# ManimAgent: Self-Evolving Multimodal Agents for Visual Education

**Wenjia Jiang**[1], **Zongyuan Cai**[2], **Yuanhang Shao**[2], **Chenru Wang**[3], **Boyan Han**[2], **Zhixue Song**[4], **Keyu Chen**[5], **Shengwei An**[3], **Xu Yang**[2], **and Zhou Yang**[1]

[1] University of Alberta [2] Southeast University [3] Virginia Tech
[4] Emotion Machine Inc. [5] Vivavia Inc.

Multi-round reflection lets agents built on large language models recover from failures within a single task, but each task remains an isolated episode: 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 Python in the open-source Manim library to render a mathematical animation. 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 validated 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.

**Project Page:** https://manimagent.github.io/ **Code:** https://github.com/jwj1342/Paper2Manim

## 1. Introduction

*"Memory is not an instrument for exploring the past but for exploring the future."*
— Endel Tulving

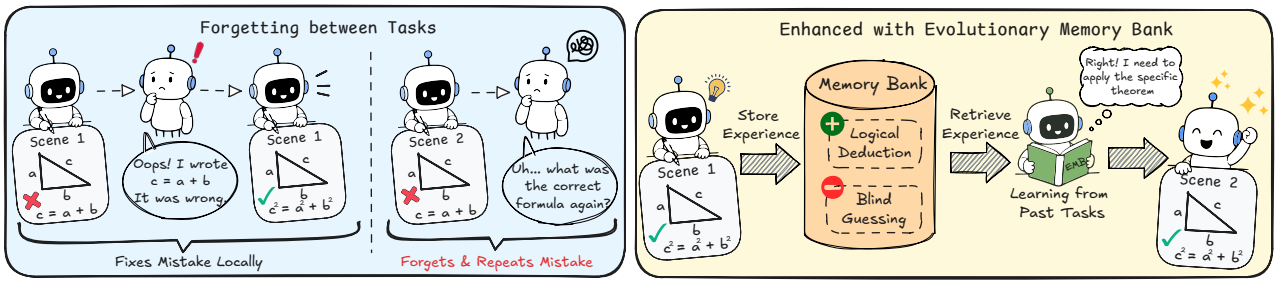


**Figure 1: ManimAgent closes the cross-task forgetting gap that reflection alone leaves open.** Each task is an isolated episode: lessons paid for in tokens are thrown away before the next task starts. **(Left)** On Task 1 the agent notices a wrong formula, revises locally, and reaches the correct scene; on Task 2 a structurally similar prompt triggers the same class of error and the full reflection cost is paid again. **(Right)** ManimAgent promotes each completed task into a dual-channel Episodic Memory Bank (EMB) that stores logical deductions that worked and blind guesses that failed. On Task 2 the bank is queried by task embedding and injected into the Coder prompt, so the agent succeeds first-try instead of re-deriving the same lesson.

Large language models and their agentic wrappers have become capable enough to handle complex, multi-stage code generation tasks (Chen et al., 2021; Shinn et al., 2023; Yao et al., 2022). 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 writes MANIM code that renders a short teaching animation, with promising results on isolated tasks (Chen et al., 2025; Jain 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 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) have closed the intra-task loop (observe error, revise, retry) but have done nothing about the inter-task loop where transferable experience lives. 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 (Lee et al., 2024; Liu et al., 2023; Zheng et al., 2023) 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 validated 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 write-time gating that makes these patterns validated 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 causal attribution (§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 (Zhang et al., 2023; Zhou et al., 2022). 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 causal-attribution 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.

**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 validated failure patterns are distilled with explicit causal-attribution 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 2) 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

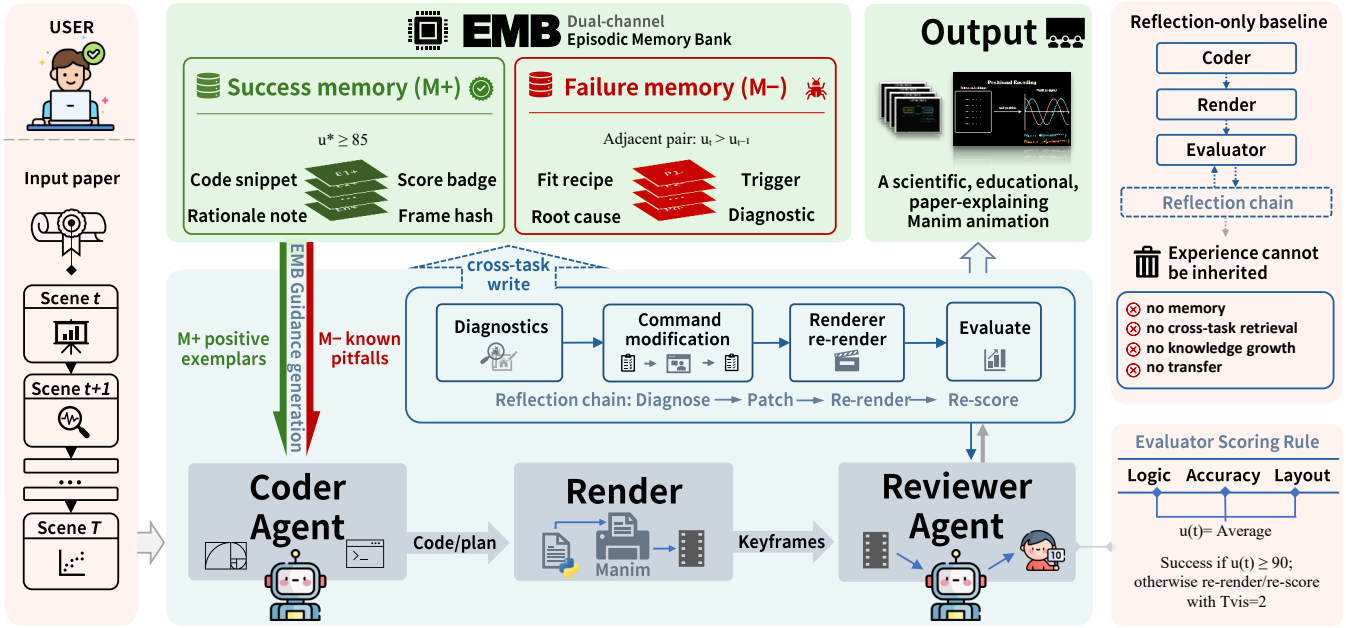


**Figure 2: ManimAgent closes a VLM-guided repair loop with dual-channel cross-task memory.** The visual reviewer uses score $u(t)$ for auto-pass and revision control, while the selected score $u^*$ and validated improvement transitions govern separate success- and failure-memory writes.

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 validated 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 STORYBOARDER $\Sigma$ decomposes $\tau$ into ordered scene specifications $\{\sigma_i\}_{i=1}^{n}$; each $\sigma_i$ 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 $(c_i^{(t)}, \text{render result}, u(t) \text{ if available}, \text{diagnostic})$ 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 block (run id, scene id, extraction source, transition ordinal, before/after scores) used for write-time deduplication; they differ in body fields by polarity (§3.3, Table 10).

### 3.2 The Self-Evolving Loop

For each scene $\sigma_i$, the loop has three nested layers: a memory retrieve call, a text-reflection loop that handles render crashes, and a visual-reflection loop that handles renderable-but-pedagogically-weak output. The two reflection loops are driven by different agents, have independent budgets, and write to the EMB through different paths. We describe each layer in turn; the full pseudocode is in Appendix A (Algorithm 1).

**Retrieve.** For scene $\sigma_i$ in task $\tau$, we form the retrieval query $e_i = \phi([s; r])$ using all-MiniLM-L6-v2 and fetch the top-$k_+ = 2$ entries $E^+$ from $\mathcal{M}^+$ and top-$k_- = 3$ entries $E^-$ from $\mathcal{M}^-$ via Faiss `IndexFlatIP` ($L_2$-normalised vectors, so inner product equals cosine). The two result sets fill disjoint prompt slots: Reference Examples for $E^+$ as soft guidance, and Known Pitfalls for $E^-$ as hard constraints. When the bank is empty, both slots carry an explicit [No entries available] marker, so prompt structure stays invariant across EMB sizes and any quality difference is attributable to retrieved content rather than to formatting changes.

**Generate and repair.** The CODER conditions on $\sigma_i$, $E^+$, and $E^-$ and emits an initial program $c_i^{(0)}$ in a single LLM call; subsequent iterations regenerate the full MANIM `Scene` subclass from scratch with new feedback appended, because patching a broken script tends to inherit the same buggy structure. A static checker rejects unsafe imports and calls before render, and the RENDERER executes each candidate inside a sandbox. On a crash, a text-only REVIEWER emits a short hint and a `retry/give_up` decision; the text-reflection loop is bounded by $T_{\text{text}}$=2 retries and short-circuits when the same error category recurs.

**Visual reflection and best-of-$N$ delivery.** Once a render succeeds, control passes to a separate visual-reflection loop driven by a VLM REVIEWER that scores keyframes on three pedagogical axes (logical flow, layout/occlusion, accuracy); a distinct VISUAL REVISER agent then rewrites the script while preserving the claim, evidence, and final takeaway of the scene. The loop terminates by VLM verdict, by an auto-pass override at $u(t) \geq \theta_{\text{conv}} = 90$, or at $T_{\text{vis}}$=2. Across the $\leq T_{\text{vis}}+1$ renderable candidates we ship the rendition $c^\star$ with the highest aggregate score $u^* = \max_t u(t)$ rather than the last candidate, guarding against polishing-pass regressions. Separating crash repair from visual revision is deliberate: the two failure modes have different prompts and budgets, and conflating them risks one budget swallowing the other. Sandbox 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 10).

**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 $c^\star$ 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 $\kappa$, $c_i^{(t)}$ 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 $v_{\text{rev}} \to v_{\text{rev}} + 1$ 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 LESSON 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 $\rho$. 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 $\ell$ 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 $v_0 \to v_1$ and $v_1 \to v_2$ 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 13); 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 updates. Because the same

| Condition / Variant | Intervention or note | Human Pass@1 ↑ | Reflection rounds ↓ | Human quality ↑ |
|---|---|---|---|---|
| **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** |
| **Ablations of Full EMB** — **(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.

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.

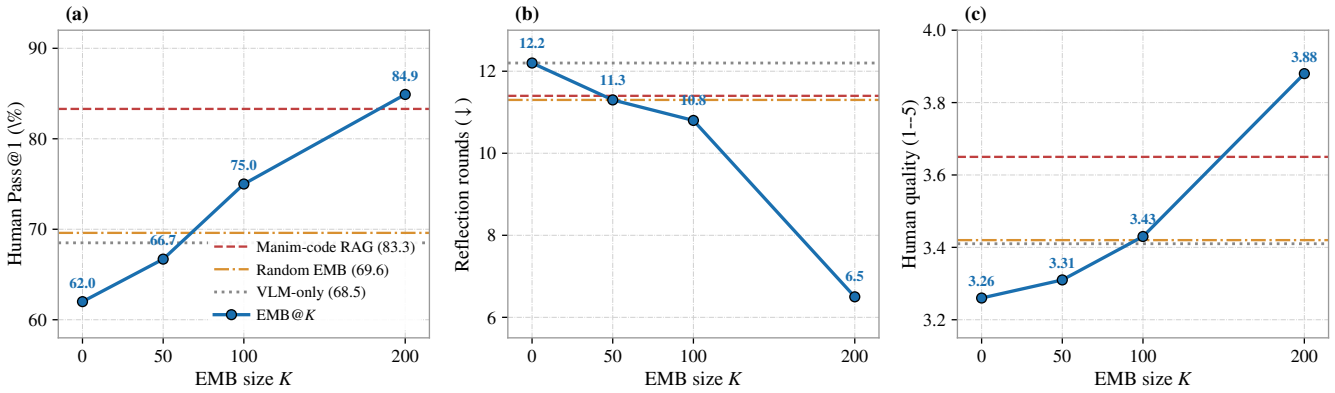


**Figure 3: EMB@$K$ scales monotonically and overtakes the strong static-RAG baseline by $K$=200.** Solid line with markers tracks the frozen EMB snapshot at $K\in\{0, 50, 100, 200\}$ on each of the three headline metrics; horizontal references are Manim-code RAG (dashed red), Random EMB (dash-dot orange), and VLM-only (dotted grey). For Human Pass@1 and Human quality, the EMB curve crosses Manim-code RAG between $K$=100 and $K$=200; for reflection rounds (lower is better) the gap to RAG widens sharply at $K$=200. Point estimates from Table 1.

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.

## 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 13.

**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., 2025), Manimator (Jain 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 per first-attempt video across 25 audited fixed-probe tasks; recruitment, stratification, questionnaire, and inter-rater 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 average number of repair rounds before convergence; 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.

EMB@$K$ moves 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 modestly improves both human metrics and substantially reduces reflection rounds (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.

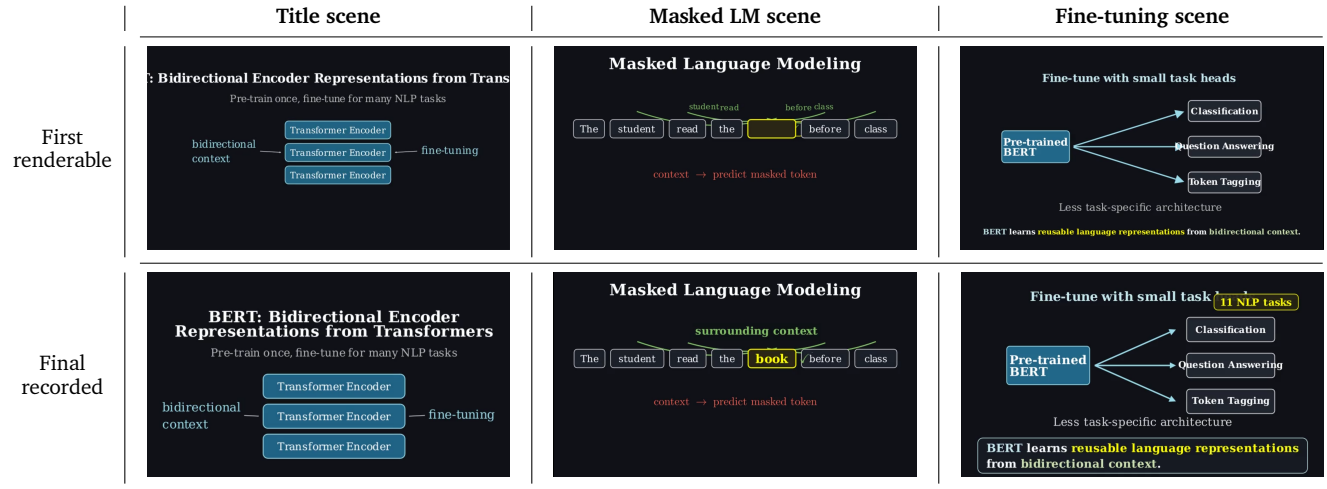


**Figure 4: A compact qualitative example traces three scene revisions for one BERT fixed-probe task.** The main paper keeps the qualitative experiment compact by showing three representative scene revisions as before/after thumbnails. The title scene becomes balanced around the encoder stack, the masked-language-modeling 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.

### 4.5 Reward Reliability and Cost

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 $\rho = -0.20$, Cohen's $\kappa = -0.07$; 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.

## 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 tested only with Manim as the rendering substrate and on five English-language scientific domains; transfer to other programmatic-animation renderers (e.g., D3.js, Three.js), to non-English literature, and to downstream learning-outcome studies with students remains future empirical work.

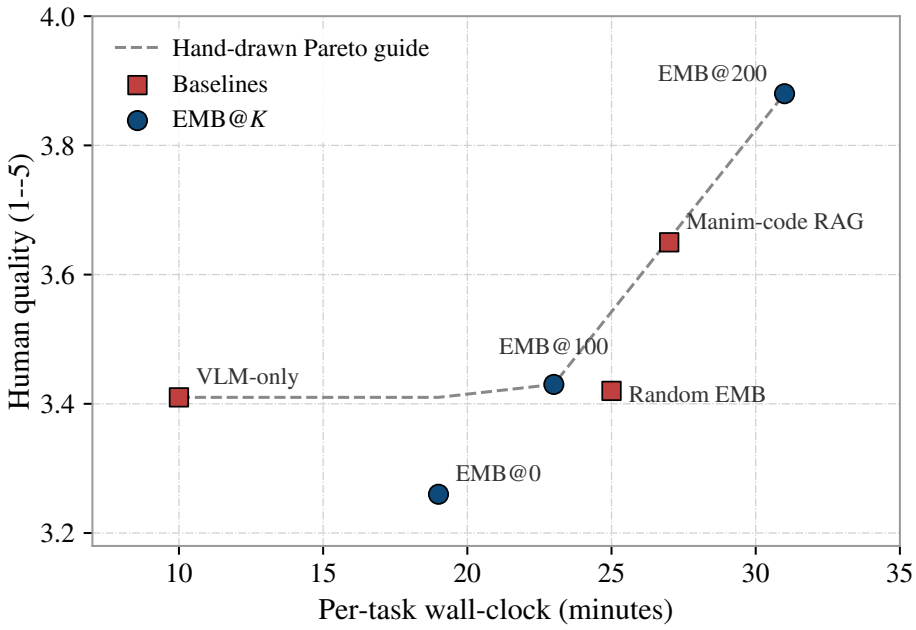


**Figure 5: EMB@200 sits on the cost–quality Pareto frontier.** Each marker is a condition; $x$ is per-task wall-clock from Table 12, $y$ is Human quality from Table 1. EMB@200 trades 4 additional minutes over Manim-code RAG for a $+0.23$ gain on Human quality while EMB@0/RAG/Random cluster around quality $\approx 3.4$. Point estimates from Table 12 and Table 1.

**Scale, cost, and memory rot.** We validate the EMB at $\mathcal{O}(10^2)$ entries; production-scale retrieval ($\mathcal{O}(10^6)$) is likely to require deduplication or hierarchical indexing, EMB write gates defined on the scores of the current VLM do not protect against memory rot after an LLM or VLM upgrade, and per-task cost (Appendix P) scales with the reflection budgets.

**Reward and prompt sensitivity.** VLM–human agreement on the paired probe subset is weak (Appendix I), so the VLM is treated only as an internal repair / write signal; the weak-VLM row of Table 1 bounds but does not eliminate shared reviewer blind spots, and retrieval-query wording and prompt-slot formatting are fixed engineering choices we do not ablate.

**Intra-stream contamination.** Paper-level split separation (§4.1) prevents leakage between memory-building and probe sets, but does not address near-duplicate accumulation within the memory-building stream; paraphrase-level deduplication is left to future work.

## References

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Yanzhe Chen, Kevin Qinghong Lin, and Mike Zheng Shou. 2025. Code2video: A code-centric paradigm for educational video generation. *arXiv preprint arXiv:2510.01174*.

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. 2025. The faiss library. *IEEE Transactions on Big Data*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, and 1 others. 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.

Vyoman Jain, Shiva Golugula, Motamarri Sai Sathvik, and 1 others. 2025. Manimator: Transforming research papers into visual explanations. *arXiv preprint arXiv:2507.14306*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. 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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

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.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, 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*, pages 1–22.

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.

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. *Advances in neural information processing systems*, 36:8634–8652.

Patrick E Shrout and Joseph L Fleiss. 1979. Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 86(2):420.

Ravidu Suien 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*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

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.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Shuyan Zhou, Uri Alon, Frank F Xu, Zhengbao Jiang, and Graham Neubig. 2022. 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. *arXiv preprint arXiv:2510.05096*.

## 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 $r \in \{\textsc{background}, \textsc{method}, \textsc{experiment}, \textsc{conclusion}\}$, 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.

**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 $\sigma_i$.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 $\leq$ 60-word hints; the LessonDistiller is capped at $\leq$ 1500 output tokens.

**$\mathcal{M}^-$ field length caps.** Trigger $\leq$ 400 chars, root cause $\leq$ 400, fix recipe $\leq$ 400, code fragments $\leq$ 800 chars each, free-form diagnostic $\leq$ 1000 chars. The RationaleWriter output stored in $\mathcal{M}^+$ is capped at $\leq$ 400 chars. Retrieval-time truncations match Table 13.

**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 $c^\star = \arg\max_t u(t)$ across renderable candidates; ties are broken by earliest $t$. These rules are enacted by Algorithm 1 in Appendix A.

## C. Extended Related Work

The two paragraphs below extend the related-work discussion in §2 to programmatic animation pipelines and reflection-based agents, deferred here for space.

| System | Reflection | VLM judge | Cross-task memory | Fixed-probe test |
|---|---|---|---|---|
| Manimator (Jain et al., 2025) | – | – | – | – |
| Code2Video (Chen et al., 2025) | multi-agent | rule-based | – | – |
| manim-generator (Joshi et al., 2026) | compile-error | – | – | – |
| Voyager (Wang et al., 2023) | – | – | human-seeded skills | partial |
| ManimAgent | multi-agent, visual | structured, multi-axis | **dual-channel, self-grown** | **primary metric** |

**Table 2: ManimAgent combines self-grown dual-channel memory with fixed-probe evaluation.** Prior systems either lack cross-task memory, rely on human-seeded skills, or do not evaluate with held-out snapshots.

**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., 2025); Manimator converts research papers into visual explanations (Jain 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.

**Reflection-based agents and intra-task loops.** Self-Refine (Madaan et al., 2023), Reflexion (Shinn et al., 2023), and CRITIC (Gou et al., 2024) let an agent observe an error signal, such as a failing test, a runtime exception, a note from a critic, or a tool-augmented external check, and revise within the same task episode. ReAct-style prompting further connects reasoning traces with tool use and environmental feedback (Yao et al., 2022). Our text- and visual-reflection loops (§3.2) follow this intra-task pattern, but reflection alone discards the reason why a revision worked once the episode ends. The central difference in our setting is that successful rationales and validated failure transitions are consolidated into a memory bank and retrieved on later scenes.

## D. Comparison with Prior Systems

Table 2 contrasts ManimAgent with prior animation-from-paper pipelines (Manimator, 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 but in a single channel that requires human-seeded primitives. 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 3 contrasts the two on five axes: corpus, growth, write rule, content, and prompt role. The corpus of

| 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 validated failure patterns |
| Prompt role | one context slot | Reference Examples and Known Pitfalls |

**Table 3: EMB differs from static RAG in source, growth, write rule, content, and prompt role.**

the EMB is self-generated task traces; it grows after each converged task under VLM-gated write rules; it stores both success rationales and validated 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.

## 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.

## G. Released Dataset Summary

Table 4 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.

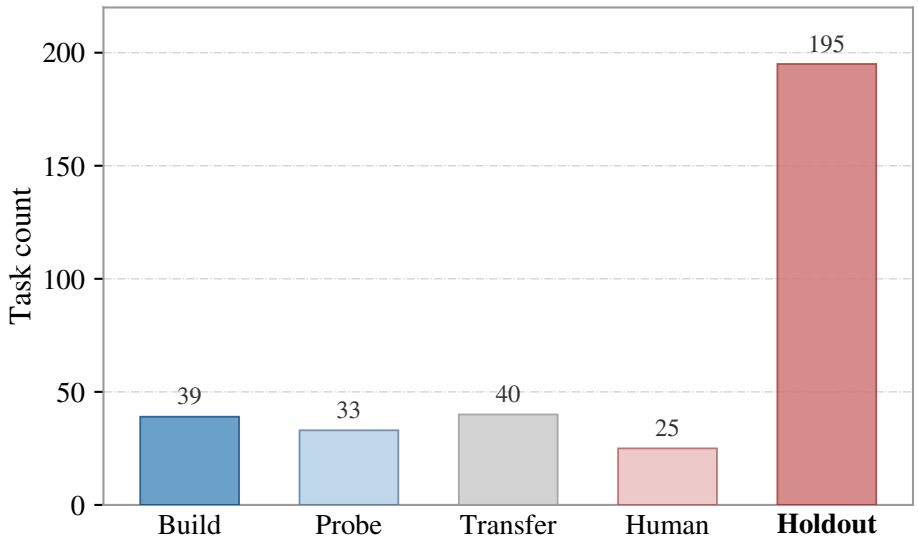


**Figure 6: Dataset split sizes are deliberately small for the three reported subsets.** The memory-building, fixed-probe, cross-test, and human-scoring subsets support construction and evaluation; the quarantined holdout is excluded from headline claims.

### G.1 Representative Fixed-Probe Examples

Table 5 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 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 5 lists representative fixed-probe tasks for coverage; Table 6 shows a schema-complete example record.

### G.2 Example Dataset Record

Table 6 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.

**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

| 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 holdout | 195 | debug metadata only (177 fallback, 18 promoted) | Excluded from all headline claims |

**Table 4: 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.

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.

**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 visualisations (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.

| 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 5: 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.

**Questionnaire and rating dimensions.** For each video, the form first asks the binary Human Pass@1 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 7 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 the proportion of individual binary judgements marked as usable within each condition, using all three blind-rater votes for every evaluated video. This vote-level metric retains every blind assessment rather than collapsing multiple assessments into one video-level vote.

| Visibility group | Field | Illustrative shortened value |
|---|---|---|
| Model-visible input | Paper title | *Attention Is All You Need* |
| | 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 |
| | Source type | Section |
| Evaluation-only fields | 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 |
| | Fatal issue flags | Closed-set annotation options used by blind raters |
| Metadata / release | 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 |
| | Quarantine metadata | Hydration status valid; included in headline probe |
| Human outputs | Ratings sidecar | Human Pass@1 votes, Likert ratings, flags, and optional notes are stored separately from model-visible records. |

**Table 6: 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.

**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.

**Inter-rater agreement.** Fleiss' $\kappa$ for binary usability votes is $-0.01$ on three-rater annotations ($n$=153 videos), indicating agreement at chance level. Intraclass correlation (ICC) for the five Likert dimensions is low across all dimensions: visual_design $= 0.24$, key_claim_coverage $= 0.00$, visual_robustness $= 0.10$, animation_flow $= 0.33$, and first_attempt_usability $= -0.09$ ($n$=23 videos with 3 raters). Binary full agreement is 59.4%. All dimensions fall below common reliability thresholds ($\kappa \geq 0.4$, ICC$\geq 0.5$), reflecting the subjective nature of animation quality assessment and the disciplinary diversity of the rater pool. Disagreements are not adjudicated; binary votes and Likert ratings are aggregated directly.

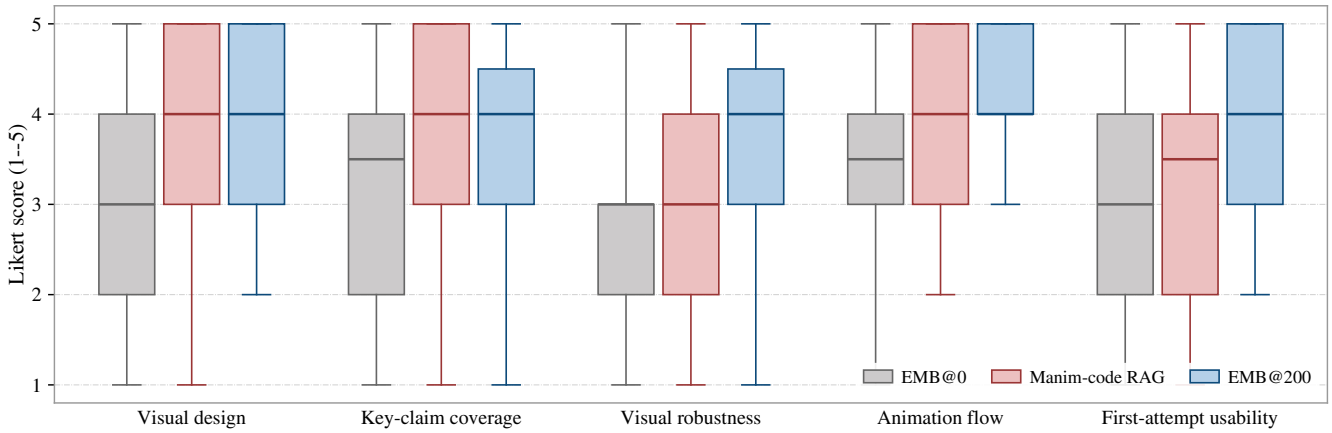


**Figure 7: Per-rater Likert scores spread across the five human-evaluation dimensions for three representative conditions.** Box distributions are derived from per-condition per-dimension rater scores (EMB@0 $n$=50, Manim-code RAG $n$=54, EMB@200 $n$=53). EMB@200 dominates EMB@0 on every dimension and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

## 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 8 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 a causal-attribution check. (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 9). 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.

| 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 7: 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.

| **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 8: 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.

| Agreement statistic | Value |
|---|---|
| Pearson $r$ (VLM aggregate vs. Human quality) | $-0.17$ |
| Spearman $\rho$ (VLM aggregate vs. Human quality) | $-0.20$ |
| Agreement rate (VLM pass/fail vs. Human Pass@1) | 0.49 |
| Cohen's $\kappa$ (VLM pass/fail vs. Human Pass@1) | $-0.07$ |

**Table 9: 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.

**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 $\rho$ between aggregate VLM score and Human quality, and Cohen's $\kappa$ between VLM pass/fail (thresholded at 90) and individual human binary usability votes. The weak, slightly negative agreement values reported in Table 9 support using blind human ratings for headline claims and treating VLM scores 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 Writer ($T = 0.3$) | Lesson Distiller ($T = 0$) |

**Table 10: 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.

## J. EMB Schema

Table 10 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 13, and per-channel content is drawn from the fields of Table 10. 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 13).

### 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 (a violation will fail 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 $\leq$ 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 validated failure$\rightarrow$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 10.

## 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 causal-attribution 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 variants close most of the Pass@1 gap but stay high in reflection rounds.

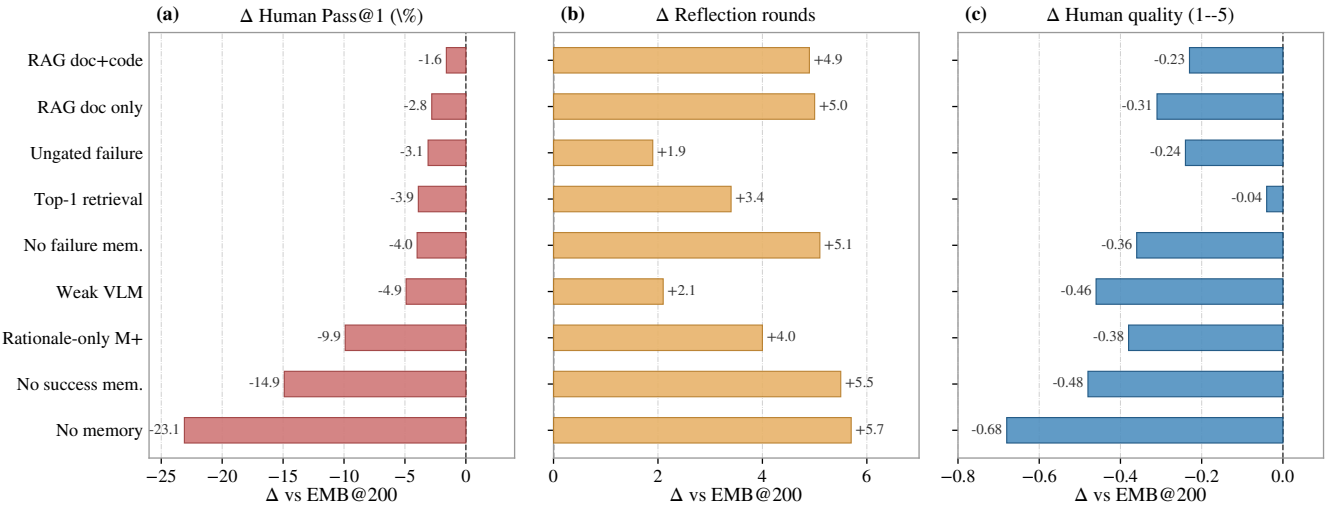


**Figure 8: Per-ablation deltas measure the gap from each variant to Full EMB.** Each row is one variant of Table 1; bars show the gap to Full EMB on each metric (negative = worse for Pass@1 / quality, positive = worse for reflection rounds, since fewer rounds is better). Removing $\mathcal{M}^+$ collapses 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.

## 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

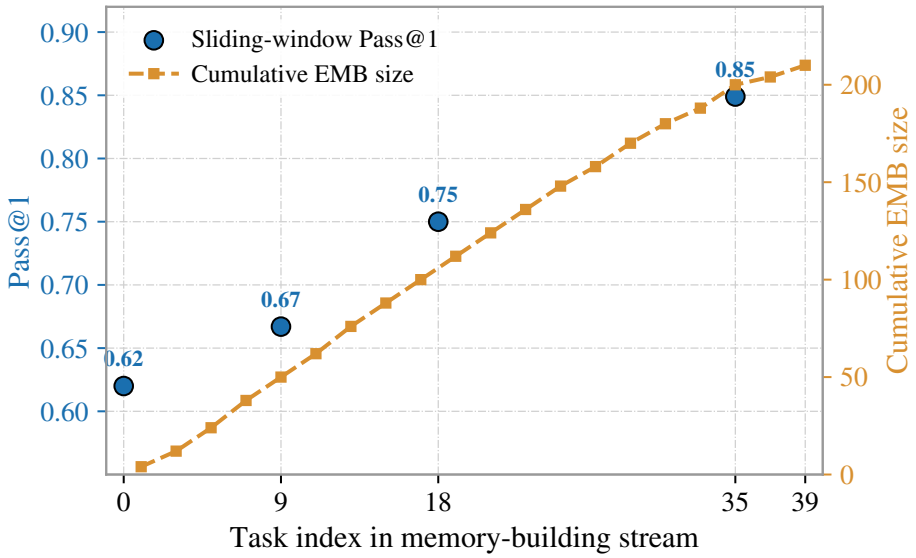


**Figure 9: ManimAgent evolves online across the memory-building stream.** Left axis: Human Pass@1 at the four frozen snapshot points (EMB@0, EMB@50, EMB@100, EMB@200). Right axis: cumulative EMB size interpolated across the 39-task memory-building stream, anchored at the four snapshot sizes and the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

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.

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 11 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

| Scene | Trace outcome | VLM axes | | Qualitative strength |
|---|---|---|---|---|
| Title | Rendered; two visual revisions; final pass | 78/58/94<br>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 context | 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<br>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 takeaway | Rendered; two visual revisions | 82/76/88<br>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 11: 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.

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 13 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 12 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 19 minutes at EMB@0 to 31 minutes at EMB@200 as retrieved scenes prompt more elaborate renders.

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.

**Pre- vs. post-cutoff stratification.** To assess whether gains are driven by memorised paper text rather than by EMB retrieval, Table 14 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.

| **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 12: 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.

| Parameter | Value |
|---|---|
| Convergence threshold (auto-pass) | 90 |
| $\mathcal{M}^+$ write threshold | 85 |
| $\mathcal{M}^-$ improvement margin | 5 |
| Text-reflection retry budget $T_{\text{text}}$ | 2 |
| Visual-reflection revision budget $T_{\text{vis}}$ | 2 |
| Positive retrieval depth $k_+$ | 2 |
| Negative retrieval depth $k_-$ | 3 |
| $\mathcal{M}^+$ code-excerpt cap | 1200 chars |

| Parameter | Value |
|---|---|
| $\mathcal{M}^+$ rationale cap | 600 chars |
| Snapshot sizes $\mathcal{K}$ | $\{0, 50, 100, 200\}$ |
| Sentence encoder $\phi$ | all-MiniLM-L6-v2 |
| Vector index | Faiss IndexFlatIP |
| Coder LLM | GPT-5.5 |
| Reviewer VLM | GPT-5.5 |
| Weak-VLM ablation | Step-3.7-Flash |
| Manim version | Community v0.20.1 |

**Table 13: All numerical hyperparameters are pinned across conditions.** Encoder $\phi$ 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.

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

**Table 14: Stratifying by LLM training cutoff shows no consistent pre-cutoff advantage.** A small gap supports EMB-mediated transfer over memorization.

**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 $(\tau_1, \tau_2, \ldots)$; agents $\Sigma$ (Storyboarder), Coder, Reviewer, VLM, VisualReviser, RationaleWriter, LessonDistiller; Renderer $R$ with static check; sentence encoder $\phi$; thresholds $\theta_{\text{conv}}$, $\theta_{\text{high}}$, $\Delta_{\text{fail}}$; budgets $T_{\text{text}}$, $T_{\text{vis}}$.

$\mathcal{M} \leftarrow \mathcal{M}^+ \cup \mathcal{M}^- \leftarrow \emptyset$
**for** each task $\tau = (s, r, d)$ in the stream **do**
  $\{\sigma_i\}_{i=1}^{n} \leftarrow \Sigma(\tau)$ ▷ decompose into scenes
  **for** each scene $\sigma_i$ **do**
    $e_i \leftarrow \phi([s; r])$; $E^+ \leftarrow$ top-$k_+(\mathcal{M}^+, e_i)$; $E^- \leftarrow$ top-$k_-(\mathcal{M}^-, e_i)$
    $c \leftarrow \text{Coder}(\sigma_i, E^+, E^-)$; textTrace $\leftarrow []$
    **for** $t_{\text{tx}} = 0, \ldots, T_{\text{text}}$ **do** ▷ text-reflection: crash → retry
      result $\leftarrow R(c)$; append $(c, \text{result})$ to textTrace ▷ static check, then sandbox render
      **if** result is success **then**
        **break**
      **end if**
      $(\text{hint}, \text{dec}) \leftarrow \text{Reviewer}(\text{result})$
      **if** dec $=$ `give_up` **or** same error category twice **then**
        **break**
      **end if**
      $c \leftarrow \text{Coder}(\sigma_i, E^+, E^-, \text{hint}, \text{result})$ ▷ regenerate from scratch
    **end for**
    **if** result is not success **then**
      **continue** ▷ scene skipped; no validated transition exists
    **end if**
    cands $\leftarrow []$; visTrace $\leftarrow []$
    **for** $t_{\text{vs}} = 0, \ldots, T_{\text{vis}}$ **do** ▷ visual-reflection: renderable → revise
      $K \leftarrow R.\text{sample\_keyframes}(c, n_K = 4)$
      $(\text{axes}, u, \text{issues}, \text{instr}, \text{dec}) \leftarrow \text{VLM}(s, \sigma_i, K)$
      append $(c, u, \text{instr})$ to visTrace; append $(c, u)$ to cands
      **if** $u \geq \theta_{\text{conv}}$ **then**
        dec $\leftarrow$ `pass` ▷ auto-pass override
      **end if**
      **if** dec $\in \{$`pass`, `fail`$\}$ **then**
        **break**
      **end if**
      $c \leftarrow \text{VisualReviser}(c, \text{instr})$ ▷ not the Coder
    **end for**
    $(c^\star, u^*) \leftarrow \arg\max_{(c,u) \in \text{cands}} u$ ▷ best-of-$N$
    **if** $u^* \geq \theta_{\text{high}}$ **then**
      $\rho \leftarrow \text{RationaleWriter}(s, \sigma_i, c^\star)$
      $\mathcal{M}^+ \leftarrow \mathcal{M}^+ \cup \{(e_i, \sigma_i, \rho, c^\star, u^*)\}$
    **end if**
    **for** each text transition $(c^{(t-1)}, c^{(t)})$ in textTrace with $c^{(t-1)}$: err, $c^{(t)}$: ok **do**
      $\ell \leftarrow \text{LessonDistiller}(c^{(t-1)}, c^{(t)}, \text{category}, \text{traceback})$
      $\mathcal{M}^- \leftarrow \mathcal{M}^- \cup \{(e_i, \sigma_i, \ell)\}$
    **end for**
    **for** each visual transition $(c^{(t-1)}, c^{(t)})$ in visTrace with $u(t) - u(t-1) \geq \Delta_{\text{fail}}$ **do**
      $\ell \leftarrow \text{LessonDistiller}(c^{(t-1)}, c^{(t)}, \text{instr}^{(t-1)})$
      $\mathcal{M}^- \leftarrow \mathcal{M}^- \cup \{(e_i, \sigma_i, \ell)\}$
    **end for**
  **end for**
**end for**
**return** $\mathcal{M}$


**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.