VIDEOWEAVER: EVALUATING AND EVOLVING SKILLS FOR AGENTIC LONG VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic long video generation requires planning, tool orchestration, and cross-clip coordination over a long horizon. Most existing video agents either rely on static, human-crafted workflows, which require substantial manual effort and poorly adapt across tasks, or iteratively refine the output of the current task without persistently distilling execution experience into reusable skills for future tasks.We introduce VideoWeaver, an agent harness and benchmark that evaluates and evolves skills for long video generation. Given a single high-level instruction, an agent dynamically composes foundation skills into its own workflow rather than following a predefined pipeline. We construct a benchmark of 16 task categories and 285 cases, with references spanning text, image, audio, video, and their combinations. We further propose an evidence-grounded agent-as-judge that inspects both the execution trace and the final video to diagnose process and output failures. Based on this feedback, our evolution algorithm progressively refines category-level composition and creator skills, allowing recurring experience to guide dynamically constructed workflows for unseen cases. Experiments show that explicit composition skills improve the generation process over foundation skills alone, while skill evolution further improves output quality and generalizes to unseen cases. Incorporating judge feedback yields additional gains, especially on output metrics, and the agent-as-judge aligns well with human, particularly on process metrics.

1 INTRODUCTION

Long video generation is advancing rapidly along two complementary lines. End-to-end models (Cui et al., 2026; Henschel et al., 2025; Yuan et al., 2026) extend a single generative process, while agentic systems (Huang et al., 2026; Liang et al., 2025; Song et al., 2026b) orchestrate generation tools across many steps. The agentic line suits the complex, multi-shot tasks we target, which require planning over a long horizon, coordinating multimodal references such as images, audio, and video, invoking different generation and editing tools, and maintaining narrative, visual, and audio consistency across many clips. Such agentic systems already address diverse scenarios, from cinematic storytelling (Mu et al., 2026; Wei et al., 2025) and personalized vlogs (Hou et al., 2025) to music videos (Tang et al., 2025) and animation (Shi et al., 2025). Representative frameworks such as ViMax (Huang et al., 2026), StoryAgent (Hu et al., 2024), UniVA (Liang et al., 2025), and Co-Director (Song et al., 2026b) demonstrate this promise. However, they rely on purpose-built orchestration mechanisms that require substantial human design and do not persistently improve from experience across cases. Meanwhile, general-purpose agent frameworks such as Claude Code (Anthropic, 2026a), Codex (OpenAI, 2026a), and OpenClaw (OpenClaw, 2026) pair a reasoning backbone with a tool-augmented runtime and show strong tool-use and orchestration. Nevertheless, their execution experience is not automatically distilled into reusable skills for future tasks. Whether such general-purpose harnesses can handle long video generation while improving their workflows from feedback has not been systematically explored.

To study this, we introduce VideoWeaver, a framework for long video generation built on a general-purpose agent harness. In our setting, a backbone model (OpenAI, 2026b; ByteDance Seed, 2026; DeepSeek-AI, 2026) is equipped with a set of *foundation skills*: self-contained, independently invocable capabilities for video, image, and audio generation, understanding, and media processing. Given a single high-level instruction, the agent composes these foundation skills into its own *composition skill*, a high-level procedural workflow, and executes it through multi-step skill invocation and clip

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Evaluation coverage and scale across video-generation, production, and general skill benchmarks; detailed statistics and counting conventions are provided in App. A.

Benchmark / Method	Domain	Scale	Process	Output
Video-generation and production benchmarks				
StoryAgent (Hu et al., 2024)	Customized storytelling	90 samples	✗	✓ (9)
GenAD-Bench (Song et al., 2026b)	Video advertising	400 scenarios	✗	✓ (4)
ThemeVlogEval (Hou et al., 2025)	Personalized Vlog generation	10 portraits	✗	✓ (12)
AutoMV (Tang et al., 2025)	Music-video generation	30 songs	✗	✓ (13)
ST-Bench (Zhang et al., 2025b)	Multi-shot storytelling	300 prompts	✗	✓ (5)
StreamingT2V (Henschel et al., 2025)	Long text-to-video generation	50 prompts	✗	✓ (3)
HeliosBench (Yuan et al., 2026)	Real-time long-video generation	240 prompts	✗	✓ (9)
AniEval (Shi et al., 2025)	Multi-shot animation	10 stories	✗	✓ (14)
UniVA-Bench (Liang et al., 2025)	Agentic video intelligence	130 cases	✓ (3)	✓ (6)
AgenticVBench (Cao et al., 2026)	Video post-production	100 tasks	✗	✓ (Var.)
General skill benchmarks and methods				
SkillsBench (Li et al., 2026a)	General	87 tasks	✗	✓ (1)
SkillOpt (Yang et al., 2026b)	ALFWorld	191 tasks	✗	✓ (1)
	LiveMathBench	177 tasks	✗	✓ (1)
	OfficeQA	246 tasks	✗	✓ (1)
SkillFlow (Zhang et al., 2026e)	General	166 tasks	✓ (5)	✓ (1)
ContinualSkillBench (Guan et al., 2026)	General	500 tasks	✗	✓ (5)
VideoWeaver (Ours)	**Agentic long-video generation**	**285 cases**	**✓ (6)**	**✓ (7)**

composition rather than following a predefined pipeline. To make this setting concrete and measurable, we construct a benchmark of 16 task categories and 285 cases, with references spanning text, image, audio, video, and their combinations (Figure 1). We use in-distribution train and test splits and hold out categories for out-of-distribution evaluation.

Evaluating such agents is itself difficult, because errors can occur at any stage of the generation process and affect the quality of the final video. A plan may be flawed, a tool call may fail, or an intermediate artifact may be corrupted in ways the final clip does not reveal. Most existing video benchmarks (Huang et al., 2023; Bansal et al., 2024; Xu et al., 2024; Han et al., 2025; Wei et al., 2026; Zhang et al., 2026d) focus on the final output (Table 1) and provide limited coverage of process-level failures. We therefore propose a skill-augmented *agent-as-judge* that inspects both the execution trace and the final video (Figure 2). It invokes tools to extract hard evidence such as video metadata and intermediate files, then returns a score and textual feedback for each of the six process and seven output metrics, with emphasis on cross-clip consistency. Using this feedback as a refinement signal, we further design a skill evolution algorithm (Figure 3) that progressively refines category-level composition and creator skills and merges the resulting creator skills into a single one.

We conduct experiments across multiple agent frameworks and foundation models. We find that performance varies notably across harness and model choices, explicit composition skills clearly improve the generation process over foundation skills alone, and skill evolution further improves output quality while generalizing to unseen cases. Incorporating judge feedback during evolution yields additional gains, especially on output metrics. In addition, a human study confirms that our agent-as-judge aligns well with human judgments. Overall, we make the following contributions.

- We introduce VideoWeaver, a framework for studying agentic long video generation, together with a benchmark spanning 16 task categories and 285 cases.
- We propose an evidence-grounded agent-as-judge that diagnoses execution traces and final outputs, providing feedback to guide skill evolution.
- We develop a evolution algorithm that distills evaluation feedback into reusable composition and creator skills, improving output quality and generalizing to unseen cases.

2 Related Work

2.1 Long Video Generation

Video generation models such as Sora (Brooks et al., 2024) and Veo (Google, 2024) produce high-fidelity short clips but still struggle with coherence and temporal drift over long durations. Recent long-video methods address this by improving autoregressive generation and memory handling: StreamingT2V (Henschel et al., 2025) uses short-term conditioning and long-term appearance

preservation for smooth chunk transitions; LoViC (Jiang et al., 2025) compresses video-text context for efficient segment-wise generation; Stable Video Infinity (Li et al., 2025) recycles autoregressive errors during training to reduce drift; Helios (Yuan et al., 2026) combines drift-aware training with context compression for real-time minute-scale generation; and LoL (Cui et al., 2026) mitigates sink-collapse with training-free RoPE jitter to enable hour-scale streaming generation.

A second line reframes long video generation as agentic orchestration, where a language model plans, decomposes, and refines across production stages, with applications to cinematic storytelling (Mu et al., 2026; Wei et al., 2025), personalized vlogs (Hou et al., 2025), music videos (Tang et al., 2025), and animation (Shi et al., 2025). ViMax (Huang et al., 2026) uses hierarchical narrative planning for long-form generation, while UniVA (Liang et al., 2025) decomposes user requests into structured video-processing steps. StoryAgent (Hu et al., 2024) and Co-Director (Song et al., 2026b) add critique, persona specialization, and feedback-based revision. FRAMEWORKERS (Li et al., 2026b) further supports dynamic sub-agent scheduling. These purpose-built systems motivate studying general-purpose agents that can build and evolve their own workflows.

2.2 Video Evaluation

Video generation is typically evaluated by reward-model-based or LLM-as-judge methods. Reward-based benchmarks such as VBench (Huang et al., 2023), VideoPhy (Bansal et al., 2024), and VisionReward (Xu et al., 2024) decompose quality into dimensions scored by specialized models, but need many task-specific evaluators, give limited textual feedback, and can be vulnerable to reward hacking (Skalse et al., 2025). LLM-as-judge benchmarks such as Video-Bench (Han et al., 2025), UniVBench (Wei et al., 2026), and CLVG-Bench (Zhang et al., 2026d) return scores and feedback across richer dimensions, while other methods dynamically formulate questions and provide interpretable feedback for multi-round visual-generation evaluation or closed-loop video prompt refinement (Zhang et al., 2025a; Song et al., 2026c). However, these methods still primarily judge the final output, offering limited diagnosis of where a long generation workflow fails. We instead use a skill-augmented agent-as-judge that inspects both outputs and execution traces and invokes tools to extract hard evidence such as video metadata and intermediate files.

2.3 Self-Evolving Agent Skills

A growing body of work studies whether LLM agents can acquire and evolve reusable skills rather than rely on static prompts. Voyager (Wang et al., 2023) first showed that an agent can accumulate reusable code skills through interaction and self-reflection. Later work refines skills from execution feedback (Alzubi et al., 2026; Zhang et al., 2026a; Ni et al., 2026), treats markdown or behavioral skills as persistent, self-improving memory (Zhou et al., 2026; Xia et al., 2026; Jiang et al., 2026; Yang et al., 2026a; Zhang et al., 2026b;c), and validates self-evolving skills in industrial and medical settings (Liu et al., 2026; Fan et al., 2026; Song et al., 2026a). We extend this line to agentic long video generation, where feedback from execution traces and final videos is used to refine reusable composition and creator skills across cases.

3 Task Definition

3.1 Dataset Construction

In our setting, **short-video generation** refers to tasks that can be completed with a single video-generation API call, with the user prompt and reference media supplied directly to the model. VideoWeaver instead focuses on **long-video generation**, which requires concatenating the outputs of multiple video-generation calls. Starting from a one-time user instruction, the agent plans and executes multi-step skill invocations, manages intermediate outputs, and progressively composes them into a coherent final video.

We construct a dataset comprising **16 categories** of long video generation tasks and **285 cases** in total, which is of the same order of magnitude as existing benchmarks shown in Table 1. Each case includes basic textual requirements such as duration and resolution, multimodal reference materials including images, audio, videos, and their combinations, instructions on how to use these materials, and a coarse story guideline. Figure 1 summarizes the dataset distribution, including the modality composition of

task inputs, and the distribution of the required video output duration. We further divide the dataset into training, test, and out-of-distribution (OOD) splits. Specifically, three task categories are reserved exclusively for OOD evaluation, while instances from the remaining categories are evenly partitioned into training and test sets. This setup enables the assessment of both in-distribution performance and generalization to unseen task categories. Details of task collection and filtering are provided in App. F.1. Representative task cases are shown in App. F.2.

3.2 Skills Workspace

In our harness environment, the backbone language model (e.g., GPT-5.5 (OpenAI, 2026b), Seed2.0 (ByteDance Seed, 2026)) has only text generation and reasoning capabilities, but does not natively support video generation or multimodal media processing. To extend the model's ability to operate in long video generation settings, we provide a set of well-formalized foundation skills: self-contained skills that expose basic, independently invocable capabilities for video, image, and audio generation, understanding, and media processing, all of which are publicly available. Examples include generating video clips and synthesizing images. The full set of foundation skills is provided in App. G.

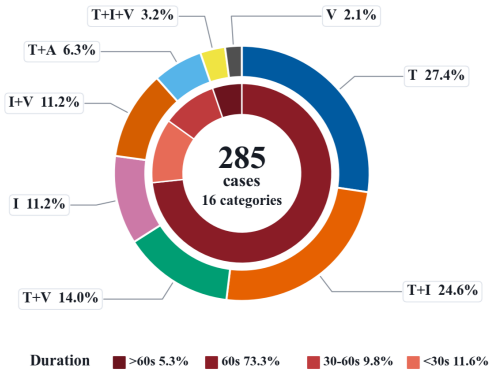

Figure 1: Dataset distribution of input modalities and required video durations. Modality abbreviations: T: Text, I: Image, V: Video, A: Audio.

However, coherent long video generation requires capabilities beyond the isolated execution of foundation skills. The agent must perform long-horizon planning: decomposing a user request into intermediate objectives, selecting and sequencing appropriate foundation skills, managing intermediate artifacts across steps, and preserving narrative, visual, and audio consistency. We define **composition skills** as high-level procedural policies that specify how foundation skills are orchestrated to accomplish such long-horizon generation tasks. We further define **skill-creator** as a skill that constructs composition skills from the available foundation skills and task cases. Together, composition and creator skills provide a persistent and editable workflow representation through which execution experience can be distilled and reused across cases.

3.3 Expert Validation

To validate the dataset and foundation skills, we manually construct expert composition skills for each task category through iterative generation, inspection, and refinement. This process verifies task feasibility, assesses the reliability and coverage of the foundation skills in realistic long video generation workflows, and provides a strong human-orchestrated reference baseline.

4 Method

4.1 Metrics

Following prior work (Lightman et al., 2023; Deshpande et al., 2025; Barke et al., 2026), we evaluate both the execution trace and the final generated video. Process metrics check execution checkpoints in the trace, such as whether the agent completes planning, input processing, and clip merging steps. Output metrics evaluate the quality and semantic consistency of the generated artifacts, including intermediate clips, images, and the final composed video. Unlike prior evaluations (Huang et al., 2023; Bansal et al., 2024) that mainly focus on a single video clip, we emphasize cross-clip consistency, assessing whether visual, audio, and narrative content remain consistent and coherent **across multiple clips**. Unless otherwise specified, each metric is scored on a binary scale of $\{0, 1\}$ with textual feedback. Details are shown in Table 2.

Table 2: Definitions and abbreviations of the process and output metrics used in our evaluation.

Metric	Abb.	Definition
Process Metrics		
Execution Error-Free Rate	EE	Successful tool calls / total tool calls.
Clip Merging	CM	Whether generated clips are merged into a final long video.
Input Processing	IP	Whether multimodal inputs are understood and preprocessed.
Planning	PL	Whether the agent produces a plan before generation.
Skill Following	SF	Proportion of prescribed composition-skill steps correctly followed.
Reference Asset Usage	RAU	Whether user-provided assets are used as required.
Output Metrics		
Format Requirements	FR	Whether duration, aspect ratio, and resolution are satisfied.
Prompt Requirements	PR	Whether the generated video satisfies instruction requirements.
Visual Consistency	VC	Whether characters and scenes remain consistent across clips.
Audio Consistency	AC	Whether speaker identity and BGM remain consistent across clips.
Audio-Video Consistency	AVC	Whether audio and video are synchronized.
Plot Logic and Coherence	PLC	Whether the story is understandable and logically connected.
Reference Fidelity	RF	Whether the final video reflects multimodal references.

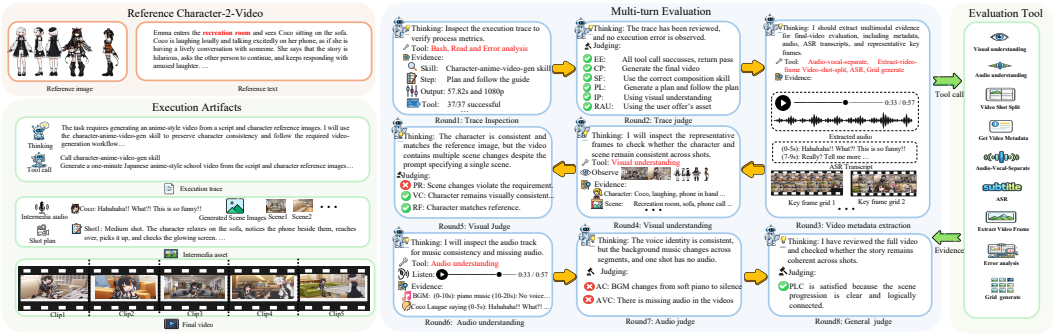

Figure 2: Evidence-grounded agent-as-judge evaluation framework. Given the user request, reference assets, execution trace, intermediate artifacts, and final video, the evaluation agent conducts an agentic evaluation. At each round, it invokes evaluation skills to inspect traces, extract video metadata, sample key frames, separate audio, run ASR, analyze errors, and understand visual or audio content. The collected evidence is then used to score process and output metrics and provide textual feedback.

Evaluation Protocol. We use Claude Code (Anthropic, 2026a) + DeepSeek V4 (DeepSeek-AI, 2026), equipped with evaluation skills as a scalable Agent-as-Judge (Zheng et al., 2023; Liu et al., 2023) for inspecting execution traces, tool-call records, generated files, logs, and final videos. The evaluation framework is shown in Figure 2.

Specifically, given an evaluation case with task inputs, execution traces, final video outputs, and metric definitions, the evaluation agent assesses each metric in an evidence-grounded manner. With evaluation skills, the agent can extend its native capabilities and dynamically invoke tools to inspect intermediate artifacts, logs, tool-call records, generated clips, and other metadata. For each metric, the agent returns a score from $\{0, 1\}$, supporting evidence, a confidence level, and textual feedback. These scores and feedback are mainly used as refinement signals for skill evolution. The rationale for using binary output-quality metrics is provided in App. B.2.

For final comparison, annotators jointly rank $n = 9$ distinct baseline outputs from Tables 3 and 4 and assign scores from $n - 1$ to 0, allowing ties; the shared OpenClaw + Seed2.0 Composition Skill baseline is counted once. The **Ranking Score** of baseline b is

$$\text{RankingScore}(b) = \frac{1}{|\mathcal{X}||\mathcal{A}|} \sum_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}} r_{x,a}(b), \tag{1}$$

where $r_{x,a}(b)$ denotes the score assigned to baseline b by annotator a for input x. Each instance is independently annotated by three annotators from diverse backgrounds, including general users, AI video researchers, and film study experts. App. B.1 details tie handling and the annotation protocol.

4.2 Evolution Algorithm

Figure 3 illustrates the overall evolution pipeline. The process consists of three stages: task inference, skill evolution, and creator skill merging.

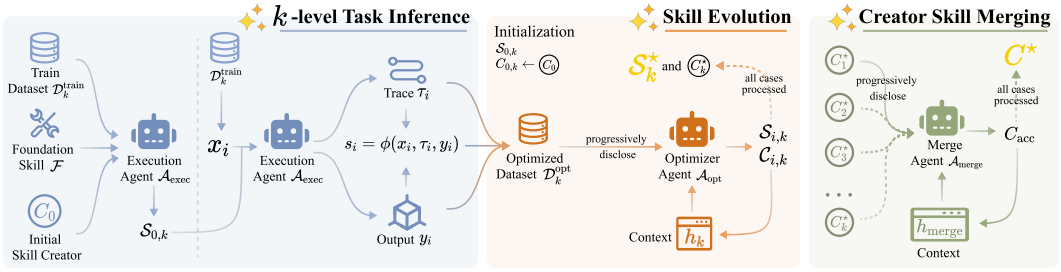

Figure 3: Overview of our evolution pipeline. The execution agent $\mathcal{A}_{\text{exec}}$ first performs task inference to generate initial category-level composition skills $\mathcal{S}_{0,k}$ and collect execution traces τ, outputs y, and evaluation feedback s_i. The optimizer agent $\mathcal{A}_{\text{opt}}$ then progressively refines composition skills and creator skills with feedback to obtain $\mathcal{S}_k^\star$ and $C_k^\star$. Finally, the merge agent $\mathcal{A}_{\text{merge}}$ integrates multiple creator skills into a single $C^\star$.

Preliminaries and Objective. Let $\mathcal{D}^{\text{train}} = \{\mathcal{D}_k^{\text{train}}\}_{k=1}^K$ and $\mathcal{D}^{\text{test}} = \{\mathcal{D}_k^{\text{test}}\}_{k=1}^K$ denote the train and test splits over K task categories, where $\mathcal{D}_k^{\text{train}}$ and $\mathcal{D}_k^{\text{test}}$ contain cases from category k. Let $\mathcal{F}$ be the fixed foundation skill set, C denote a creator skill, C_0 the initial creator skill, and $\mathcal{S}_k$ a category-level composition skill. We use three agents: the execution agent $\mathcal{A}_{\text{exec}}$, the optimizer agent $\mathcal{A}_{\text{opt}}$, and the merge agent $\mathcal{A}_{\text{merge}}$. Given an input x, an execution trace τ, and an output y, the score function $\phi(x, \tau, y)$ is instantiated by the evaluation agent and measures execution and output quality.

We optimize K category-level composition skills using only training cases:

$$\mathcal{S}_k^\star = \arg\max_{\mathcal{S}_k} \sum_{x \in \mathcal{D}_k^{\text{train}}} \phi\big(x, \mathcal{A}_{\text{exec}}(x, \mathcal{S}_k)\big), \tag{2}$$

where $\mathcal{A}_{\text{exec}}(x, \mathcal{S}_k) = (\tau, y)$ and $\mathcal{S}^\star = \{\mathcal{S}_k^\star\}_{k=1}^K$. We freeze $\mathcal{S}^\star$ before held-out testing and never use test feedback for updates.

We learn category-local creator skills on training categories and merge them for OOD transfer. Each C_k produces

$$\mathcal{S}_{C_k,k} = \mathcal{A}_{\text{exec}}(C_k, \mathcal{F}, \mathcal{D}_k^{\text{train}}), \tag{3}$$

and is optimized using training-case feedback:

$$C_k^\star = \arg\max_{C_k} \sum_{x \in \mathcal{D}_k^{\text{train}}} \phi\big(x, \mathcal{A}_{\text{exec}}(x, \mathcal{S}_{C_k,k})\big). \tag{4}$$

The category-local creators are merged into $C^\star$, which is frozen before receiving each unseen category's task-level skill query; no OOD execution results or evaluation feedback are used for updates.

Task Inference. We first run a forward inference pass to collect optimization data. For each task category k, the execution agent uses C_0 and $\mathcal{F}$ to produce an initial composition skill, $\mathcal{S}_{0,k} = \mathcal{A}_{\text{exec}}(C_0, \mathcal{F}, \mathcal{D}_k^{\text{train}})$. Each train case $x_i \in \mathcal{D}_k^{\text{train}}$ is then executed with $\mathcal{S}_{0,k}$ to obtain $(\tau_i, y_i) = \mathcal{A}_{\text{exec}}(x_i, \mathcal{S}_{0,k})$, and scored as $s_i = \phi(x_i, \tau_i, y_i)$. Thus, task inference follows the forward pass:

$$\begin{aligned} C_0, \mathcal{F}, \mathcal{D}_k^{\text{train}} &\xrightarrow{\mathcal{A}_{\text{exec}}} \mathcal{S}_{0,k}, \\ x_i \in \mathcal{D}_k^{\text{train}},\ \mathcal{S}_{0,k} &\xrightarrow{\mathcal{A}_{\text{exec}}} (x_i, \tau_i, y_i) \xrightarrow{\phi} s_i. \end{aligned} \tag{5}$$

We store (x_i, τ_i, y_i, s_i) in $\mathcal{D}_k^{\text{opt}}$, yielding $\mathcal{D}^{\text{opt}} = \{\mathcal{D}_k^{\text{opt}}\}_{k=1}^K$ and $\mathcal{S}_0 = \{\mathcal{S}_{0,k}\}_{k=1}^K$.

Skill Evolution. Using $\mathcal{D}^{\text{opt}}$, $\mathcal{A}_{\text{opt}}$ optimizes each task category in an independent session. For category k, we initialize the composition skill as $\mathcal{S}_{0,k}$, initialize a category-local creator skill $C_{0,k} \leftarrow$

C_0, and maintain a shared context h_k. Cases in $\mathcal{D}_k^{\text{opt}}$ are ***progressively disclosed*** to $\mathcal{A}_{\text{opt}}$ within the same context. At step i, the optimizer receives $z_i = (h_k, \{x_i, \tau_i, y_i, s_i, \mathcal{S}_{i-1,k}, C_{i-1,k}\})$ and updates the current skills to $(\mathcal{S}_{i,k}, C_{i,k})$. $\mathcal{A}_{\text{opt}}$ ***backpropagates*** feedback along the forward chain in Eq. (5): it analyzes the score, trace, and output to localize failures, first revises the composition skill, and then optimizes the creator skill. The persistent context h_k lets the optimizer accumulate experience across cases and summarize shared patterns within the category. After all cases are processed, we archive the final composition skill set as $\mathcal{S}^\star = \{\mathcal{S}_k^\star\}_{k=1}^K$ and obtain K category-level creator skills $\{C_k^\star\}_{k=1}^K$.

Creator Skill Merging. To integrate experience from multiple category-level creator skills, we merge them sequentially in a shared context h_{merge}. Starting from an accumulator $C_{\text{acc}} \leftarrow C_1^\star$, each remaining creator skill $C_k^\star$ for $k = 2, \ldots, K$ is ***progressively disclosed*** to $\mathcal{A}_{\text{merge}}$ through $z_k = (h_{\text{merge}}, \{C_{\text{acc}}, C_k^\star\})$, and the accumulator is updated as $C_{\text{acc}} \leftarrow \mathcal{A}_{\text{merge}}(z_k)$. The final accumulator is returned as $C^\star$, a single creator skill that absorbs experience across categories.

5 EXPERIMENTS

5.1 BASELINES

For the execution agent, we evaluate multiple harness environments, including Claude Code (Anthropic, 2026a), Codex (OpenAI, 2026a), and the open-source OpenClaw framework (OpenClaw, 2026), paired with multiple foundation models, including GPT-5.5 (OpenAI, 2026b), DeepSeek V4 (DeepSeek-AI, 2026), Opus 4.7 (Anthropic, 2026b), and Seed2.0 (ByteDance Seed, 2026). Unless otherwise specified, skill evolution uses OpenClaw with Seed2.0 as the execution agent, Codex with GPT-5.5 as the optimizer agent, and Claude Code with DeepSeek V4 as the evaluation agent to control cost and improve reproducibility. Using three distinct harness-model pairs further helps mitigate potential self-preference bias (Panickssery et al., 2024). We compare the following baselines:

- **No-Composition-Skill:** the agent is provided only with foundation skills and must orchestrate the workflow during execution.
- **Composition-Skill:** the agent first creates a task-specific composition skill that specifies the orchestration procedure over foundation skills, and then executes the task by following it.
- **Self-Evolution:** The agent updates its skills from its own execution outcomes without agent-as-judge feedback.
- **Evolution with Feedback:** Skills are updated using execution results and evidence-grounded feedback produced by the agent-as-judge.
- **Expert Skill:** A composition skill manually constructed by experts.

This comparison allows us to assess the effect of externalizing orchestration into composition skills, optimizing them through evolution, and incorporating judge feedback during the evolution process.

Table 3: Composition-skill results across harness–model pairs on the test set. **Bold**/underlined values are best/second-best before rounding.

Baseline	Process Metrics							Output Metrics								Ranking Score
	EE	CM	IP	PL	SF	RAU	Avg	FR	PR	VC	RF	AC	AVC	PLC	Avg	
OpenClaw + Seed2.0	**0.975**	**0.986**	0.886	0.869	0.921	0.895	0.922	0.728	0.255	0.493	0.582	0.567	0.605	0.631	0.552	2.96
Claude Code + Seed2.0	0.858	0.971	0.724	0.478	0.625	0.493	0.692	0.715	0.439	**0.550**	0.568	0.507	0.669	0.507	0.565	3.41
Claude Code + DeepSeek V4	0.944	0.840	0.816	0.785	0.857	0.826	0.845	0.836	0.301	0.446	0.561	0.523	0.754	0.720	0.591	4.45
Claude Code + Opus 4.7	0.975	0.986	**1.000**	0.895	**0.966**	0.961	**0.964**	0.912	0.406	0.381	0.553	0.485	0.784	0.763	0.612	4.64
Codex + GPT-5.5	**0.975**	0.985	0.987	**0.914**	0.911	**0.970**	0.957	**0.984**	**0.458**	0.521	**0.605**	**0.626**	**0.840**	**0.803**	**0.691**	**4.90**

Table 4: Skill-evolution results with OpenClaw + Seed2.0 on the test set. **Bold**/underlined values are best/second-best before rounding.

Baseline	Process Metrics							Output Metrics								Ranking Score
	EE	CM	IP	PL	SF	RAU	Avg	FR	PR	VC	RF	AC	AVC	PLC	Avg	
No Composition Skill	0.975	**0.992**	0.571	0.465	–	0.820	0.765	0.789	0.254	0.395	0.565	0.482	0.772	0.562	0.546	2.82
Composition Skill	0.975	0.986	0.886	0.869	0.921	0.895	0.922	0.728	0.255	0.493	**0.582**	**0.567**	0.605	0.631	0.552	2.96
Self-Evolution	**0.986**	0.986	**0.996**	**0.989**	**0.988**	0.982	**0.988**	0.905	0.301	0.444	0.581	0.398	0.692	0.649	0.567	4.59
Evolution with Feedback	0.986	**0.992**	0.971	0.985	0.985	**0.991**	0.985	0.913	**0.342**	0.473	0.545	0.448	0.701	0.655	0.582	5.23
Expert Composition Skill	0.965	0.971	0.948	0.985	0.964	0.986	0.970	**0.928**	0.319	**0.498**	0.528	0.524	**0.799**	**0.744**	**0.620**	**5.63**

5.2 MAIN RESULTS

We note that process scores are generally high. This is because they measure whether key execution checkpoints are completed, rather than how well each step is performed. For example, a method receives credit for performing planning if the corresponding action is observed in the trace, while the quality of the resulting plan is primarily reflected in the output metrics.

Harness and Model Comparison: Table 3 shows that performance varies substantially across different harness-model pairs. Industrial harness settings, such as Codex + GPT-5.5 and Claude Code + Opus 4.7, achieve the strongest overall performance on both process and output metrics.

The results also show clear model-harness compatibility effects. Within Claude Code, replacing Opus 4.7 with Seed2.0 leads to notable drops in both process and output performance, suggesting that different models adapt differently to the same harness interface and orchestration patterns.

Skill Evolution Results: Table 4 evaluates the optimized composition skills and other variants on test cases under the same OpenClaw + Seed2.0 execution setting. Expert Composition Skill achieves the best output average, showing that human-designed orchestration still provides the strongest reference point. Among automatic methods, Evolution with Feedback achieves a higher output average than Self-Evolution, although its process average is slightly lower. This suggests that self-evolution captures many trace-visible procedural fixes, while agent-as-judge feedback supplies output-level signals that are difficult for the agent to infer from its own execution alone. We further evaluate robustness to different case-disclosure and skill merging orders in App. E; the average output improvement is also observed over three random orders.

Table 5: Pairwise generalization results (mean over three independent runs). Test rows compare optimized with initial composition skills; OOD rows compare videos produced using new composition skills generated by optimized versus initial creator skills. Averages are sample-level.

Category	Win	Tie	Loss
Test Categories			
AI Avatar Video	30.0%	30.0%	**40.0%**
One-Sentence Anime Video	**22.2%**	66.7%	11.1%
Product Unboxing Video	12.5%	68.8%	**18.8%**
SSL Class	**40.0%**	30.0%	30.0%
Anime Video	**40.0%**	40.0%	20.0%
Character Anime Video	**16.7%**	83.3%	0.0%
Object Evolution Video	**28.6%**	71.4%	0.0%
Product Video Demo	**40.0%**	60.0%	0.0%
Cinematic Video	**50.0%**	30.0%	20.0%
Ancient Poetry Teaching	**30.0%**	70.0%	0.0%
Audio-Driven Story	11.1%	66.7%	**22.2%**
E-Commerce Subject Replace	8.3%	66.7%	**25.0%**
Long Video Edit	12.5%	62.5%	**25.0%**
Test Avg.	**25.3%**	58.2%	16.5%
OOD Categories			
AI Avatar Tech Content Video	**37.5%**	50.0%	12.5%
Theme Video	**21.1%**	68.4%	10.5%
Brand Promotion Video	20.0%	53.3%	**26.7%**
OOD Avg.	**23.8%**	59.7%	16.5%
All	**24.9%**	58.6%	16.5%

The comparison between No Composition Skill and Composition Skill indicates that foundation skills alone are insufficient for long video generation. No Composition Skill baseline achieves a high Clip Merging score, suggesting that the agent can satisfy the basic target video duration. However, its substantially lower Input Processing and Planning scores indicate that it often omits preparatory steps essential for producing high-quality, cross-clip coherent videos, such as interpreting references, structuring the narrative, and coordinating dependencies across clips. The improvements from Composition Skill therefore demonstrate the importance of explicit procedural guidance for orchestration beyond basic tool use.

5.3 GENERALIZATION

The OOD results in Table 5 evaluate the creator skills by using them to generate new composition skills, which are then executed on OOD cases.

We report means over three independent runs and omit standard deviations due to space constraints.

Table 5 shows that the optimized skills (with agent feedback) generalize beyond the training cases. On test categories, optimized composition skills achieve more wins than losses over their initial versions (25.3% vs. 16.5%), suggesting reusable orchestration improvements on held-out cases. On OOD categories, videos produced using skills generated by the optimized creator achieve more wins than losses against those produced using skills generated by the initial creator (23.8% vs. 16.5%),

suggesting transfer of learned orchestration experience to unseen categories. The relatively high tie rates may be attributed to the fact that, for many cases, the initial skills already yield competitive outputs, while pairwise comparison captures only perceptible differences in the generated videos.

5.4 Human Alignment and Feedback Utility

To validate the reliability of Agent-as-Judge, we compare the scores against human annotations on 200 sampled cases. Details of the metric annotation protocol and inter-annotator agreement are provided in App. B.3. As shown in Table 6, we report exact match and Kendall's τ_b. The alignment scores are higher for process metrics and FR mainly because they are easier to evaluate than open-ended output quality: they can be judged consistently from traces, logs, generated files, and video metadata. Thus, we use human rankings for final comparison and use the agent-as-judge output primarily as optimization signal. As shown in Table 4, this signal is effective: Evolution with Feedback outperforms Self-Evolution in both Output Avg. (0.582 vs. 0.567) and human Ranking Score (5.23 vs. 4.59).

Table 6: Alignment between Agent-as-Judge scores and human annotations on 200 sampled cases.

Type	Metric	Exact	τ_b
Process	CM	0.995	0.892
	IP	0.917	0.763
	PL	0.921	0.792
	SF	0.949	0.773
	RAU	0.970	0.845
Output	FR	0.945	0.815
	PR	0.679	0.490
	VC	0.678	0.563
	RF	0.768	0.601
	AC	0.702	0.509
	AVC	0.804	0.540
	PLC	0.763	0.390

5.5 Resource Usage Statistics

Table 7 reports aggregate resource usage statistics for each baseline. To measure efficiency, we report Relative Generation Time (**RGT**), defined as the ratio of total execution time to the duration of the generated video (e.g., an RGT of 40 means that generating a one-minute video takes 40 minutes). We also report Vision Content RGT (**VC-RGT**), defined as the ratio of time spent on video/image generation calls to total execution time. In the OpenClaw + Seed2.0 setting, better performance generally comes with higher resource usage: tool calls, LLM tokens, and RGT increase from No Composition Skill to Expert Composition Skill. Meanwhile, VC-RGT decreases from 0.80 to 0.72, suggesting that stronger methods spend more time on planning, reasoning, and coordination rather than waiting for video/image generation calls. We report monetary costs and discuss cost matching in Appendix H.

Table 7: Resource usage statistics for each baseline. RGT denotes Relative Generation Time, and VC-RGT denotes Vision Content Relative Generation Time.

Baseline	RGT	VC-RGT	Tool Calls	Video Tokens	Image Tokens	LLM Tokens
Composition Skill Baselines with Different Harness+Model						
Claude Code + Seed2.0	36.28	0.59	28.19	1.52M	4.49K	1.31M
Claude Code + DeepSeek V4	63.41	0.66	61.89	1.65M	31.59K	10.09M
Claude Code + Opus 4.7	29.54	0.59	32.91	1.50M	9.29K	4.27M
Codex + GPT-5.5	54.67	0.78	86.65	1.83M	27.21K	4.32M
Baselines with OpenClaw + Seed2.0						
No Composition Skill	38.29	0.80	31.30	1.66M	8.02K	2.08M
Composition Skill	48.81	0.78	37.11	1.85M	3.84K	2.72M
Self-Evolution	61.78	0.73	46.98	1.52M	10.26K	3.74M
Evolution with Feedback	56.08	0.74	45.95	1.72M	9.17K	3.62M
Expert Composition Skill	65.94	0.72	43.07	1.53M	53.24K	3.39M

6 Conclusion

We presented VideoWeaver, an agent harness and benchmark agentic long video generation. VideoWeaver includes 16 task categories and 285 cases, an evidence-grounded agent-as-judge that evaluates execution traces and final videos, and a evolution algorithm that distills feedback into reusable composition and creator skills. Experiments show that composition skills improve the generation process, while skill evolution improves output quality and generalizes to unseen cases.

AI USE STATEMENT

We used an AI assistant, Codex, to support code development, debugging, and writing review during the preparation of this work. All AI-assisted outputs were carefully inspected and revised by the authors. The assistant was not used to make autonomous research decisions or alter experimental outcomes. We manually verified all code, data processing steps, experimental results, and paper content to avoid hallucination, bias, or unsupported claims.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the benchmark construction and filtering procedure in Sec. 3.1 and App. F.1, the metric definitions and evaluation protocol in Sec. 4.1, and the skill evolution procedure in Sec. 4.2 and Alg. 1. The harness–model selections are reported in Sec. 5.1, while the foundation-tool configurations and exact model identifiers are provided in App. G. We further report the human annotation protocol in App. B.1, robustness results across random orders in App. E, and benchmark execution costs in App. H.

REFERENCES

Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Weiyuan Chen, and Tu Vu. Evoskill: Automated skill discovery for multi-agent systems, 2026. URL https://arxiv.org/abs/2603.02766.

Anthropic. Claude code overview. https://code.claude.com/docs/en/overview, 2026a. Accessed: 2026-05-24.

Anthropic. Claude opus 4.7 system card. https://www.anthropic.com/system-cards, 2026b. Accessed: 2026-05-24.

Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation, 2024. URL https://arxiv.org/abs/2406.03520.

Shraddha Barke, Arnav Goyal, Alind Khare, Avaljot Singh, Suman Nath, and Chetan Bansal. Agentrx: Diagnosing ai agent failures from execution trajectories, 2026. URL https://arxiv.org/abs/2602.02475.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

ByteDance Seed. Seed2.0. https://github.com/ByteDance-Seed/Seed2.0, 2026. Accessed: 2026-05-24.

Zongheng Cao, Yi Zheng, Rui Song, and Xinyu Hu. Agenticvbench: Can ai agents complete real-world post-production tasks?, 2026. URL https://arxiv.org/abs/2605.27705.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Lol: Longer than longer, scaling video generation to hour, 2026. URL https://arxiv.org/abs/2601.16914.

DeepSeek-AI. Deepseek-v4. https://www.deepseek.com/en/transparency/, 2026. Accessed: 2026-05-24.

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. Trail: Trace reasoning and agentic issue localization, 2025. URL https://arxiv.org/abs/2505.08638.

Lin Fan, Pengyu Dai, Zhipeng Deng, Haolin Wang, Xun Gong, Yefeng Zheng, and Yafei Ou. Evolving medical imaging agents via experience-driven self-skill discovery, 2026. URL https://arxiv.org/abs/2603.05860.

Google. Google i/o 2024: Introducing veo and imagen 3 generative ai tools, May 2024. URL https://blog.google/technology/ai/google-generative-ai-veo-imagen-3/.

Tianyi Guan, Yiding Wang, Haotong Yang, Siyuan Cao, Shirui Liu, Yi Hu, Jiaqi Li, and Muhan Zhang. Continualskillbench: Can llm agents truly evolve their capabilities? *Computing Research Repository*, arXiv:2608.03874, 2026. URL https://arxiv.org/abs/2608.03874.

Hui Han, Siyuan Li, Jiaqi Chen, et al. Video-bench: Human-aligned video generation benchmark. *arXiv preprint arXiv:2504.04907*, 2025. URL https://arxiv.org/abs/2504.04907.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Xiaolu Hou, Bing Ma, Jiaxiang Cheng, Xuhua Ren, Kai Yu, Wenyue Li, Tianxiang Zheng, and Qinglin Lu. Personavlog: Personalized multimodal vlog generation with multi-agent collaboration and iterative self-correction, 2025. URL https://arxiv.org/abs/2508.13602.

Panwen Hu, Jin Jiang, Jianqi Chen, Mingfei Han, Shengcai Liao, Xiaojun Chang, and Xiaodan Liang. Storyagent: Customized storytelling video generation via multi-agent collaboration, 2024. URL https://arxiv.org/abs/2411.04925.

Lingxuan Huang, Sizhe He, Hengji Zhou, Liqiang Nie, Lianghao Xia, and Chao Huang. Vimax: Agentic video generation, 2026. URL https://arxiv.org/abs/2606.07649.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. *arXiv preprint arXiv:2311.17982*, 2023. URL https://arxiv.org/abs/2311.17982.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R. Fung. Xskill: Continual learning from experience and skills in multimodal agents, 2026. URL https://arxiv.org/abs/2603.12056.

Jiaxiu Jiang, Wenbo Li, Jingjing Ren, Yuping Qiu, Yong Guo, Xiaogang Xu, Han Wu, and Wangmeng Zuo. Lovic: Efficient long video generation with context compression, 2025. URL https://arxiv.org/abs/2507.12952.

Wuyang Li, Wentao Pan, Po-Chien Luan, Yang Gao, and Alexandre Alahi. Stable video infinity: Infinite-length video generation with error recycling, 2025. URL https://arxiv.org/abs/2510.09212.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, Chujun Tao, Binxu Li, Xuandong Zhao, Hejia Geng, Xiaojun Wu, Junwei Zhou, Xiaokun Chen, Hanwen Xing, Yubo Li, Qunhong Zeng, Di Wang, Yuanli Wang, Roey Ben Chaim, Penghao Jiang, Haotian Shen, Luyang Kong, Xinyi Liu, Runhui Wang, Xuanqing Liu, Jiachen Li, Xin Lan, Yueqian Lin, Wengao Ye, Junwei He, Songlin Li, Yue Zhang, Yipeng Gao, Yijiang Li, Ze Ma, Liqiang Jing, Tianyu Wang, Kaixin Li, Yiqi Xue, Haoran Lyu, Yizhuo He, Yuchen Tian, Shutong Wu, Bowei Wang, Yixuan Gao, Bo Chen, Litong Liu, Sikai Cheng, Jiajun Bao, Shuaicheng Tong, Shuwen Xu, Terry Yue Zhuo, Tinghan Ye, Qi Qi, Miao Li, Longtai Liao, Zelin Tan, Chang Shi, Xilin Tang, Srinath Tankasala, Boqin Yuan, Yaoyao Qian, Jianhong Tu, Chenguang Wang, Yizhou Sun, Wei Wang, Aaron Taylor, Ziyue Yang, Changkun Guan, Zhikang Dong, Xinyu Zhang, Steven Dillmann, Han chung Lee, and Dawn Song. Skillsbench: Benchmarking how well agent skills work across diverse tasks, 2026a. URL https://arxiv.org/abs/2602.12670.

Zhendong Li, Lei Sun, Letian Shi, Deheng Zhang, Ruibo Ming, Mengshun Hu, Dannong Xu, Jian Wang, Danda Paudel, Luc Van Gool, and Jinjin Gu. Frameworkers: A dynamic multi-agent framework for ai-generated video production, 2026b. URL https://arxiv.org/abs/2608.29814.

Zhengyang Liang, Daoan Zhang, Huichi Zhou, Rui Huang, Bobo Li, Yuechen Zhang, Shengqiong Wu, Xiaohan Wang, Jiebo Luo, Lizi Liao, and Hao Fei. Univa: Universal video agent towards open-source next-generation video generalist, 2025. URL https://arxiv.org/abs/2511.08521.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Xingyan Liu, Xiyue Luo, Linyu Li, Ganghong Huang, Jianfeng Liu, and Honglin Qiao. Skillforge: Forging domain-specific, self-evolving agent skills in cloud technical support. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 4763–4768. ACM, July 2026. doi: 10.1145/3805712.3808466. URL http://dx.doi.org/10.1145/3805712.3808466.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023. URL https://arxiv.org/abs/2303.16634.

Chenyu Mu, Xin He, Qu Yang, Wanshun Chen, Jiadi Yao, Huang Liu, Zihao Yi, Bo Zhao, Xingyu Chen, Ruotian Ma, Fanghua Ye, Erkun Yang, Cheng Deng, Zhaopeng Tu, Xiaolong Li, and Linus. The script is all you need: An agentic framework for long-horizon dialogue-to-cinematic video generation, 2026. URL https://arxiv.org/abs/2601.17737.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2skill: Distill trajectory-local lessons into transferable agent skills, 2026. URL https://arxiv.org/abs/2603.25158.

OpenAI. Introducing the codex app. https://openai.com/index/introducing-the-codex-app/, 2026a. Accessed: 2026-05-24.

OpenAI. Gpt-5.5 system card. https://openai.com/index/gpt-5-5-system-card/, 2026b. Accessed: 2026-05-24.

OpenClaw. OpenClaw. https://github.com/openclaw/openclaw, 2026. Accessed: 2026-09-24.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems*, 37:68772–68802, 2024.

H. Shi, Y. Li, X. Chen, L. Wang, B. Hu, and M. Zhang. Animaker: Multi-agent animated storytelling with mcts-driven clip generation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking, 2025. URL https://arxiv.org/abs/2209.13085.

Weijia Song, Jiashu Yue, and Zhe Pang. Abstral: Automatic design of multi-agent systems through iterative refinement and topology optimization, 2026a. URL https://arxiv.org/abs/2603.22791.

Yale Song, Yiwen Song, Nick Losier, Nathan Hodson, Ye Jin, Rhyard Zhu, Yan Xu, Daniel Vlasic, Carina Claassen, Jasmine Leon, Khanh G. LeViet, Zack Chomyn, Joe Timmons, Brett Slatkin, Scott Penberthy, and Tomas Pfister. Co-director: Agentic generative video storytelling, 2026b. URL https://arxiv.org/abs/2604.24842.

Yiwen Song, Tomas Pfister, and Yale Song. Vqqa: An agentic approach for video evaluation and quality improvement, 2026c. URL https://arxiv.org/abs/2603.12310.

X. Tang, X. Lei, C. Zhu, S. Chen, R. Yuan, Y. Li, C. Oh, G. Zhang, W. Huang, E. Benetos, et al. Automv: An automatic multi-agent system for music video generation. *arXiv preprint arXiv:2512.12196*, 2025.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, et al. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Jianhui Wei, Xiaotian Zhang, Yichen Li, et al. Univbench: Towards unified evaluation for video foundation models. *arXiv preprint arXiv:2602.21835*, 2026. URL https://arxiv.org/abs/2602.21835.

Z. Wei, M. Li, Z. Zhang, R. Yuan, P. Hui, H. Qu, J. Evans, M. Agrawala, and A. Rao. Hollywood town: Long-video generation via cross-modal multi-agent orchestration. *arXiv preprint arXiv:2510.22431*, 2025.

Peng Xia, Jianwen Chen, Xinyu Yang, Haoqin Tu, Jiaqi Liu, Kaiwen Xiong, Siwei Han, Shi Qiu, Haonian Ji, Yuyin Zhou, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. Metaclaw: Just talk – an agent that meta-learns and evolves in the wild, 2026. URL https://arxiv.org/abs/2603.17187.

Jiazheng Xu, Yu Huang, Jiale Cheng, et al. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. *arXiv preprint arXiv:2412.21059*, 2024. URL https://arxiv.org/abs/2412.21059.

Y. Yang, J. Li, Q. Pan, et al. Autoskill: Experience-driven lifelong learning via skill self-evolution. *Computing Research Repository*, arXiv:2603.01145, 2026a. URL https://arxiv.org/abs/2603.01145.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. Skillopt: Executive strategy for self-evolving agent skills, 2026b. URL https://arxiv.org/abs/2605.23904.

Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026.

Fan Zhang, Shulin Tian, Ziqi Huang, Yu Qiao, and Ziwei Liu. Evaluation agent: Efficient and promptable evaluation framework for visual generative models, 2025a. URL https://arxiv.org/abs/2412.09645.

Hanrong Zhang, Shicheng Fan, Henry Peng Zou, Yankai Chen, Zhenting Wang, Jiayu Zhou, Chengze Li, Wei-Chieh Huang, Yifei Yao, Kening Zheng, Xue, Liu, Xiaoxiao Li, and Philip S. Yu. Coevoskills: Self-evolving agent skills via co-evolutionary verification, 2026a. URL https://arxiv.org/abs/2604.01687.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. Memskill: Learning and evolving memory skills for self-evolving agents, 2026b. URL https://arxiv.org/abs/2602.02474.

Kaiwen Zhang, Liming Jiang, Angtian Wang, Jacob Zhiyuan Fang, Tiancheng Zhi, Qing Yan, Hao Kang, Xin Lu, and Xingang Pan. Storymem: Multi-shot long video storytelling with memory, 2025b. URL https://arxiv.org/abs/2512.19539.

Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, Zhuo Li, Yujie Zheng, Weinan Zhang, Ying Wen, Zhiyu Li, Feiyu Xiong, Yutao Qi, Bo Tang, and Muning Wen. Memrl: Self-evolving agents via runtime reinforcement learning on episodic memory, 2026c. URL https://arxiv.org/abs/2601.03192.

Xiaotian Zhang, Jianhui Wei, Yuan Wang, et al. How far are video models from true multimodal reasoning? *arXiv preprint arXiv:2604.19193*, 2026d. URL https://arxiv.org/abs/2604.19193.

Ziao Zhang, Kou Shi, Shiting Huang, Avery Nie, Yu Zeng, Yiming Zhao, Zhen Fang, Qishen Su, Haibo Qiu, Wei Yang, Qingnan Ren, Shun Zou, Wenxuan Huang, Lin Chen, Zehui Chen, and Feng Zhao. Skillflow: Benchmarking lifelong skill discovery and evolution for autonomous agents. *Computing Research Repository*, arXiv:2604.17308, 2026e. URL https://arxiv.org/abs/2604.17308.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023. URL https://arxiv.org/abs/2306.05685.

Huichi Zhou, Siyuan Guo, Anjie Liu, Zhongwei Yu, Ziqin Gong, Bowen Zhao, Zhixun Chen, Menglong Zhang, Yihang Chen, Jinsong Li, Runyu Yang, Qiangbin Liu, Xinlei Yu, Jianmin Zhou, Na Wang, Chunyang Sun, and Jun Wang. Memento-skills: Let agents design agents, 2026. URL https://arxiv.org/abs/2603.18743.

A BENCHMARK STATISTICS

This section summarizes the scale and evaluation metrics of the benchmarks in Table 1. We count fixed, explicitly named leaf-level metrics and exclude averages or other scores derived solely by aggregating those metrics. When a benchmark uses task-specific verifiers or rubric sets whose contents vary across tasks, we report them as task-specific rather than assigning a fixed metric count.

A.1 VIDEO-GENERATION AND PRODUCTION BENCHMARKS

StoryAgent (Hu et al., 2024). The public-data evaluation contains 90 samples: five subjects from PororoSV and four from FlintstonesSV, with ten four-shot samples for each subject. An additional open-domain set contains eight subjects. The four reference-based metrics are FVD, PSNR, SSIM, and LPIPS; the five open-domain metrics are Inception Score, CLIP-score, CLIP-temp, Warping Error, and Flow-score, giving nine distinct outcome metrics in total. StoryAgent does not separately score the generation process.

GenAD-Bench (Song et al., 2026b). GenAD-Bench contains 400 advertising scenarios for 200 fictional products, with 200 scenarios used for hill-climbing and 200 for validation. It evaluates Visual Asset Fidelity, Demographic Alignment, Marketing Appeal, and Visual Quality. These four metrics evaluate the produced advertisements rather than the intermediate process.

ThemeVlogEval (Hou et al., 2025). The reported source data contain ten celebrity portraits and two visual styles, together with LLM-generated and manually screened themes whose total number is not stated. Its twelve outcome metrics comprise four storyboard metrics (Story Interest, Temporal Continuity, Behavioral Diversity, and Thematic Consistency), two keyframe metrics (Text–Image Alignment and Character Consistency), and six video metrics (Subject Consistency, Background Consistency, Motion Smoothness, Dynamic Degree, Aesthetic Quality, and Imaging Quality). It does not define process metrics.

AutoMV (Tang et al., 2025). AutoMV evaluates 30 professionally released songs. Its outcome evaluation contains ImageBind Score and twelve rubric criteria: Character Consistency, Physical Authenticity, Lip-sync Accuracy, Visual Style Harmony, Shot Continuity, Audio–Visual Correlation, Musical Theme Relevance, Storytelling, Emotional Expression, Visual Composition and Quality, Creativity, and AI Novelty. We therefore count thirteen outcome metrics and no process metrics.

ST-Bench (Zhang et al., 2025b). ST-Bench contains 30 long story scripts, each comprising 8–12 shots, and 300 shot-level prompts in total. Its five outcome metrics are Aesthetic Quality, Global Prompt Following, Single-shot Prompt Following, Overall Cross-shot Consistency, and Top-10 Cross-shot Consistency. It does not evaluate the generation process separately.

StreamingT2V (Henschel et al., 2025). StreamingT2V uses 50 prompts and reports three quantitative outcome metrics: Motion-Aware Warp Error (MAWE), the number of detected scene cuts (SCuts), and CLIP text–video alignment. It does not report process metrics.

HeliosBench (Yuan et al., 2026). HeliosBench contains 240 prompts spanning four duration regimes of 81, 240, 720, and 1,440 frames. Its five base quality metrics are Aesthetic, Dynamic, Motion Smoothness, Semantic, and Naturalness. It additionally reports drifting for Aesthetic, Motion Smoothness, Semantic, and Naturalness, giving nine leaf-level outcome metrics. Composite totals and throughput scores are not counted, and the benchmark does not score the generation process.

AniEval (Shi et al., 2025). AniEval is evaluated on ten stories; its human study covers 90 videos from nine methods. Its four primary domains contain fourteen fine-grained outcome metrics. Overall Video Quality contains VQA-A, VQA-T, and MusIQ. Text–Video Alignment contains Text–Video Consistency, Text–Story Consistency, Detection-Score, and Count-Score. Video Consistency contains DreamSim, Face Consistency, Warping Error, and Semantic Consistency. Motion Quality contains Action Recognition, Action Strength, and Motion AC-Score. AniEval does not define process metrics.

UniVA-Bench (Liang et al., 2025). The scale of 130 cases in Table 1 is derived from the reported task subsets because the paper does not directly state an aggregate size. The paper reports ten understanding videos, ten Image/Entities2Video cases, three ten-case Video2Video settings, ten editing cases, ten segmentation cases, and a 50-instance planning set, in addition to its LongText2Video set. The three process metrics are weighted Plan Edit Distance (wPED), Dependency Coverage (DepCov), and Replanning Quality (ReplanQ). The six leaf-level outcome metrics are CLIP Score, DINO Score, MLLM Preference, Understanding Accuracy, J-mean, and F-mean. J&F-mean is excluded because it aggregates J-mean and F-mean.

AgenticVBench (Cao et al., 2026). AgenticVBench contains 100 tasks: 18 Assembly, 18 Repair, 28 Sequencing, and 36 Repurpose tasks. It evaluates submitted artifacts rather than agent trajectories. Assembly uses a chance-adjusted correct-selection score; Repair uses task-dependent programmatic repair measurements; Sequencing combines Normalized Distance, Longest Increasing Subsequence, and Adjacent Fidelity; and Repurpose uses task-specific binary rubrics covering Format, Visual, Narrative, Sound, and optional Penalty items. Because the verifier and rubric contents vary by task family and instance, we denote its outcome metrics as task-specific in Table 1.

A.2 General Skill Benchmarks and Methods

SkillsBench (Li et al., 2026a). SkillsBench contains 87 tasks across 8 domains. Its primary outcome metric is task-macro pass rate, computed from deterministic pass/fail verification. Absolute and normalized gains are derived comparisons between conditions. It does not separately score the task-execution process.

SkillOpt (Yang et al., 2026b). The released SkillOpt manifests contain 191 ALFWorld tasks (39 train, 18 validation, and 134 test), 177 LiveMathematicianBench tasks (35, 17, and 125), and 246 OfficeQA tasks (50, 24, and 172). Each benchmark uses one native outcome measure: hard success rate for ALFWorld and hard or exact-match accuracy for LiveMathematicianBench and OfficeQA. SkillOpt uses execution traces to optimize skills but does not report a separate process score.

SkillFlow (Zhang et al., 2026e). SkillFlow contains 166 tasks organized into 20 families of 8–9 tasks across five domains. Its outcome metric is task success rate. It also reports five process, efficiency, or skill-lifecycle statistics: interaction turns, monetary cost, output tokens, the number of retained skills, and skill reuse rate.

ContinualSkillBench (Guan et al., 2026). ContinualSkillBench contains 500 tasks, comprising 100 sequential subtasks in each of five domains. Its five leaf-level outcome scores are Exact Match, F1, Numeric, Rubric Judge, and Programmatic evaluation. Raw Reward and Normalized Reward aggregate these task-level scores and are therefore excluded from the count. The benchmark analyzes task reuse, core-skill coverage, and generated skill libraries, but does not directly score the execution workflow of each task; we therefore mark process evaluation as absent in Table 1.

A.3 VideoWeaver

VideoWeaver contains 285 cases across 16 task categories. Its six process metrics are Execution Error-Free Rate (EE), Clip Merging (CM), Input Processing (IP), Planning (PL), Skill Following (SF), and Reference Asset Usage (RAU). Its seven outcome metrics are Format Requirements (FR), Prompt Requirements (PR), Visual Consistency (VC), Reference Fidelity (RF), Audio Consistency (AC), Audio–Video Consistency (AVC), and Plot Logic and Coherence (PLC).

B Human Annotation Details

B.1 Ranking Annotation Interface

We provide the human annotation interface in Figure 4. For each evaluation instance, annotators are shown the user input and candidate videos generated by different baselines, with method identities hidden and candidate presentation order randomized. They rank the videos according to holistic

video quality by either dragging videos to reorder them or using the up/down controls in the interface. We use competition ranking: tied videos share the same rank, and subsequent ranks skip the occupied positions (e.g., $1, 1, 3$). A video at rank q receives $n - q$ points, where $n = 9$. Thus, two videos tied for first each receive 8 points, and the next video is ranked third and receives 6 points. Scores are averaged over cases and annotators as in Sec. 4.1.

All annotators were compensated at a rate of 10 USD per hour. The ranking annotation task involves viewing generated videos and task prompts used in our benchmark. It does not require annotators to provide personal information beyond standard annotation-platform account information, and no private or personally identifiable data is included in the released annotations. The collected human judgments are used only for evaluation and analysis in this work, and are not used to train any model.

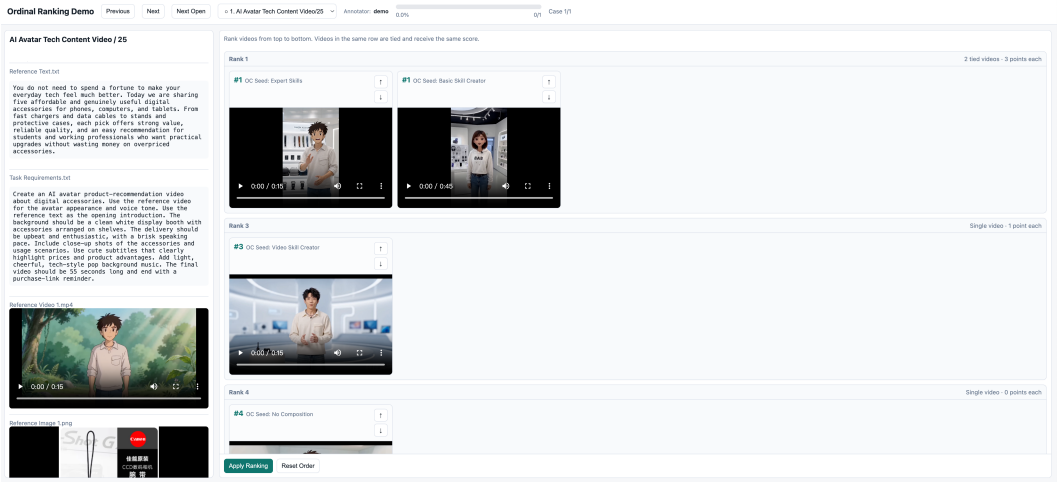

Figure 4: Illustration of the ranking annotation interface. Annotators reorder candidate videos for the same input; videos in the same row are tied. Method labels shown in this demonstration are hidden during annotation, and candidate presentation order is randomized.

B.2 Rationale for Binary Output-Quality Metrics

The binary output-quality metrics are used to provide scalable diagnostic feedback for skill evolution, rather than as the sole basis for final performance comparison. As described in Sec. 4.1, our final cross-system quality comparison is based on human rankings. We choose binary scores to reduce calibration ambiguity. Fine-grained scales such as 0–5 require defining subtle boundaries between adjacent levels, such as "slightly coherent," "moderately coherent," and "mostly coherent," which may be interpreted inconsistently across evaluators. Evaluators can more reliably determine whether an obvious audio-video mismatch, reference mismatch, or plot inconsistency is present than calibrate its precise degree. We therefore use binary output metrics for diagnostic feedback and human rankings for final performance comparison.

B.3 Metric Annotation and Inter-Annotator Agreement

For the human annotations in Table 6, three annotators independently label each evaluation dimension on 200 sampled cases using binary scores in $\{0, 1\}$. We measure human–human agreement by strict three-rater unanimity: a case–metric pair receives an agreement indicator of 1 if all three labels are identical, and 0 otherwise, including a two-versus-one split. Let $\mathcal{P}$ denote the set of annotated case–metric pairs and $\ell_{j,p}$ the label from annotator j for pair p. The overall Agreement Ratio is

$$\text{AgreementRatio} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathbf{1}[\ell_{1,p} = \ell_{2,p} = \ell_{3,p}] = 0.67. \tag{6}$$

This 67% measures unanimous agreement among human annotators.

C SKILL EVOLUTION ALGORITHM DETAILS

Algorithm 1 Skill Evolution

Require: Train dataset $\mathcal{D}^{\text{train}} = \{\mathcal{D}_k^{\text{train}}\}_{k=1}^K$, where each case i contains input x_i; fixed foundation skill set $\mathcal{F}$; initial creator skill C_0; score function ϕ; execution agent $\mathcal{A}_{\text{exec}}$; optimizer agent $\mathcal{A}_{\text{opt}}$; merge agent $\mathcal{A}_{\text{merge}}$

Ensure: Optimized composition skills $\mathcal{S}^\star = \{\mathcal{S}_k^\star\}_{k=1}^K$ and optimized creator skill $C^\star$

$\mathcal{D}^{\text{opt}}, \mathcal{S}_0 \leftarrow \text{INFERENCE}(\mathcal{D}^{\text{train}}, C_0, \mathcal{F}, \mathcal{A}_{\text{exec}}, \phi)$
$\mathcal{S}^\star, \{C_k^\star\}_{k=1}^K \leftarrow \text{EVOLVE}(\mathcal{D}^{\text{opt}}, \mathcal{S}_0, C_0, \mathcal{A}_{\text{opt}})$
$C^\star \leftarrow \text{MERGECREATORS}(\{C_k^\star\}_{k=1}^K, \mathcal{A}_{\text{merge}})$
return $\mathcal{S}^\star, C^\star$
function INFERENCE($\mathcal{D}^{\text{train}}, C_0, \mathcal{F}, \mathcal{A}_{\text{exec}}, \phi$)
  for all task categories $k \in \{1, \ldots, K\}$ **in parallel do**
    $\mathcal{S}_{0,k} \leftarrow \mathcal{A}_{\text{exec}}(C_0, \mathcal{F}, \mathcal{D}_k^{\text{train}})$
    for all cases $i \in \mathcal{D}_k^{\text{train}}$ **in parallel do**
      $(\tau_i, y_i) \leftarrow \mathcal{A}_{\text{exec}}(x_i, \mathcal{S}_{0,k})$
      $s_i \leftarrow \phi(x_i, \tau_i, y_i)$
      Store (x_i, τ_i, y_i, s_i) in $\mathcal{D}_k^{\text{opt}}$
    end for
  end for
  $\mathcal{D}^{\text{opt}} \leftarrow \{\mathcal{D}_k^{\text{opt}}\}_{k=1}^K, \mathcal{S}_0 \leftarrow \{\mathcal{S}_{0,k}\}_{k=1}^K$
  return $\mathcal{D}^{\text{opt}}, \mathcal{S}_0$
end function
function EVOLVE($\mathcal{D}^{\text{opt}}, \mathcal{S}_0, C_0, \mathcal{A}_{\text{opt}}$)
  for all task categories $k \in \{1, \ldots, K\}$ **in parallel do**
    Initialize $C_{0,k} \leftarrow C_0$ and context $h_k \leftarrow \emptyset$
    for all cases $i \in \mathcal{D}_k^{\text{opt}}$ **sequentially do**
      $z_i = (h_k, \{x_i, \tau_i, y_i, s_i, \mathcal{S}_{i-1,k}, C_{i-1,k}\})$
      $(\mathcal{S}_{i,k}, C_{i,k}) \leftarrow \mathcal{A}_{\text{opt}}(z_i)$; $h_k \leftarrow h_k \oplus (\mathcal{S}_{i,k}, C_{i,k})$
    end for
    Archive $\mathcal{S}_k^\star \leftarrow \mathcal{S}_{|\mathcal{D}_k^{\text{opt}}|,k}$ and $C_k^\star \leftarrow C_{|\mathcal{D}_k^{\text{opt}}|,k}$
  end for
  $\mathcal{S}^\star \leftarrow \{\mathcal{S}_k^\star\}_{k=1}^K$
  return $\mathcal{S}^\star, \{C_k^\star\}_{k=1}^K$
end function
function MERGECREATORS($\{C_k^\star\}_{k=1}^K, \mathcal{A}_{\text{merge}}$)
  Initialize $C_{\text{acc}} \leftarrow C_1^\star$ and merge context $h_{\text{merge}} \leftarrow \emptyset$
  for $k = 2$ to K **do**
    $z_k = (h_{\text{merge}}, \{C_{\text{acc}}, C_k^\star\})$
    $C_{\text{acc}} \leftarrow \mathcal{A}_{\text{merge}}(z_k)$; $h_{\text{merge}} \leftarrow h_{\text{merge}} \oplus C_{\text{acc}}$
  end for
  $C^\star \leftarrow C_{\text{acc}}$
  return $C^\star$
end function

D FULL-PRECISION MAIN RESULTS

Tables 8 and 9 report the full-precision values underlying the rounded results in Tables 3 and 4. Best and second-best results are determined using these full-precision values.

Table 8: Full-precision results on the test set for the harness and model comparison in Table 3.

Baseline	Process Metrics							Output Metrics								Ranking Score
	EE	CM	IP	PL	SF	RAU	Avg	FR	PR	VC	RF	AC	AVC	PLC	Avg	
OpenClaw + Seed2.0	**0.9750**	**0.9857**	0.8861	0.8691	0.9213	0.8953	0.9221	0.7284	0.2553	0.4929	0.5816	0.5674	0.6050	0.6312	0.5517	2.96
Claude Code + Seed2.0	0.8581	0.9710	0.7237	0.4783	0.6252	0.4928	0.6915	0.7154	0.4388	**0.5504**	0.5683	0.5072	0.6691	0.5072	0.5652	3.41
Claude Code + DeepSeek V4	0.9438	0.8404	0.8156	0.7849	0.8567	0.8258	0.8445	0.8355	0.3010	0.4464	0.5606	0.5225	0.7543	0.7197	0.5914	4.45
Claude Code + Opus 4.7	0.9749	0.9856	**1.0000**	0.8951	**0.9657**	0.9612	**0.9638**	0.9123	0.4055	0.3814	0.5533	0.4845	0.7835	0.7629	0.6119	4.64
Codex + GPT-5.5	**0.9750**	0.9849	0.9867	**0.9137**	0.9112	**0.9698**	0.9569	**0.9843**	**0.4580**	0.5210	**0.6050**	**0.6261**	**0.8403**	**0.8025**	**0.6910**	**4.90**

Table 9: Full-precision results on the test set for the skill evolution comparison in Table 4.

Baseline	Process Metrics							Output Metrics								Ranking Score
	EE	CM	IP	PL	SF	RAU	Avg	FR	PR	VC	RF	AC	AVC	PLC	Avg	
No Composition Skill	0.9753	**0.9919**	0.5708	0.4651	–	0.8195	0.7645	0.7893	0.2536	0.3949	0.5652	0.4819	0.7717	0.5616	0.5455	2.82
Composition Skill	0.9750	0.9857	0.8861	0.8691	0.9213	0.8953	0.9221	0.7284	0.2553	0.4929	**0.5816**	**0.5674**	0.6050	0.6312	0.5517	2.96
Self-Evolution	**0.9864**	0.9858	**0.9959**	**0.9890**	**0.9877**	0.9823	**0.9879**	0.9047	0.3011	0.4444	0.5806	0.3978	0.6918	0.6487	0.5670	4.59
Evolution with Feedback	0.9860	**0.9919**	0.9707	0.9853	0.9845	**0.9911**	0.9849	0.9128	**0.3416**	0.4733	0.5445	0.4484	0.7011	0.6548	0.5824	5.23
Expert Composition Skill	0.9652	0.9708	0.9481	0.9848	0.9640	0.9860	0.9698	**0.9283**	0.3187	**0.4982**	0.5275	0.5238	**0.7985**	**0.7436**	**0.6198**	**5.63**

Task completeness and consistency. The output metrics in Table 9 should be interpreted jointly with task completeness. A relevant failure mode is incomplete generation: for example, the initial Composition Skill may produce only two clips totaling 30 seconds for a 60-second request. Such an output fails the duration requirement and may omit requested events or narrative transitions, affecting Format Requirements (FR), Prompt Requirements (PR), and Plot Logic and Coherence (PLC). At the same time, fewer clips and a shorter temporal span provide fewer opportunities for visual, reference, and audio drift, potentially yielding relatively high Visual Consistency (VC), Reference Fidelity (RF), and Audio Consistency (AC) scores on the content that is generated. This offers a possible explanation for the mixed changes from Composition Skill to Evolution with Feedback: FR, PR, and Audio-Video Consistency (AVC) increase by 0.1844, 0.0863, and 0.0961, respectively, and PLC improves more modestly by 0.0236, while VC, RF, and AC decrease.

E Order Effect Ablation

To address the possible order effect, we conduct additional experiments with different random orders. We use three different random seeds to shuffle both the order of case disclosure and the order of skill merging. The evaluation protocol is kept unchanged.

Table 10: Results under different random orders of case disclosure and creator skill merging. Values are mean $\pm$ standard deviation over three random seeds. Mean Δ is the difference between the reported means (Evolution with Feedback minus Self-Evolution).

Metric	Evolution with Feedback	Self-Evolution	Mean Δ
Process Metrics			
EE	0.9367 ± 0.0608	0.9935 ± 0.0112	-0.0568
CM	0.9465 ± 0.0927	0.9455 ± 0.0480	$+0.0010$
IP	0.9701 ± 0.0253	0.9529 ± 0.0510	$+0.0172$
PL	0.9131 ± 0.0845	1.0000 ± 0.0000	-0.0869
SF	0.9959 ± 0.0070	0.9560 ± 0.0387	$+0.0399$
RAU	0.9359 ± 0.0402	0.9089 ± 0.0508	$+0.0270$
Avg	0.9497 ± 0.0163	0.9595 ± 0.0141	-0.0098
Output Metrics			
FR	0.9216 ± 0.0558	0.8866 ± 0.0234	$+0.0350$
PR	0.2697 ± 0.0392	0.3511 ± 0.1251	-0.0814
VC	0.5669 ± 0.0474	0.5032 ± 0.1071	$+0.0637$
RF	0.5640 ± 0.1255	0.5672 ± 0.0913	-0.0032
AC	0.5562 ± 0.0381	0.4499 ± 0.0720	$+0.1063$
AVC	0.6919 ± 0.1079	0.6356 ± 0.1024	$+0.0563$
PLC	0.5796 ± 0.0361	0.6239 ± 0.0578	-0.0443
Avg	0.5928 ± 0.0189	0.5739 ± 0.0422	$+0.0189$

Averaged over the three random orders, Evolution with Feedback achieves a higher Output Avg. than Self-Evolution, with a mean difference of $+0.0189$.

F Dataset Details

We construct a benchmark of 16 long-video generation task categories with 285 cases in total. Following the split used in the main experiments, 13 categories are treated as in-distribution categories and split into 119 training cases and 123 held-out test cases, while three categories are held out as OOD categories with 43 cases. Each category is associated with a task-level skill query that describes the intended generation capability and the expected input-output pattern. At 285 cases, VideoWeaver is of the same order of magnitude as existing skill-related and real-world agentic benchmarks shown in Table 1. Moreover, video generation tasks are generally costly to construct and evaluate, especially for long-video generation. Scaling the benchmark much further would potentially increase computation costs, making the benchmark less accessible and harder to reproduce. We therefore choose the current scale as a practical balance between task coverage, evaluation cost, and accessibility.

F.1 Task Collection and Filtering

The user tasks and textual requirements were collected through a platform purposely built for this benchmark. The platform allows users to submit video generation requests with any input modalities, including text, images, videos, audio, and their combinations. Users can also specify constraints such as duration, resolution, aspect ratio, visual style, reference usage, and others. We then filtered the data by keeping approximately one-minute tasks and removing ambiguous requests, grouping similar tasks together, yielding the final benchmark.

F.2 Dataset Case Examples

Figure 5 shows two representative cases from the benchmark. The first case illustrates a long-video editing task conditioned on both a reference image and a reference video, where the agent must replace the target object in the final video segment while preserving the original scene context. The second case illustrates an audio-driven story generation task conditioned on a reference text and a reference audio clip, where the generated visual storyline should follow the narration and align scene transitions with the audio.



Figure 5: Representative dataset cases. Top: a long-video editing case with reference image and reference video inputs. Bottom: an audio-driven story generation case with reference text and reference audio inputs.

F.3 DATASET SKILL QUERIES

Table 11 lists the dataset categories and their corresponding skill queries. The names and queries are taken from the `skill_query.json` file under each dataset directory.

Table 11: Dataset categories and task queries. Categories below the separator are reserved as OOD categories.

Name	Query
AI Avatar Video	Generate AI-avatar talking-head videos from narration scripts and avatar reference videos, preserving the speaker appearance, style, voice, target duration, aspect ratio, and other user-specified constraints.
One-Sentence Anime Video	Generate anime or animated story videos from a short plot description and user-specified constraints such as duration, aspect ratio, and visual style.
Product Unboxing Video	Generate product unboxing videos from product reference images and textual introductions, including product presentation, feature explanation, and close-up demonstrations.
SSL Class	Generate children's teaching videos from scripts, teaching requirements, and character references, producing classroom-style instructional clips with coherent teacher-student interactions.
Anime Video	Generate anime videos from text and reference anime videos, preserving the reference visual style, tone, color palette, and story requirements.
Character Anime Video	Generate character-consistent anime story videos from reference character illustrations and plot text, supporting customized style, scene, duration, and whole-video character consistency.
Object Evolution Video	Generate coherent videos showing an object, creature, or scene evolving across time from multi-stage reference images and textual descriptions.
Product Video Demo	Generate product demonstration videos from product screenshots and feature descriptions, with narration and visual explanation of key product functions.
Cinematic Video	Generate cinematic videos from scripts or long story outlines, following user-specified duration, style, scenes, and visual details.
Ancient Poetry Teaching	Generate ancient-poetry teaching animations from poem text and reference images, with narration, pedagogical structure, and coherent visual presentation.
Audio-Driven Story	Generate story videos aligned with a provided narration audio file and style references, matching the audio content, timing, and duration.
E-Commerce Subject Replace	Replace the product subject or visual style in e-commerce livestream-style short videos while preserving the original selling logic, duration, and rhythm.
Long Video Edit	Edit long videos according to text instructions and reference materials, preserving duration and the required identity, scene, object, audio, and background constraints.
OOD Categories	
AI Avatar Tech Content Video	Generate technology-oriented AI-avatar explanation videos from textual requirements, product reference images, and an avatar reference video, preserving the avatar identity, voice, and product-specific content.
Theme Video	Generate long-form theme videos from a topic or short text, supporting knowledge explanation, scenery montage, and story narration under user-specified duration and format constraints.
Brand Promotion Video	Generate high-quality brand promotion videos from brand references and textual introductions, following user-specified visual style, music, narrative rhythm, and brand consistency requirements.

G FOUNDATION SKILLS

Table 12 lists the foundation skills provided to the agent. These skills expose basic video, image, audio, understanding, metadata, and media-processing capabilities. They are independently invocable and form the primitive action space over which composition skills define longer procedural workflows.

For generation skills, the video generation skill uses ByteDance Doubao Seedance 2.0, with `doubao-seedance-2-0-fast-260128` as the default model and `doubao-seedance-2-0-260128` as the higher-quality standard variant. The image generation skill uses Doubao Seedream 5.0 by default (`doubao-seedream-5-0-260128`) and also supports `gpt-image-2`. For multimodal perception, the vision-understanding skill uses the Volcengine Ark Doubao multimodal model `doubao-seed-2-0-pro-260215`, while the audio-understanding skill uses `doubao-seed-2-0-lite-260428`, which supports native audio input. Speech recognition is implemented with Volcengine BigASR using the `volc.bigasr.auc_turbo` resource and `bigmodel` recognition backend. Text-to-speech uses OpenSpeech TTS V3 with `seed-tts-2.0` as the default resource. Non-generative media-processing skills, such as metadata extraction, frame extraction, shot splitting, and grid construction, rely on deterministic tools including ffprobe, exiftool, OpenCV, PySceneDetect, and Pillow rather than learned foundation models. We adopt this ByteDance-backed full-modality tool stack consistently across experiments, as it provides unified access to video, image, audio, speech, and multimodal understanding capabilities, thereby simplifying experimental control, reproducibility, and future research extensions. These models are all publicly available through API calls.

Table 12: Foundation Skills.

Name	Description
add-audio-track	Add an extra audio track to a video while preserving the original audio track.
audio-gen	Generate speech audio for narration or dialogue; background music and sound effects are outside this skill's scope.
audio-understanding	Analyze audio content with a multimodal audio model, including speaker timbre, BGM style, emotion, continuity, silence, and audio consistency.
audio-vocal-separate	Separate vocals from background music or accompaniment in an audio or video file, producing clean vocal and background tracks.
automatic-speech-recognition	Transcribe local or online audio/video files into text, JSON, or SRT subtitles.
change-fps	Convert a video to a target frame rate.
extract-video-frame	Extract a specific frame from a video, such as the first, middle, last, frame-indexed, or timestamped frame.
get-image-metadata	Extract image metadata such as resolution, format, file size, and EXIF information.
get-output-dir	Return the standardized output directory for the current generation task.
get-video-metadata	Extract video metadata such as duration, resolution, frame rate, and frame count.
image-gen	Generate or edit images from prompts, reference images, resolution, aspect ratio, model, and seed settings.
merge-video	Merge multiple videos after normalizing resolution, aspect ratio, and frame rate.
pair-wise-skill-merge	Progressively merge an incoming skill or skill creator into an accumulator skill directory while deduplicating and resolving conflicts.
replace-audio	Replace a video's original audio with a specified audio file.
resize-image-resolution	Resize an image to a target resolution.
resize-video-resolution	Resize a video to a target resolution.
split-audio	Separate a video's audio track from its visual stream.
split-image	Split an image into a 2×2 grid, commonly used to preprocess human reference images for video generation.
text-to-speech	Synthesize speech audio from text with configurable voice, speed, volume, pitch, and optional word timestamps.
trim-audio	Trim an audio file to a specified start and end time.
trim-video	Trim a video to a specified start and end time.
video-gen	Generate or edit videos with video-generation APIs, prompt rewriting, reference inputs, preprocessing, and iterative logging.
video-shot-split	Count and split video shots into separate clip files.
vision-understanding	Analyze image or video content and answer visual understanding questions.

H BENCHMARK EXECUTION COSTS

We report the cost in USD for three representative harness/model combinations used in our main experiments. For Claude Code + Opus 4.7 and Codex + GPT-5.5, we use the coding plan price, where the LLM backend cost is a fixed 200 USD subscription cost for running the full 285-case benchmark, which is equivalent to 0.7018 USD per case.

Table 13: Average cost per case in USD for three representative harness/model combinations.

Harness / Model	**Image**	**Video**	**Model**	**Total**
Claude Code + Opus 4.7	0.0194	9.3016	0.7018	10.0228
Claude Code + DeepSeek	0.0633	9.2084	0.1853	9.4570
Codex + GPT-5.5	0.0244	9.7484	0.7018	10.4746

Running one case costs approximately 9.5–10.5 USD. This cost is dominated by video-generation API calls.

Why we do not enforce a matched-cost baseline. We do not enforce an identical monetary budget across methods because resource consumption is an endogenous outcome of agentic orchestration. Different composition skills may choose different numbers and types of generation calls, intermediate artifacts, inspection steps, and recovery attempts. Imposing a fixed budget would therefore alter the execution policy itself and may terminate a method before it satisfies the same duration and content requirements, conflating budget exhaustion with generation quality.