VideoWeaver: Evaluating and Evolving Skills for Agentic Long Video Generation

Anonymous ACL submission

Abstract

Recent agent frameworks such as Claude Code, Codex, and OpenClaw are strong at tool use and orchestration, but whether they can handle long video generation, a long-horizon multimodal task, remains underexplored. Unlike earlier video agents whose pipeline is handcrafted, these frameworks can build and refine their own workflows. We introduce VideoWeaver, an agent harness and benchmark that evaluates and evolves skills for long video generation, where an agent turns a single instruction into a long video by composing foundation skills into its own workflow rather than following a predefined pipeline. The benchmark has 16 task categories and 285 cases, with references spanning text, image, audio, video, and their combinations. Because errors can arise at any stage and not just in the final video, we propose an agent-as-judge that inspects both the execution trace and the final video, grounding its scores in evidence such as metadata and intermediate files. Using this feedback, we further design a skill evolution algorithm that refines and merges the agent's skills. Across multiple frameworks and models, we find that an explicit composition skill improves the generation process over using foundation skills alone, that skill evolution further improves output quality, and that performance varies notably across harness and model choices. The proposed agent-as-judge also aligns well with human judgments, especially 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 (Yuan et al., 2024; Yang et al., 2025; Zhang et al., 2026d) 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 heterogeneous generation and editing tools, and maintaining narrative, visual, and audio consistency across many clips. It can even use generation models themselves as callable tools. 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 Mora (Yuan et al., 2024), StoryAgent (Pan et al., 2024), UniVA (Yang et al., 2025) and Co-Director (Zhang et al., 2026d) demonstrate this promise, but they are purpose-built systems whose orchestration follows fixed, human-designed workflows. Meanwhile, general-purpose agent frameworks such as Claude Code (Anthropic, 2026a), Codex (OpenAI, 2026b), and OpenClaw (OpenClaw, 2026) pair a reasoning backbone with a tool-augmented runtime, show strong tool-use and orchestration, and can build and refine their own workflows directly from a task. Whether such general harnesses can handle long video generation has not been systematically explored.

To study this, we introduce VideoWeaver, a benchmark for long video generation built around an agent harness. In our harness, a backbone foundation model (OpenAI, 2026a; 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 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), split into in-distribution train, test sets and three out-of-distribution categories.

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. 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., 2026c) judge only the final output (Table 1), leaving these process-level failures undiagnosed. 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 6 process and 7 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. We make the following contributions.

- To the best of our knowledge, we are the first to study general-purpose agent harnesses for long video generation, and we build a benchmark of 16 task categories and 285 cases with foundation skills and expert references.
- We propose a new agent-as-judge method that diagnoses both execution traces and final outputs, achieving broader process coverage than prior video benchmarks (Table 1).
- We refine composition skills and creator skills from evaluation feedback and further imporve the long video genertaion quality and generalize to unseen cases.

Benchmark	Process	Output
StoryAgent (Pan et al., 2024)	✗	✓ (5)
ScriptBench (Mu et al., 2026)	✗	✓ (5)
Hollywood Town (Wei et al., 2025)	✗	✓ (6)
GenAD-Bench (Zhang et al., 2026d)	✗	✓ (4)
ThemeVlogEval (Hou et al., 2025)	✗	✓ (12)
AutoMV (Tang et al., 2025)	✗	✓ (12)
ST-Bench (Zhang et al., 2025b)	✗	✓ (5)
StreamingT2V (Henschel et al., 2025)	✗	✓ (4)
HeliosBench (Yuan et al., 2026)	✗	✓ (9)
AniEval (Shi et al., 2025)	✗	✓ (4)
UniVA-Bench (Yang et al., 2025)	✓ (3)	✓ (6)
VideoWeaver (Ours)	**✓ (6)**	**✓ (7)**

Table 1: Comparison of metric coverage across video generation benchmarks. Numbers indicate the number of metrics in each group.

2 Related Works

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). Representative frameworks such as VideoDirectorGPT (Lin et al., 2023), Mora (Yuan et al., 2024), and UniVA (Yang et al., 2025) use hierarchical planning to turn scripts into scenes, shots, and prompts, while StoryAgent (Pan et al., 2024) and Co-Director (Zhang et al., 2026d) add critique, persona specialization, and feedback-based revision. These systems are effective but rely on fixed, human-designed workflows, which motivates

agents that can build and evolve their own.

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). LLLM-as-judge benchmarks such as Video-Bench (Han et al., 2025), UniVBench (Wei et al., 2026), and CLVG-Bench (Zhang et al., 2026c) 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., 2026). 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 (Hu et al., 2026; Chen et al., 2026a,b), treats markdown or behavioral skills as persistent, self-improving memory (Liu et al., 2026a,b; Jiang et al., 2026; Yang et al., 2026; Wang et al., 2026b; Xu et al., 2026), and validates self-evolving skills in industrial and medical settings (Zhang et al., 2026b; Liu et al., 2026d; Li et al., 2026b). A parallel line organizes and maintains skill libraries at scale through encapsulation, hierarchical knowledge bases, retrieval, dependency management, and active curation (Kaleb et al., 2026; Zhu et al., 2026; Gao et al., 2026; Liu et al., 2026c; Zhang et al., 2026a; Li et al., 2026a; Wang et al., 2026c; L. et al., 2026), while others couple skills with training, memory, and policy optimization (Lu et al., 2026; Xia et al., 2026; Shi et al., 2026; Wang et al., 2026a).

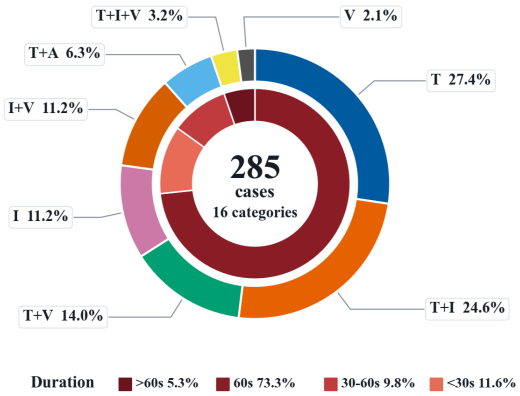

Figure 1: Dataset distribution of input modality and required duration. Modality abbreviations: T: Text, I: Image, V: Video, A: Audio.

3 Task Definition

3.1 Dataset Construction

We focus on **long video generation**, where the target video generation requires a sequence of actions. Starting from a one-time user instruction, the agent plan and execute multi-step skill invocations, manage intermediate outputs, and progressively compose them into a coherent final video.

We construct a dataset comprising **16 categories** of long video generation tasks and **285 test cases** in total. 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 previously unseen task categories. Representative task cases are shown in App. C.1.

3.2 Skills Workspace

In our harness environment, the backbone language model (e.g., GPT-5.5 (OpenAI, 2026a), Seed2.0 (ByteDance Seed, 2026)) has only text generation and reasoning capabilities, but does not natively support video generation or multimodal

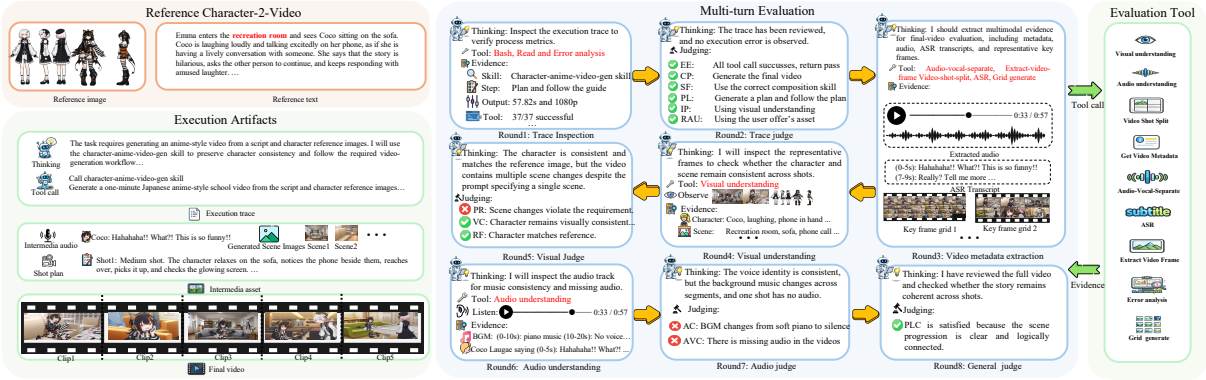

Figure 2: Evidence-grounded agent-as-judge evaluation framework. Given the user request, reference assets, execution trace, intermediate artifacts, and final video, the evaluation agent conducts a multi-turn evaluation. At each round, it dynamically invokes evaluation skills to inspect traces, extract 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.

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. For example, generating video clips and synthesizing images. The full set of foundation skills is provided in App. D.

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 multiple steps, and preserving narrative, visual, and audio consistency across clips. 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.

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 works (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 assess the quality of the final video. Unlike prior evaluations (Huang et al., 2023; Bansal et al., 2024) that mainly focus on 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.

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

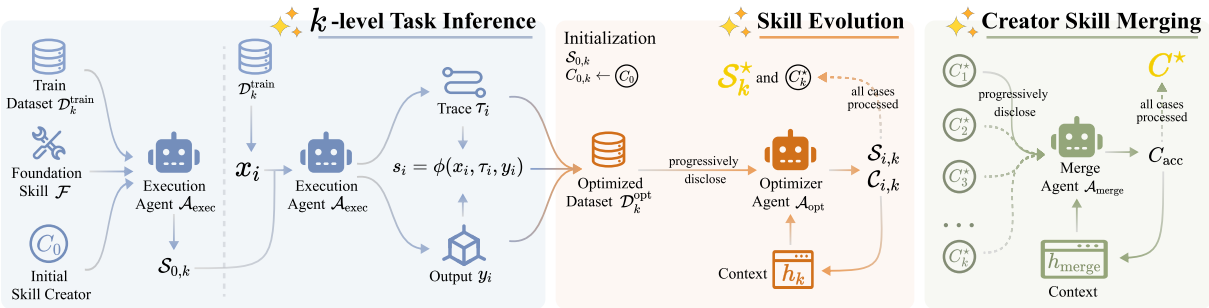

Figure 3: Overview of our skill evolution pipeline. The execution agent $\mathcal{A}_{\text{exec}}$ first performs task inference to generate initial category-level composition skills $\mathcal{S}_{0,k}$ and collect execution traces τ, outputs y, and evaluation feedback s_i. The optimizer agent $\mathcal{A}_{\text{opt}}$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

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 hard constraints such as duration, aspect ratio, and resolution are satisfied.
Prompt Requirements	PR	Whether the generated video satisfies user-specified content and instruction requirements.
Visual Consistency	VC	Whether characters and scenes remain consistent across clips.
Audio Consistency	AC	Whether speaker identity and background music style remain consistent across clips.
Audio-Video Consistency	AVC	Whether audio and video are synchronized and all required audio tracks are present.
Plot Logic and Coherence	PLC	Whether the story is understandable and logically connected.
Reference Fidelity	RF	Whether the final video reflects multimodal references.

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

evidence, confidence level, and textual feedback. These scores and feedbacks are mainly used as refinement signals for skill evolution.

For final performance comparison, we use human annotators to assess video quality. For each input, annotators rank n candidate outputs from different baselines and assign ordinal scores from $n-1$ to 0. 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. Figure 4 in Appendix shows the annotation interface.

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.

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 $\mathcal{D}_k$ contains 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 a output y, the score function $\phi(x, \tau, y)$ is instantiated by the evaluation agent and measures execution and output quality.

Our first objective is to optimize K category-level composition skills for held-out test cases:

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

where $\mathcal{A}_{\text{exec}}(x, \mathcal{S}_k) = (\tau, y)$. The optimized skill set is $\mathcal{S}^\star = \{\mathcal{S}_k^\star\}_{k=1}^K$.

Our second objective is to optimize a creator skill for OOD categories. Let $\mathcal{D}^{\text{ood}} = \{\mathcal{D}_k^{\text{ood}}\}_{k=1}^{K_{\text{ood}}}$

denote the OOD split. For each OOD category, the creator first produces a composition skill,

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

The creator skill is optimized as

$$C^\star = \arg\max_{C} \sum_{k=1}^{K_{\text{ood}}} \sum_{x \in \mathcal{D}_k^{\text{ood}}} \phi\big(x, \mathcal{A}_{\text{exec}}(x, \mathcal{S}_{C,k})\big). \tag{4}$$

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 $(\mathcal{S}_{i,k}, C_{i,k})$. $\mathcal{A}_{\text{opt}}$ ***back-propagates*** 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 Experiment

5.1 Baselines

For the execution agent, we evaluate multiple harness environments, including Claude Code (Anthropic, 2026a), CodeX (OpenAI, 2026b), and the open-source OpenClaw framework (OpenClaw, 2026), paired with multiple foundation models, including GPT-5.5 (OpenAI, 2026a), DeepSeek V4 (DeepSeek-AI, 2026), Opus 4.7 (Anthropic, 2026b), and Seed 2.0 (ByteDance Seed, 2026). Unless otherwise specified, skill evolution uses OpenClaw with Seed 2.0 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 evolves the skill using only the optimization objectives defined by the process and output metrics, without additional feedback from the judging agent.
- **Evolution with Feedback:** the agent evolves the skill using feedback generated by judge agent during optimization.
- **Expert Skill:** expert constructed composition skill baseline.

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.

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.

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	0.6108	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 3: Composition skill baselines with different harness and model results on long video generation. Best results are in **bold** and second-best results are underlined.

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	**0.9965**	0.5708	0.4651	–	0.8195	0.7654	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	**0.9965**	0.9707	0.9853	0.9845	**0.9911**	0.9857	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**

Table 4: Main results on skill evolution with OpenClaw + Seed2.0. Best results are in **bold** and second-best results are underlined.

Harness and Model Comparison: Table 3 shows that performance varies substantially across different harness-model pairs. Mature 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 compares all variants 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 already captures many trace-visible procedural fixes, and external feedback mainly provide valuable output metrics feedbacks that are harder for the agent to self evaluate from the its own execution.

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

Table 6 shows that the optimized skills (with agent feedbacks) 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%), indicating reusable orchestration improvements rather than overfitting. On OOD categories, the optimized creator also wins more than it loses (23.8% vs. 16.5%), suggesting transfer of learned orchestration experience to unseen tasks. The relatively high tie rates may be attributed to 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

To validate the reliability of Agent-as-Judge, we compare the scores against human annotations on 200 sampled cases. As shown in Table 7, We report exact match, Kendall's τ_b, and mean absolute error (MAE). 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. Execution Errors are evaluated by script-based checks.

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

Table 5: Resource usage statistics for each baseline. RGT denotes Relative Generation Time, and VC-RGT denotes Vision Content Relative Generation Time.

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%

Table 6: Pairwise generalization results. Test categories compare optimized composition skills with their initial versions, while OOD categories compare optimized and initial creator-generated skills.

Type	Metric	Exact	τ_b	MAE
Process	CM	0.995	0.892	0.005
	IP	0.917	0.763	0.083
	PL	0.921	0.792	0.079
	SF	0.949	0.773	0.037
	RAU	0.970	0.845	0.029
Output	FR	0.945	0.815	0.073
	PR	0.679	0.490	0.179
	VC	0.678	0.563	0.173
	RF	0.768	0.601	0.129
	AC	0.702	0.509	0.167
	AVC	0.804	0.540	0.111
	PLC	0.763	0.390	0.126

Table 7: Alignment between Agent-as-Judge scores and human annotations on 200 sampled cases.

5.5 Resource Usage Statistics

Table 5 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.

6 Conclusion

We presented VideoWeaver, an agent harness and benchmark for long video generation. Our work makes three main contributions: first, we formulate long video generation as an agentic skill-composition task and build a benchmark with 16 categories and 285 cases; second, we propose an evidence-grounded agent-as-judge that evaluates both execution traces and final videos; third, we introduce a skill evolution algorithm that refines composition and creator skills from evaluation feedback. Experiments across multiple harnesses and models show that explicit composition skills and skill evolution improve long video generation, and that our automatic judge aligns well with human judgments, especially on process metrics.

Limitations

Our work has several limitations. First, the current dataset contains only hundreds of cases, mainly due to the high computational cost of generating and evaluating approximately one-minute videos; future work will scale the cases and cover a broader range of task categories. Second, the current foundation skills are primarily built on models and tools within the ByteDance ecosystem. Although the harness itself is model-agnostic, future versions should incorporate more diverse backends, including open-source models, third-party generation APIs, and broader video, image, audio, and editing tools. Third, future work will explore more principled optimization methods, better credit assignment over execution traces, and stronger skill verification to further improve robustness and generalization.

Potential Risks

VideoWeaver may involve generating long videos that use copyrighted reference assets or resemble protected content, which raises potential intellectual-property concerns. In addition, long video generation requires multiple model calls and media-processing steps, leading to substantial computational cost. Practical deployment should include provenance tracking, consent and copyright checks, watermarking, safety filters, human oversight, and transparent resource reporting.

References

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

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

Hritik Bansal, Zongyu Lin, Tianyi Xie, Ziniu Zhang, Yizhou Yu, Juan Carlos Niebles, Bernard Ghanem, and Bo Li. 2024. Videophy: Evaluating physical commonsense for video generation. *arXiv preprint arXiv:2406.03520*.

Shraddha Barke, Arnav Goyal, Alind Khare, Avaljot Singh, Suman Nath, and Chetan Bansal. 2026. Agentrx: Diagnosing ai agent failures from execution trajectories. *Preprint*, arXiv: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. 2024. Video generation models as world simulators.

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

S. Chen, J. Gai, and 1 others. 2026a. Coevoskills: Self-evolving agent skills via co-evolutionary reflection. *Computing Research Repository*, arXiv:2604.01687.

Z. Chen and 1 others. 2026b. Trace2skill: Distilling transferable skills from execution trajectories. *Computing Research Repository*, arXiv:2603.25158.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. 2026. Lol: Longer than longer, scaling video generation to hour. *Preprint*, arXiv:2601.16914.

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

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. 2025. Trail: Trace reasoning and agentic issue localization. *Preprint*, arXiv:2505.08638.

X. Gao and 1 others. 2026. Skillrouter: Retrieve-and-rerank skill selection for llm agents. *Computing Research Repository*, arXiv:2603.22455.

Google. 2024. Google i/o 2024: Introducing veo and imagen 3 generative ai tools.

Hui Han, Siyuan Li, Jiaqi Chen, and 1 others. 2025. Video-bench: Human-aligned video generation benchmark. *arXiv preprint arXiv:2504.04907*.

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

X. Hou, B. Ma, J. Cheng, X. Ren, K. Yu, W. Li, T. Zheng, and Q. Lu. 2025. Personavlog: Personalized multimodal vlog generation with multi-agent collaboration and iterative self-correction. *arXiv preprint arXiv:2508.13602*.

S. Hu, C. Lu, J. Clune, and 1 others. 2026. Evoskill: Automated skill discovery for multi-agent systems. *Computing Research Repository*, arXiv:2603.02766.

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. 2023. Vbench: Comprehensive benchmark suite for video generative models. *arXiv preprint arXiv:2311.17982*.

G. Jiang, J. Zhang, and 1 others. 2026. Xskill: Continual learning from experience and skills in multimodal agents. *Computing Research Repository*, arXiv:2603.12056.

Jiaxiu Jiang, Wenbo Li, Jingjing Ren, Yuping Qiu, Yong Guo, Xiaogang Xu, Han Wu, and Wangmeng Zuo. 2025. Lovic: Efficient long video generation with context compression. *Preprint*, arXiv:2507.12952.

K. Kaleb, S. Chen, J. Gai, and 1 others. 2026. Skillcraft: Can llm agents learn to use tools skillfully? *Computing Research Repository*, arXiv:2603.00718.

Yuan L. and 1 others. 2026. Skillnet: Create, evaluate, and connect ai skills. *Computing Research Repository*, arXiv:2603.04448.

J. Li, W. Zhao, and 1 others. 2026a. Skillflow: Benchmarking lifelong skill discovery and evolution for agents. *Computing Research Repository*, arXiv:2604.17308.

L. Li and 1 others. 2026b. Abstral: Automated multi-agent system design via skill abstraction. *Computing Research Repository*, arXiv:2603.22791.

Wuyang Li, Wentao Pan, Po-Chien Luan, Yang Gao, and Alexandre Alahi. 2025. Stable video infinity: Infinite-length video generation with error recycling. *Preprint*, arXiv:2510.09212.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Han Lin and 1 others. 2023. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. *arXiv preprint arXiv:2309.15091*.

Jie Liu and 1 others. 2026a. Memento-skills: Let agents design agents. *arXiv preprint arXiv:2603.18743*.

Jie Liu and 1 others. 2026b. Metaclaw: Just talk – an agent that meta-learns and evolves in the wild. *arXiv preprint arXiv:2603.17187*.

X. Liu and 1 others. 2026c. Skilldex: A package manager and registry for agent skill bundles. *Computing Research Repository*, arXiv:2604.16911.

Y. Liu and 1 others. 2026d. Evolving medical imaging agents via experience-driven self-skill acquisition. *Computing Research Repository*, arXiv:2603.05860.

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. *arXiv preprint arXiv:2303.16634*.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. 2026. Skill0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*.

C. Mu, X. He, Q. Yang, W. Chen, J. Yao, H. Liu, Z. Yi, B. Zhao, X. Chen, R. Ma, and 1 others. 2026. The script is all you need: An agentic framework for long-horizon dialogue-to-cinematic video generation. *arXiv preprint arXiv:2601.17737*.

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

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

OpenClaw. 2026. What is openclaw? `https://openclawdoc.com/docs/getting-started/what-is-openclaw/`. Accessed: 2026-05-24.

Y. Pan and 1 others. 2024. Storyagent: Customized storytelling video generation via multi-agent collaboration. *Computing Research Repository*, arXiv:2411.04925.

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

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

Haochen Shi, Xingdi Yuan, and Bang Liu. 2026. Evolving programmatic skill networks. *arXiv preprint arXiv:2601.03509*.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2025. Defining and characterizing reward hacking. *Preprint*, arXiv:2209.13085.

Yiwen Song, Tomas Pfister, and Yale Song. 2026. Vqqa: An agentic approach for video evaluation and quality improvement. *Preprint*, arXiv:2603.12310.

X. Tang, X. Lei, C. Zhu, S. Chen, R. Yuan, Y. Li, C. Oh, G. Zhang, W. Huang, E. Benetos, and 1 others. 2025. Automv: An automatic multi-agent system for music video generation. *arXiv preprint arXiv:2512.12196*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, and 1 others. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Juntong Wang, Haoyue Zhao, Guanghui Pan, Xiyuan Wang, Yanbo Wang, Qiyan Deng, and Muhan Zhang. 2026a. Sage: A self-evolving agentic graph-memory engine for structure-aware associative memory. *arXiv preprint arXiv:2605.12061*.

K. Wang and 1 others. 2026b. Memskill: Learning and evolving memory skills for self-evolving agents. *Computing Research Repository*, arXiv:2602.02474.

K. Wang and 1 others. 2026c. Skillsbench: Evaluating how well agent skills work across diverse tasks. *Computing Research Repository*, arXiv:2602.12670.

Jianhui Wei, Xiaotian Zhang, Yichen Li, and 1 others. 2026. Univbench: Towards unified evaluation for video foundation models. *arXiv preprint arXiv:2602.21835*.

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

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. 2026. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*.

B. Xu and 1 others. 2026. Memrl: Self-evolving agents via runtime reinforcement learning on episodic memory. *Computing Research Repository*, arXiv:2601.03192.

Jiazheng Xu, Yu Huang, Jiale Cheng, and 1 others. 2024. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. *arXiv preprint arXiv:2412.21059*.

X. Yang and 1 others. 2025. Univa: Universal video agent towards open-source next-generation video generalist. *Computing Research Repository*, arXiv:2511.08521.

Y. Yang, J. Li, Q. Pan, and 1 others. 2026. Autoskill: Experience-driven lifelong learning via skill self-evolution. *Computing Research Repository*, arXiv:2603.01145.

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

Z. Yuan and 1 others. 2024. Mora: Enabling generalist video generation via a multi-agent framework. *Computing Research Repository*, arXiv:2403.13248.

Fan Zhang, Shulin Tian, Ziqi Huang, Yu Qiao, and Ziwei Liu. 2025a. Evaluation agent: Efficient and promptable evaluation framework for visual generative models. *Preprint*, arXiv:2412.09645.

H. Zhang and 1 others. 2026a. Skillos: Learning skill curation for self-evolving agents. *Computing Research Repository*, arXiv:2605.06614.

Kaiwen Zhang, Liming Jiang, Angtian Wang, Jacob Zhiyuan Fang, Tiancheng Zhi, Qing Yan, Hao Kang, Xin Lu, and Xingang Pan. 2025b. Storymem: Multi-shot long video storytelling with memory. *Preprint*, arXiv:2512.19539.

Q. Zhang and 1 others. 2026b. Skillforge: Forging domain-specific, self-evolving agent skills in llm agents. *Computing Research Repository*, arXiv:2604.08618.

Xiaotian Zhang, Jianhui Wei, Yuan Wang, and 1 others. 2026c. How far are video models from true multimodal reasoning? *arXiv preprint arXiv:2604.19193*.

Y. Zhang and 1 others. 2026d. Co-director: Agentic generative video storytelling with collaborative critique. *Computing Research Repository*, arXiv:2604.24842.

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. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

T. Zhu, J. Li, and 1 others. 2026. Skillx: Automatically constructing skill knowledge bases for agents. *Computing Research Repository*, arXiv:2604.04804.

A Human Annotation Details

A.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. 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. The resulting order is converted into ordinal scores as described in Sec. 4.1.

All annotators were compensated at a rate of 10 USD per hour. The annotation task only 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.

B 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 $C^\star$

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

C 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.

C.1 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.

C.2 Dataset Skill Queries

Table 8 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.

D Foundation Skills

Table 9 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 gen-

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.

Table 8: Dataset categories and task queries. Categories below the separator are reserved as OOD categories.

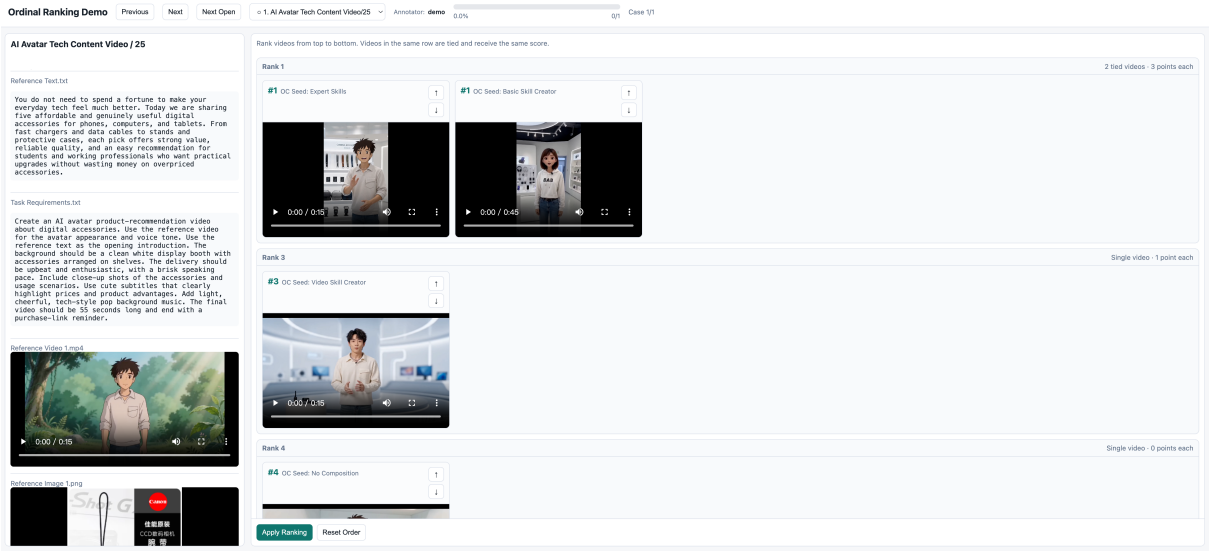

Figure 4: Screenshot of the ranking annotation interface used in human evaluation. Annotators compare candidate videos generated by different baselines for the same input and assign ordinal rankings by dragging videos or using the up/down controls.

eration 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.

E AI Assistant Usage

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.

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 2x2 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.

Table 9: Foundation Composition Skills.

Long Video Edit Case: Image + Video (30s)

Reference image | Reference video | Edit target

Reference object

Reference video frames

Requirement

Replace the blue trash bin that appears at the end of the reference video with the trash bin shown in the reference image, while preserving the original scene and motion context.

Audio-Driven Story Case: Text + Audio (33s)

Reference text | Reference audio | Timing-aligned video

Reference style text

2D anime style, warm literary lighting, gentle healing atmosphere, delicate nostalgic shots, and restrained emotional expression.

Reference audio

Audio drives both content progression and shot boundaries.

Requirement

Generate video scenes synchronized with the input narration: a blurred memory of past conflict, a later reunion and apology, reconciliation while walking side by side, and a warm close-up of the two friends toasting at the end.

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.