus retrieval gain.

Weak-VLM sensitivity. Replacing the reviewer with a weaker VLM (Step-3.7-Flash) tests whether the improvement depends on one particular internal scoring model: results that survive a weaker reviewer cannot be attributed to the idiosyncrasies of a single VLM. Blind human ratings remain the reported quality evidence.

Forest-plot summary. Figure 8 aggregates the per-variant deltas in a single panel: the channel asymmetry (removing $\mathcal{M}^+$ hurts Pass@1, removing $\mathcal{M}^-$ inflates reflection rounds) is the most visible signal, and the two matched-budget RAG vari-

ants close most of the Pass@1 gap but stay high in reflection rounds.

M Snapshot Position Curve

The fixed-probe snapshot experiment (§4.3) is the headline result because it controls for task-order effects. For completeness, we align its four snapshot measurements with their positions in the memory-building stream.

Protocol. MANIMAGENT processes the memory-building split sequentially. Before each task, the current EMB is available for retrieval; after convergence, new entries may be written back. The fixed-probe evaluation uses a separate run with a frozen EMB snapshot. Figure 9 aligns fixed-probe Human Pass@1 at the four frozen snapshot sizes with the positions at which those snapshots were taken in the memory-building stream.

Interpretation. The curve visualises when each frozen bank size was reached; its quality values remain fixed-probe measurements rather than online evaluations of memory-building tasks. The fixed-probe snapshot result is the primary evidence.

N Qualitative Inspection of Retrieved Memories

An exploratory probe of what kind of information the EMB stores. This is not headline evidence.

We manually label the top- k retrieved $\mathcal{M}^+$ entries for a stratified sample of probe queries, categorizing each as domain-specific, layout/pacing/formula-rendering, or mixed. In this inspection, 83.3% of top- k matches concern layout, pacing, or formula rendering, 16.7% are mixed, and domain-specific matches are absent, indicating that the positive channel primarily transfers visual-presentation lessons rather than domain-specific content.

O Qualitative Trace Analysis

We additionally inspected one complete test run directory for the Introduction section of a BERT paper source (arxiv-src:1810.04805). This inspection is a trace-level diagnostic, not headline evidence: it uses the saved summary.json, storyboard.json, trace.jsonl, rendered keyframe montages, and per-scene render manifests to understand the strengths of the generated paper-section animation.

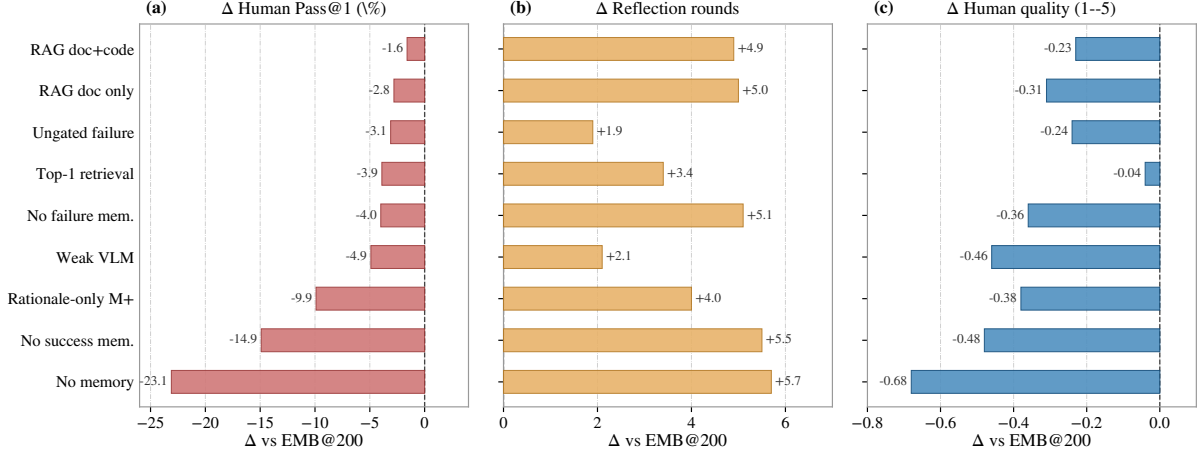


Figure 8: **Per-ablation deltas measure the gap from each variant to Full EMB.** Each row is one variant of Table 1; bars show the gap to Full EMB on each metric (negative = worse for Pass@1 / quality, positive = worse for reflection rounds, since fewer rounds is better). Removing $\mathcal{M}^+$ collapses Pass@1 (left panel) more than removing $\mathcal{M}^-$, while removing $\mathcal{M}^-$ inflates reflection rounds nearly as much as removing memory entirely (middle panel), supporting the asymmetric channel-role reading in §4.4. Matched-budget RAG variants (top two rows) close most of the Pass@1 gap but stay high in reflection rounds. Point deltas computed from Table 1.

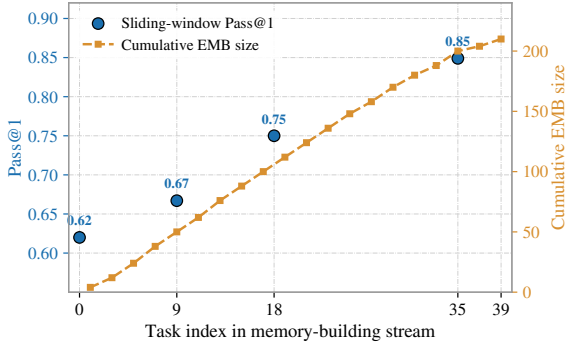


Figure 9: **MANIMAGENT evolves online across the memory-building stream.** Left axis: Human Pass@1 at the four frozen snapshot points (EMB@0, EMB@50, EMB@100, EMB@200). Right axis: cumulative EMB size interpolated across the 39-task memory-building stream, anchored at the four snapshot sizes and the final EMB size (≈ 210 records).

The summarizer extracted the concepts emphasized by the source introduction: bidirectional pre-training, masked language modeling, next-sentence prediction, the Transformer encoder, and fine-tuning. The storyboard converted these concepts into a teaching sequence that introduces BERT, contrasts left-to-right and bidirectional context, visualizes masked-token prediction, and closes with a fine-tuning takeaway. The final rendered output gives a compact visual explanation of how bidirectional context and reusable representations support downstream NLP tasks.

Table 12 summarises the four rendered scenes,

pairing the first-attempt and final VLM axis scores with a short qualitative strength of each delivered render.

This trace shows how the pipeline turns an abstract NLP introduction into a coherent visual narrative. The title scene establishes the model identity and main conceptual hooks; the bidirectional-context scene makes the central architectural contrast concrete through opposing arrow patterns; the masked-language-modeling scene animates the pre-training objective as a visible prediction process; and the fine-tuning scene connects the shared encoder to multiple downstream task heads. Across the rendered scenes (Figure 10), visual reflection mainly improves readability: long text is broken into stable blocks, arrows and labels become easier to parse, and the final takeaway is given stronger visual emphasis.

P Reproducibility Details

Table 14 lists the fixed hyperparameters used in the main experiment. The two VLM thresholds ($\theta_{\text{high}} = 85$, $\theta_{\text{conv}} = 90$) and the failure-channel improvement margin ($\Delta_{\text{fail}} = 5$) are the only memory-write knobs; the retrieval depths ($k_+ = 2$, $k_- = 3$) and snapshot sizes $\{0, 50, 100, 200\}$ are pinned across all EMB conditions.

Table 13 reports approximate per-task resource use. LLM and VLM token budgets are stable across conditions (within roughly $\pm 10\%$); the main cost gradient is wall-clock render time, which rises from

Scene		Trace outcome	VLM axes		Qualitative strength
Title		Rendered; two visual revisions; final pass	78/58/94 92/91/95	→	The final version splits the long title across two balanced lines, centers the encoder stack, and presents the two core cues of BERT—bidirectional context and fine-tuning—as an immediately readable overview.
Bidirectional text	con-	Rendered and passed visual review	92/88/96		The left-to-right arrows and two-sided mask context directly match the bidirectionality claim of the paper, making the contrast between unidirectional and bidirectional pre-training visually explicit.
Masked LM		Rendered; two visual revisions	72/83/58 84/88/62	→	The revised scene groups surrounding context with clear green arrows and highlights the predicted token, turning the masked-language-modeling objective into a step-by-step visual explanation.
Fine-tuning away	take-	Rendered; two visual revisions	82/76/88 88/82/86	→	Larger task boxes, a task-count cue, and a boxed takeaway improve the hierarchy from pre-trained BERT to downstream heads and make the final message easier to read.

Table 12: **One BERT test run shows positive qualitative progress across scenes.** Scores are reported as logical flow / layout-occlusion / accuracy for the first and final renderable versions recorded in the trace.

19 minutes at EMB@0 to 31 minutes at EMB@200 as retrieved scenes prompt more elaborate renders.

Deployment-scale memory management. We evaluate the EMB only through 400 entries ($\mathcal{O}(10^2)$). Production-scale retrieval ($\mathcal{O}(10^6)$) is likely to require hierarchical indexing, semantic deduplication, consolidation of similar memories, utility-based eviction, and dynamic forgetting. Write gates defined on the current VLM scores also do not prevent memory rot after an LLM or VLM upgrade, and per-task cost continues to depend on the reflection budgets.

A complete run manifest records: the LLM and VLM model identifiers with version tags, max output tokens, retrieval configuration (k_+ , k_- , index type, embedding model), renderer version, random seed, evaluation split, and ablation preset.

Condition	LLM tokens	VLM tokens	Render time
Manim-code RAG	38K	23K	27 min
Random EMB	38K	23K	25 min
EMB@0	37K	23K	19 min
EMB@100	37K	22K	23 min
EMB@200	36K	20K	31 min

Table 13: **Per-task cost stays comparable across conditions.** Token counts estimated from output characters and review-call counts; no explicit token tracking was enabled. Render time is wall-clock duration per task.

Pre- vs. post-cutoff stratification. To assess whether gains are driven by memorised paper text

rather than by EMB retrieval, Table 15 stratifies probe tasks by publication date relative to the training cutoff of the LLM. A large gap favouring pre-cutoff papers would suggest memorised source text; a small or non-systematic gap supports EMB-mediated transfer. The observed deltas (Pass@1 -5.9% pre minus post, quality $+0.17$) point in opposite directions and are small, ruling out a consistent pre-cutoff advantage and so supporting the EMB-mediated reading.

Stratum	Human Pass@1	Human quality
Pre-cutoff papers	74.1%	3.56
Post-cutoff papers	80.0%	3.40
Δ (pre – post)	-5.9%	$+0.17$

Table 15: **Stratifying by LLM training cutoff shows no consistent pre-cutoff advantage.** A small gap supports EMB-mediated transfer over memorization.

Manim-code RAG corpus. The fixed corpus contains 360 Manim API reference entries, 10 standalone code examples from the official examples page, and one quick-start tutorial excerpt, totalling approximately 370,000 characters (54,000 tokens). The API documentation mainly covers `manim.mobject`, `manim.utils`, `manim.animation`, `manim.scene`, and `manim.camera`; the examples include coordinate axes, bar charts, equation derivations, and three-dimensional surfaces. Median entry length is 629 characters and the maximum is 6,000. Each entry records its source provenance and MIT License, with original notices and attribution



Figure 10: **Detailed qualitative visual revisions in one BERT test run improve readability across scenes.** This appendix figure keeps the original long-strip keyframe montages at a readable width. It compares the first renderable version and final recorded version for three rendered scenes: the title scene becomes a balanced overview, the masked-language-modeling scene makes context aggregation and token prediction clearer, and the fine-tuning scene gives the downstream-task hierarchy and final takeaway stronger visual emphasis.

Parameter	Value	Parameter	Value
Convergence threshold (auto-pass)	90	$\mathcal{M}^+$ rationale cap	600 chars
$\mathcal{M}^+$ write threshold	85	Snapshot sizes $\mathcal{K}$	$\{0, 50, 100, 200\}$
$\mathcal{M}^-$ improvement margin	5	Sentence encoder ϕ	all-MiniLM-L6-v2
Text-reflection retry budget T_{text}	2	Vector index	Faiss IndexFlatIP
Visual-reflection revision budget T_{vis}	2	Coder LLM	GPT-5.5
Positive retrieval depth k_+	2	Reviewer VLM	GPT-5.5
Negative retrieval depth k_-	3	Weak-VLM ablation	Step-3.7-Flash
$\mathcal{M}^+$ code-excerpt cap	1200 chars	MANIM version	Community v0.20.1

Table 14: **All numerical hyperparameters are pinned across conditions.** Encoder ϕ outputs 384-dimensional vectors; the Faiss index applies inner product to L_2 -normalised embeddings (equivalent to cosine similarity). Caps are enforced at retrieval time. Values match code defaults and run manifests.

preserved. Retrieval uses the same encoder, Faiss vector-index configuration, and context budget as the EMB.

Q Retrieval-to-Revision Case Study

We trace one fixed-probe example from memory retrieval to code revision. The task is to animate self-attention over image tokens: the scene contains

a row of token rectangles labelled x_1 through x_6 , curved attention arrows, a Transformer encoder stack, and a formula. In the initial render, the dense token row and numerous curved arrows are difficult to read: token labels are small, the arrows are thin and visually uniform, and the attention arcs do not stand out from the rest of the scene.

Retrieved memories. The positive channel retrieves success record 14872d25 with cosine similarity 0.781 from arxiv:2010.11929 / Introduction / TransformerAttention; its internal VLM score is 87.0. The record describes a successful left-to-right layout that separates tokens, attention arcs, the encoder stack, and the formula. It also recommends staged FadeIn/Create animation and high-contrast yellow-green arcs above blue tokens to make global attention readable.

The negative channel retrieves failure record 2c4c3609 with cosine similarity 0.781 from the same TransformerAttention task. This record captures a revision whose internal VLM score increased from 75.33 to 87.0. Its trigger is a dense token row that must be read together with many curved arrows and the encoder stack or formula. The diagnosed failure pattern is small token labels combined with uniform thin arcs; the stored remedy is to enlarge the tokens and text, thicken the arrows, vary their curvature, and keep the arcs above the background with a higher z-index.

Resulting code revision. After these memories are injected into the reflection prompt, the system makes the following representative changes to the Manim code:

```
- rect = RoundedRectangle(width=0.72, height=0.58,
-                         stroke_width=3)
- tex = MathTex(label, font_size=28)
- arc = CurvedArrow(a, b, angle=-PI/2.35,
-                  stroke_width=3)
- arc.set_z_index(1)
+ rect = RoundedRectangle(width=0.92, height=0.70,
+                         stroke_width=3.2)
+ tex = MathTex(label, font_size=34)
+ for k, (i, j) in enumerate(conns):
+     arc = CurvedArrow(a, b,
+                       angle=-PI/(2.25 + 0.12*k),
+                       stroke_width=3.6)
+     arc.set_z_index(2)
```

The retrieved $\mathcal{M}^+$ entry supplies the left-to-right organization, staged animation, and high-contrast attention arcs, while the $\mathcal{M}^-$ entry helps avoid small labels, uniform thin arcs, and weak visual separation between attention links and the rest of the scene. After the revision, the internal VLM score increases from 75.33 to 87.0, with clearer

layout, labels, and attention-link salience. This is an internal trace-level diagnostic rather than an independently human-rated result.

This example also illustrates the memory representation: the EMB does not store complete historical code or full conversations. It stores distilled positive and negative experiences, retrieves those relevant to the current task, and uses them to guide code reflection and revision.