CoEvoWhen: Policy-Tool Coevolution for Ultra-Long Video Temporal Grounding

Anonymous authors
Paper under double-blind review

Abstract

Ultra-long video temporal grounding requires balancing long-range evidence search with fine-grained event understanding under a limited visual budget, yet existing agentic methods still rely largely on predefined policies and tool capabilities. Motivated by this, we propose a novel policy–tool coevolution framework that jointly evolves high-level policies and executable media tools from the agentic reasoning trajectories of a VLM, forming a reusable skill without updating model parameters. During evolution, an external skill updater distills transferable task experience in long-video temporal grounding, accordingly refining the orchestration of long-range image-based and fine-grained video-based observations. Alongside these policy updates, the updater employs its coding capabilities to upgrade existing tools or create new ones, adapting the tools to long-video evidence acquisition. Equipped with the evolved skill, the VLM autonomously orchestrates tools under the guidance of the evolved policy, coordinating image and video observations for agentic inference without relying on a separate, stronger planning model. Extensive experiments spanning five benchmarks and three VLMs show that policy–tool coevolution consistently improves temporal grounding accuracy in ultra-long videos while reducing visual token cost at inference, and that the evolved skill yields substantial performance gains on general long-video QA without additional task-specific evolution, demonstrating the effectiveness and generalizability of our framework for long-video understanding.

1 Introduction

Multimodal agents that support footage retrieval, clip extraction, and editing assistance must not only understand what happens in a video, but also translate users' semantic intent into concrete segments on the timeline for subsequent operations (Vidi Team et al., 2025). Video temporal grounding, which determines the time intervals of events described in natural language, is a fundamental capability bridging video understanding and downstream video operations (Gao et al., 2017; Lei et al., 2021). However, as videos extend to tens of minutes or even hours, target events may occupy only a tiny fraction of the timeline, so accurate grounding requires both searching for relevant content across a vast temporal range and discerning event details, action changes, and temporal boundaries locally (Soldan et al., 2022; Hannan et al., 2025; Seo & Kim, 2026). Under a limited visual budget, coarse observation may overlook brief events or key details, whereas repeated fine-grained inspection incurs high visual cost (Buch et al., 2025; Hong et al., 2025). Balancing long-range evidence search with fine-grained event understanding is therefore a central challenge in ultra-long video temporal grounding.

From an agentic perspective, ultra-long video temporal grounding can be organized as a process of actively acquiring, comparing, and verifying visual evidence through tool interactions, rather than a one-shot prediction on a fixed video input (Liu et al., 2026). Conditioned on the query and accumulated observations, an agent can dynamically decide which time range to search next, which candidates to examine, and whether additional evidence is needed to confirm the event and its boundaries. Existing studies have demonstrated the potential of multi-step search and tool use, yet the underlying observation mechanisms and media processing capabilities remain largely predefined (Wang et al., 2026). Tools are not merely interfaces for executing instructions; they also determine what visual input the model actually receives (Hu et al., 2024). Beyond studying how an agent uses existing tools, it is therefore necessary to consider how the tools themselves should be

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.

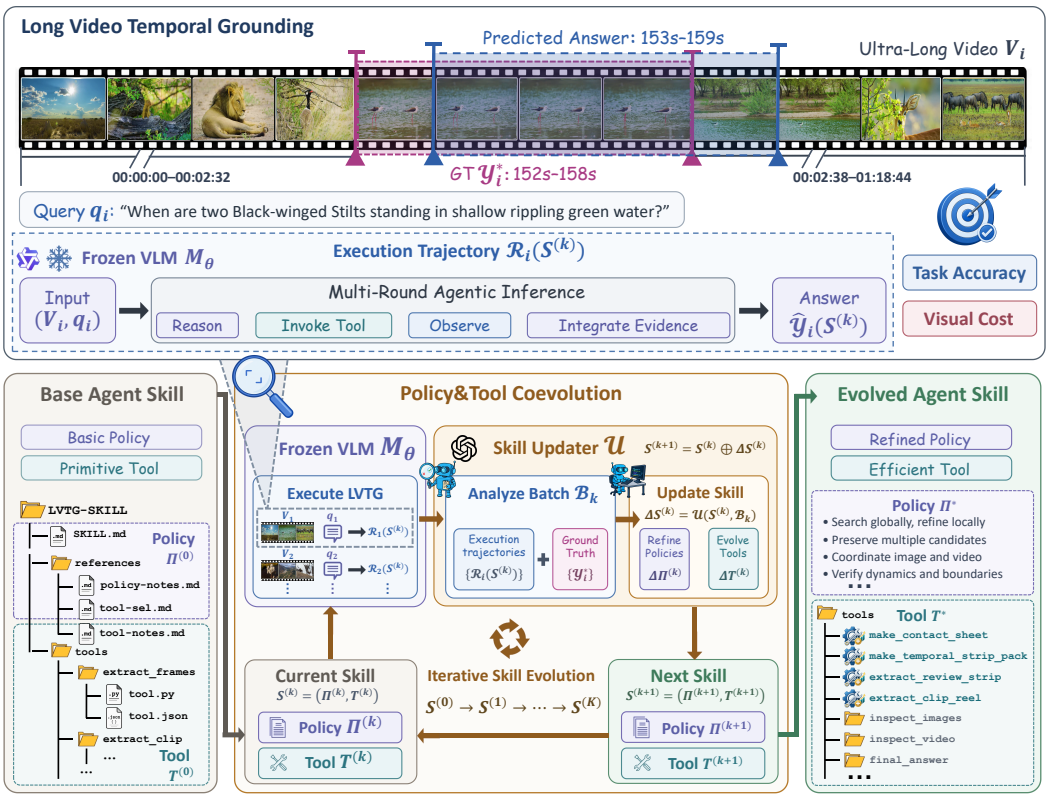

Figure 1: **Policy–tool coevolution framework for ultra-long video temporal grounding.** A frozen VLM executes grounding tasks with an external skill comprising high-level policies and executable media tools. Across evolution rounds, an external skill updater jointly refines the policies and evolves the tools from execution trajectories, synthesizing code to upgrade existing tools or create new ones.

designed to better support search and verification. Moreover, evidence needs vary across videos and queries (Wu et al., 2024), making it difficult to predesign a general pipeline that balances grounding accuracy with visual cost. This raises the central question of this work: *how can task experience be leveraged to jointly improve a video agent's tool capabilities and the policies guiding their use, so that grounding evidence can be acquired more accurately and efficiently?*

Our key insight is that tool-use policies and tool design are interdependent: high-level policies determine what evidence is needed, when to acquire it, and how to organize the search, while media tools determine the form and granularity of the evidence the tools can present to the model. For example, comparing temporally distant candidates calls for image-based observations that compactly cover long temporal ranges while preserving their temporal correspondence, whereas judging action order, state changes, and event continuity may require local video-based observations (Ye et al., 2025; Wu et al., 2025; Li et al., 2024). Effectively exploiting these complementary capabilities requires policies that select and orchestrate observations according to evidence needs, together with tools that support the corresponding sampling and presentation. Refining policies alone may be limited by existing tool capabilities, while expanding tools alone does not necessarily enable an agent to make effective use of the new capabilities. We therefore explore coevolving policies and tools from the same execution feedback, enabling mutual adaptation between observation capabilities and their orchestration. Candidate omissions, boundary errors, and dynamic ambiguities exposed in execution trajectories can inform not only adjustments to search decisions but also improvements to media tools, thereby accumulating task experience in both policies and executable capabilities.

Building upon these insights, we propose CoEvoWhen, a policy–tool coevolution framework for ultra-long video temporal grounding. The framework jointly represents high-level policies and ex-

ecutable media tools as a reusable external skill, and iteratively improves it from task experience without updating the parameters of the vision-language model (VLM). Starting from a minimal base skill that provides only the basic grounding protocol, initial observation instructions, and primitive image and video observation operations, the VLM executes grounding tasks to generate trajectories, which are paired with the corresponding ground-truth temporal intervals to form task feedback. An external skill updater then distills transferable experience from this feedback, refining policies for task planning and observation orchestration while synthesizing code to modify existing media tools or create new ones, so that improvements extend beyond prompts and invocation sequences to the actual evidence sampling and presentation capabilities. The evolved skill supports image-based search, candidate refinement, and selective video verification. At inference, the VLM autonomously invokes tools from the fixed skill based on the query and accumulated evidence, without relying on a separate, stronger planning model.

We conduct systematic evaluations on three ultra-long video temporal grounding benchmarks and two long-video question answering (QA) benchmarks. The results demonstrate that coevolution improves grounding accuracy while simultaneously reducing visual token cost. Notably, on ExtremeWhenBench (Seo & Kim, 2026), evolving the base skill for Qwen3.5-27B (Qwen Team, 2026a) raises mIoU by **74.9%**, while reducing the average cumulative visual token cost by **11.4%**. Consistent gains are also observed when skills are evolved separately on different VLMs. Moreover, applying the evolved grounding skill directly to long-video QA improves performance without additional task-specific evolution, demonstrating the cross-task reusability of the accumulated experience. Ablation studies further show that coevolution attains a better combination of accuracy and visual cost than evolving either policies or tools alone, supporting the rationale for jointly adapting observation capabilities and their usage policies.

In summary, our main contributions are as follows:

- **1)** We propose CoEvoWhen, which jointly evolves high-level policies and executable media tools from a VLM's execution trajectories, distilling ultra-long video grounding experience into a reusable external skill without updating model parameters.
- **2)** We evolve the orchestration of complementary image-based and video-based observations, enabling the VLM to acquire evidence autonomously without manually predefined coordination strategies or a stronger external planner.
- **3)** We conduct systematic experiments across five benchmarks and three VLMs, complemented by policy–tool and image–video ablations, substantiating the effectiveness and generalizability of our framework, as well as the transferability of the evolved skill to general long-video understanding.

2 Related Work

2.1 Long-Video Temporal Grounding

Video temporal grounding aims to identify temporal intervals corresponding to natural language queries in untrimmed videos (Mu et al., 2024). Early methods typically rely on precomputed video features to predict these intervals in a single pass (Zhang et al., 2020; Moon et al., 2023). VTimeLLM (Huang et al., 2024), TimeChat (Ren et al., 2024), and UniTime (Li et al., 2025) further enhance the temporal awareness of generative multimodal models through grounding-specific adaptation, enabling direct generation of query-relevant timestamps or intervals. As video duration grows to tens of minutes or even several hours, CONE (Hou et al., 2023), SOONet (Pan et al., 2023), and ReVisionLLM (Hannan et al., 2025) narrow the temporal search space through query-guided window selection, single-pass scanning, and recursive refinement, respectively. More flexible agentic frameworks acquire query-relevant visual evidence dynamically: VideoAgent (Wang et al., 2025c) employs an LLM agent to iteratively identify relevant visual information, VideoTree (Wang et al., 2025d) constructs a query-adaptive hierarchical video representation, and DVD (Zhang et al., 2025b) uses an LLM to plan and orchestrate visual tools. Although these frameworks advance evidence acquisition from one-shot prediction to multi-step search and tool use, their underlying media processing capabilities remain largely predefined. We instead use the VLM's temporal grounding trajectories to jointly evolve media tools and the coordination of image-based and video-based observations, enabling more accurate and efficient grounding in ultra-long videos.

2.2 Self-Evolving Agent Skills

Self-evolving skill frameworks turn execution trajectories and interaction feedback into reusable policies, programs, or tools, allowing task experience to accumulate without updating model parameters (Shinn et al., 2023; Wang et al., 2024; Zhang et al., 2025a). For task planning and tool use, ExpeL (Zhao et al., 2024) distills execution experience into reusable natural language guidance, while XSkill (Jiang et al., 2026) organizes multimodal experience into task-level skills. Experience can also be retained in executable form, with Voyager (Wang et al., 2023) accumulating successful programs as reusable code skills for retrieval and composition across tasks. These approaches advance experience reuse, but do not explicitly couple high-level policy refinement with updates to underlying tool implementations. Although SkillSmith (Wei et al., 2026) adapts both skills and tools, it restricts tool updates to predefined operations on the existing tool library. For long-video understanding, META (Huang et al., 2026) abstracts tool trajectories into reusable macro-tools and refines tool-specific usage constraints, but leaves the task-level orchestrator outside the evolution loop. In contrast, our framework jointly evolves high-level policies and executable media tools, adapting visual sampling and presentation alongside observation orchestration to yield a reusable skill that the frozen VLM executes autonomously without relying on a stronger external planner.

3 Method

3.1 Problem Formulation

Given an ultra-long video V_i of duration L_i and a natural language query q_i, video temporal grounding aims to precisely localize all temporal intervals that semantically correspond to q_i. For a frozen VLM M_θ equipped with an external skill $\mathcal{S}$, we denote the ground-truth and predicted interval sets by

$$\mathcal{Y}_i^* = \{[s_{in}, e_{in}]\}_{n=1}^{N_i}, \quad \widehat{\mathcal{Y}}_i(\mathcal{S}) = M_\theta(V_i, q_i; \mathcal{S}) = \{[\widehat{s}_{in}, \widehat{e}_{in}]\}_{n=1}^{\widehat{N}_i},$$

where $0 \le s_{in} < e_{in} \le L_i$, and $N_i = 0$ indicates that the target event is absent from the video. Starting from a base skill $\mathcal{S}^{(0)}$, we evolve it on the evolution set while keeping the model parameters θ fixed, and evaluate the resulting skill $\mathcal{S}^*$ by both grounding performance and visual token cost.

3.2 Policy–Tool Coevolution

As shown in Figure 1, we jointly evolve the high-level policies and executable media tools within the external skill over K evolution rounds. At each evolution round, the frozen VLM uses the current skill to execute a batch of ultra-long video temporal grounding tasks. An external skill updater then analyzes the resulting trajectories to update the policies and tools.

3.2.1 Skill Representation

At evolution round k, the external skill $\mathcal{S}^{(k)}$ comprises high-level policies $\Pi^{(k)}$ and a set of executable media tools $\mathcal{T}^{(k)}$:

$$\mathcal{S}^{(k)} = \left(\Pi^{(k)}, \mathcal{T}^{(k)}\right), \qquad \mathcal{T}^{(k)} = \{\tau_j^{(k)}\}_{j=1}^{J_k}.$$

$\Pi^{(k)}$ guides task planning and observation orchestration, specifying what visual evidence is needed and how it is acquired. $\mathcal{T}^{(k)}$ provides executable media tools adapted to diverse observation contexts. Each tool $\tau_j = (d_j, \sigma_j, f_j)$ consists of a description d_j of its capability and intended use, an interface specification σ_j, and source code f_j. J_k denotes the current number of tools and changes as tools are created, consolidated, or retired.

3.2.2 Skill Update

Equipped with the current skill $\mathcal{S}^{(k)}$, the frozen VLM M_θ performs temporal grounding for a batch of queries indexed by $\mathcal{I}_k$. Each trajectory $\mathcal{R}_i(\mathcal{S}^{(k)})$ records the complete execution process and final prediction for query q_i. After the batch is completed, these trajectories are paired with the corresponding ground-truth intervals to form the task feedback used to update the skill:

$$\mathcal{B}_k = \left\{\left(\mathcal{R}_i(\mathcal{S}^{(k)}), \mathcal{Y}_i^*\right)\right\}_{i \in \mathcal{I}_k}, \quad \Delta\mathcal{S}^{(k)} = \mathcal{U}\left(\mathcal{S}^{(k)}, \mathcal{B}_k\right) = \left(\Delta\Pi^{(k)}, \Delta\mathcal{T}^{(k)}\right).$$

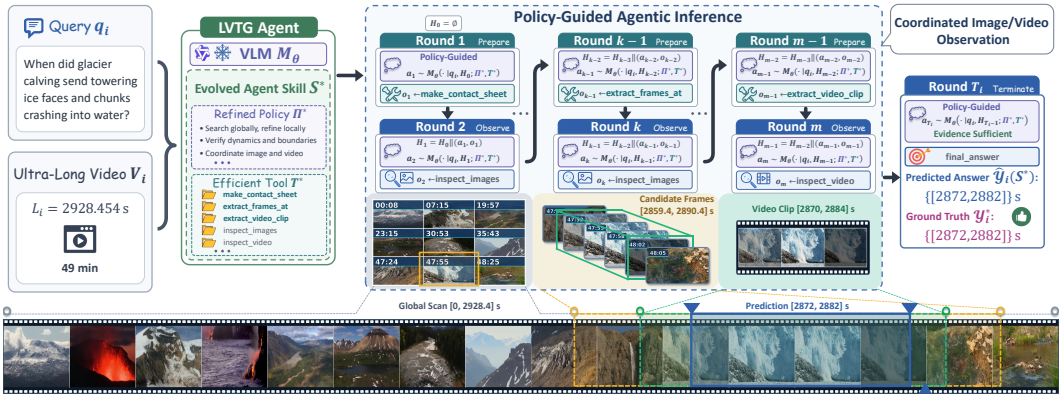

Figure 2: **Policy-guided agentic inference with the evolved skill.** Guided by the evolved policy, the VLM autonomously orchestrates media tools, using accumulated evidence to decide on subsequent observations without relying on a separate, stronger planning model. The illustrated trajectory combines image-based global search and candidate refinement with selective video verification to localize the queried event.

Here, $\mathcal{B}_k$ denotes the feedback batch at evolution round k, and $\mathcal{U}$ denotes the external skill updater.

For policy evolution, $\Delta\Pi^{(k)}$ refines the strategies for task planning and observation orchestration with reusable experience distilled from the feedback. For tool evolution, $\Delta\mathcal{T}^{(k)}$ adapts and expands the media processing capabilities available to the model by upgrading existing executable tools or creating new ones. Existing tools evolve through updates $\Delta\tau_j^{(k)} = (\Delta d_j^{(k)}, \Delta\sigma_j^{(k)}, \Delta f_j^{(k)})$ to their descriptions, interfaces, and source code, while the tool set is restructured through tool creation, capability consolidation, or the retirement of unsuitable tools. We formalize this process as:

$$\Pi^{(k+1)} = \Pi^{(k)} \oplus \Delta\Pi^{(k)}, \quad \mathcal{T}^{(k+1)} = \mathcal{T}^{(k)} \oplus \Delta\mathcal{T}^{(k)} = \left\{\tau_j^{(k)} \oplus \Delta\tau_j^{(k)}\right\}_{j\in\mathcal{J}_k^{\mathrm{ret}}} \cup \mathcal{T}_k^{\mathrm{new}}.$$

Here, $\oplus$ denotes the application of an update patch, $\mathcal{J}_k^{\mathrm{ret}}$ indexes the tools retained after the update, and $\mathcal{T}_k^{\mathrm{new}}$ denotes newly created or consolidated tools. The skill update $\mathcal{S}^{(k+1)} = \mathcal{S}^{(k)} \oplus \Delta\mathcal{S}^{(k)}$ proceeds for K rounds, yielding the final skill $\mathcal{S}^* = \mathcal{S}^{(K)}$.

3.2.3 Coordinated Image–Video Observation

We initialize coevolution with a minimal base skill $\mathcal{S}^{(0)}$ that provides complementary image and video observation primitives without prescribing any sophisticated strategies for task planning or observation orchestration across the two modalities. $\Pi^{(0)}$ specifies only the basic temporal grounding protocol and elementary descriptions of the available observation modes, while $\mathcal{T}^{(0)}$ comprises only basic image, video, and auxiliary operations.

Image-based observations can provide compact coverage of extended temporal ranges and facilitate comparisons across distant candidate regions, whereas video-based observations can preserve local temporal continuity for reasoning about motion, event order, state transitions, and temporal boundaries. During coevolution, the tools are adapted to the observation needs exposed in execution trajectories, while the policies are refined to make effective use of the evolving capabilities. The resulting skill coordinates image-based search and candidate refinement with selective video verification to meet varying observation needs in long-video temporal grounding.

3.3 Policy-Guided Agentic Inference

At inference, the evolved skill $\mathcal{S}^* = (\Pi^*, \mathcal{T}^*)$ is fixed and applied to the same VLM M_θ for ultra-long video temporal grounding. As illustrated in Figure 2, the VLM performs agentic inference autonomously using $\mathcal{S}^*$, without relying on a separate, stronger planning model. For each pair (V_i, q_i), Π^* guides task planning and observation orchestration, while $\mathcal{T}^*$ provides the executable media tools through which the model interacts with V_i.

Starting from an empty interaction history $H_0 = \varnothing$, the model invokes a tool from $\mathcal{T}^*$ at inference round t based on the query q_i, the accumulated history H_{t-1}, and the guidance from Π^*. Each round of model reasoning and tool interaction is appended to the history. The policy-guided agentic inference process is formalized as:

$$a_t = (j_t, \eta_t) \sim M_\theta\left(\cdot \mid q_i, H_{t-1}; \Pi^*, \mathcal{T}^*\right), \quad H_t = H_{t-1} \parallel \left(a_t, f^*_{j_t}(V_i; \eta_t)\right).$$

Here, j_t indexes the selected tool in $\mathcal{T}^*$, η_t denotes invocation arguments conforming to its interface $\sigma^*_{j_t}$, and $\parallel$ denotes history accumulation across inference rounds. The maximum number of inference rounds is set to $T_{\max}$. Once the model considers the accumulated evidence sufficient or reaches this limit, it invokes the termination tool at round $T_i \leq T_{\max}$ to submit the final prediction $\widehat{\mathcal{Y}}_i(\mathcal{S}^*)$.

4 Experiments

4.1 Experimental Setup

Benchmarks, baselines, and metrics. We evaluate ultra-long video temporal grounding on three benchmarks. VUE-LVTR comprises visual queries from VUE-TR (Vidi Team et al., 2025) and VUE-TR-V2 (Vidi Team et al., 2026) restricted to videos of at least 30 minutes. Videos in ExtremeWhenBench (Seo & Kim, 2026) average approximately 76 minutes, while the evaluation on CoMET-Bench (Zou et al., 2026) covers multi-event grounding in videos of at least 30 minutes. We select 100 challenging VUE-LVTR queries from distinct videos for evolution and evaluate the same evolved skill on the three benchmarks. Results on VUE-LVTR are reported on the disjoint held-out set. We also assess long-video QA transfer on LVBench (Wang et al., 2025a) and LSD-Bench (Qu et al., 2025). For baselines, we primarily compare the base and evolved skills under identical task protocols and inference settings, while also including other VLM-based and agent-based methods. These comparisons cover the open-source VLMs Qwen3.5-27B (Qwen Team, 2026a), InternVL3.5-8B (Wang et al., 2025b), and TimeLens-7B (Zhang et al., 2026), as well as the closed-source VLMs Gemini 2.5 Flash (Comanici et al., 2025) and GPT-5.6 Luna (OpenAI, 2026b), with VideoMind-7B (Liu et al., 2026) and EvoGround-7B (Jung et al., 2026) serving as agent-based baselines. Performance on all five benchmarks is measured using their official metrics, and for efficiency, we compute cumulative visual token cost by summing visual tokens received by the VLM across rounds and averaging over queries, with separate image and video costs. Counts of model calls, local tool calls, and visual observations characterize interaction overhead. Detailed baseline evaluation protocols and metric definitions are provided in Appendix B.2.

Evolution and inference. We evolve the skill on Qwen3.5-27B with Codex (GPT-5.5, xhigh) as the external skill updater (OpenAI, 2025; 2026a). The base skill's policy specifies only the basic grounding protocol, without predefined task planning or observation coordination, while its tool set supports only basic image and video observations, media probing, frame extraction, and clip extraction. After each batch of four queries, Codex analyzes the VLM's execution trajectories to revise policy documents and uses its coding capabilities to modify, consolidate, or create executable media tools. We make one pass over the evolution set, keeping VLM weights frozen throughout. During policy-guided agentic inference, we incorporate the evolved policy into the VLM's system prompt and register the skill's tools as callable functions for multi-turn reasoning. Both skills are evaluated with thinking enabled and greedy decoding, using at most 24 inference rounds per query. Further implementation details can be found in Appendices B and F.

4.2 Main Results

4.2.1 Ultra-Long Video Temporal Grounding

As shown in Table 1, the evolved skill achieves the best results on all reported metrics, surpassing seven VLM-based and agent-based baselines. Relative to the base skill, IoU AUC rises from 0.4107 to 0.5137 on the VUE-LVTR held-out set. For ExtremeWhenBench, evolution further improves mIoU and Recall@0.5 by **74.9%** and **88.4%**, respectively, suggesting that the evolved skill supports effective evidence search in hour-long videos and generalizes beyond the evolution data. Results on CoMET-Bench show gains of **0.0280** in mIoU and **10.93** points in Rejection-F1, indicating that the coevolved observation capabilities help the frozen VLM acquire, verify, and select evidence across multi-event and target-absent queries. Together, these gains indicate that policy–tool coevolution

Table 1: **Gains from policy–tool coevolution on ultra-long video temporal grounding. Bold**: best; blue: gains over the base skill, with ↑ indicating relative improvements.

Method	Setting	VUE-LVTR (30.1–105.5 min, mean 50.9 min) Precision AUC ↑	VUE-LVTR Recall AUC ↑	VUE-LVTR IoU AUC ↑	VUE-LVTR IoU @0.5 ↑	ExtremeWhenBench (45.0–542.5 min, mean 75.8 min) Mean IoU ↑	ExtremeWhenBench Recall @0.5 ↑	CoMET-Bench (30.0–123.7 min, mean 50.5 min) Mean IoU ↑	CoMET-Bench Recall @0.5 ↑	CoMET-Bench F1 @0.5 ↑	CoMET-Bench Rejection F1 ↑
Qwen3.5-27B	768 frames	0.2645	0.2657	0.1904	0.1889	0.0392	0.0233	0.0683	0.0708	0.0455	55.05
InternVL3.5-8B	128 frames	0.1452	0.1649	0.0804	0.0619	0.0048	0.0018	0.0098	0.0037	0.0018	57.18
TimeLens-7B	384 frames	0.3665	0.3873	0.2758	0.2866	0.1024	0.0612	0.0277	0.0129	0.0126	72.17
Gemini 2.5 Flash	128 frames	0.1670	0.1815	0.0907	0.0684	0.0089	0.0057	0.0498	0.0369	0.0088	57.41
GPT-5.6 Luna	128 frames	0.3114	0.4347	0.2510	0.2280	0.0444	0.0202	0.0573	0.0310	0.0309	41.46
VideoMind-7B	Agentic	0.2968	0.4223	0.2000	0.1629	0.0221	0.0000	0.0185	0.0059	0.0074	0.00
EvoGround-7B	Agentic	0.0659	0.1174	0.0495	0.0423	0.0050	0.0075	0.0170	0.0113	0.0113	43.65
Qwen3.5-27B + Base Skill		0.4706	0.4789	0.4107	0.4267	0.1507	0.1478	0.1355	0.1440	0.1529	66.13
Qwen3.5-27B + Evolved Skill		**0.5918**	**0.5773**	**0.5137**	**0.5603**	**0.2636**	**0.2785**	**0.1635**	**0.1702**	**0.1831**	**77.06**
Evolution Gain		**+0.1212** ↑25.8%	**+0.0984** ↑20.5%	**+0.1030** ↑25.1%	**+0.1336** ↑31.3%	**+0.1129** ↑74.9%	**+0.1307** ↑88.4%	**+0.0280** ↑20.7%	**+0.0262** ↑18.2%	**+0.0302** ↑19.8%	**+10.93** ↑16.5%

Table 2: **Inference efficiency with the base and evolved skills.** Green (↓) and purple (↑) indicate decreases and increases relative to the base skill.

Benchmark	Method	Token Efficiency (k/query) Visual Tokens ↓	Image Tokens	Video Tokens	Interaction Efficiency (count/query) Model Calls	Local Tool Calls	Visual Obs.	Image Obs.	Video Obs.
VUE-LVTR (30.1–105.5 min, mean 50.9 min)	Qwen3.5-27B + Base Skill	202.58	162.01	40.57	17.03	8.16	8.11	7.35	0.76
	Qwen3.5-27B + Evolved Skill	**141.89**	**141.37**	**0.52**	**9.63**	**4.79**	**3.86**	**3.82**	**0.04**
	Δ (Evolved − Base)	**−60.69** ↓29.96%	**−20.64** ↓12.74%	**−40.05** ↓98.72%	**−7.40** ↓43.5%	**−3.37** ↓41.3%	**−4.25** ↓52.4%	**−3.53** ↓48.0%	**−0.72** ↓94.7%
ExtremeWhenBench (45.0–542.5 min, mean 75.8 min)	Qwen3.5-27B + Base Skill	227.10	179.26	47.84	16.65	8.98	6.93	5.69	1.24
	Qwen3.5-27B + Evolved Skill	**201.17**	197.49	**3.68**	**11.19**	**5.69**	**4.56**	**4.35**	**0.21**
	Δ (Evolved − Base)	**−25.93** ↓11.42%	+18.23 ↑10.17%	**−44.16** ↓92.31%	**−5.46** ↓32.8%	**−3.29** ↓36.6%	**−2.37** ↓34.2%	**−1.34** ↓23.6%	**−1.03** ↓83.1%
CoMET-Bench (30.0–123.7 min, mean 50.5 min)	Qwen3.5-27B + Base Skill	119.66	67.57	52.09	11.71	6.51	4.63	3.01	1.62
	Qwen3.5-27B + Evolved Skill	**97.10**	91.65	**5.45**	**8.88**	**4.77**	**3.42**	3.07	**0.35**
	Δ (Evolved − Base)	**−22.56** ↓18.85%	+24.08 ↑35.64%	**−46.64** ↓89.54%	**−2.83** ↓24.2%	**−1.74** ↓26.7%	**−1.21** ↓26.1%	+0.06 ↑2.0%	**−1.27** ↓78.4%

distills temporal grounding experience into a transferable skill for acquiring long-video evidence more accurately without updating model parameters.

4.2.2 Inference Efficiency

Table 2 illustrates the improvements in inference efficiency achieved through policy–tool coevolution. Average visual tokens per query drop from 202.6k to 141.9k (**30.0%** fewer) on the VUE-LVTR held-out set, with reductions of **11.4%** on ExtremeWhenBench and **18.9%** on CoMET-Bench. Model calls also decrease across all three benchmarks, suggesting that the VLM equipped with the evolved skill makes more efficient and targeted decisions about how to observe ultra-long videos. A closer examination of the ExtremeWhenBench results reveals that tokens for image-based observations increase from 179.3k to 197.5k, whereas those for video-based observations fall from 47.8k to 3.7k, reflecting the reallocation of the visual budget during coevolution.

4.2.3 Cross-VLM Generalization and Cross-Task Transfer

For different VLM backbones, as summarized in Table 3, policy–tool coevolution improves ExtremeWhenBench mIoU by **99.2%** and **24.7%** on Qwen3.5-9B (Qwen Team, 2026a) and Qwen3.6-27B (Qwen Team, 2026b), respectively, which supports the framework's cross-VLM generalizability. In addition, when the evolved skill is applied directly to general long-video QA, Qwen3.5-27B's

Table 3: **Generalization across VLMs and transfer to long-video QA. Bold**: results with the evolved skill; blue: gains over the base skill, with ↑ indicating relative improvements.

VLM	Method	VUE-LVTR 30.1–105.5 min (mean 50.9 min) Precision AUC ↑	VUE-LVTR Recall AUC ↑	VUE-LVTR IoU AUC ↑	ExtremeWhenBench 45.0–542.5 min (mean 75.8 min) Mean IoU ↑	ExtremeWhenBench Recall @0.5 ↑	CoMET-Bench 30.0–123.7 min (mean 50.5 min) Mean IoU ↑	CoMET-Bench Rejection F1 ↑	LVBench LVQA Overall Acc. (%) ↑	LSDBench LVQA Overall Acc. (%) ↑
Qwen3.5-9B	+ Base Skill	0.3240	0.3425	0.2405	0.0626	0.0576	0.0672	67.51	40.09	49.08
	+ Evolved Skill	**0.4287**	**0.4707**	**0.3244**	**0.1247**	**0.1170**	**0.0852**	**68.80**	**45.84**	**53.68**
	Evolution Gain	**+0.1047**	**+0.1282**	**+0.0839**	**+0.0621**	**+0.0594**	**+0.0180**	**+1.29**	**+5.75**	**+4.60**
		↑32.3%	**↑37.4%**	**↑34.9%**	**↑99.2%**	**↑103.1%**	**↑26.8%**	**↑1.9%**	**↑14.3%**	**↑9.4%**
Qwen3.5-27B	+ Base Skill	0.4706	0.4789	0.4107	0.1507	0.1478	0.1355	66.13	45.45	61.27
	+ Evolved Skill	**0.5918**	**0.5773**	**0.5137**	**0.2636**	**0.2785**	**0.1635**	**77.06**	**54.10**	**69.25**
	Evolution Gain	**+0.1212**	**+0.0984**	**+0.1030**	**+0.1129**	**+0.1307**	**+0.0280**	**+10.93**	**+8.65**	**+7.98**
		↑25.8%	**↑20.5%**	**↑25.1%**	**↑74.9%**	**↑88.4%**	**↑20.7%**	**↑16.5%**	**↑19.0%**	**↑13.0%**
Qwen3.6-27B	+ Base Skill	0.5098	0.5465	0.4506	0.1691	0.1694	0.1312	68.53	48.55	61.12
	+ Evolved Skill	**0.5519**	**0.5727**	**0.4985**	**0.2109**	**0.2129**	**0.1477**	**72.37**	**54.36**	**65.64**
	Evolution Gain	**+0.0421**	**+0.0262**	**+0.0479**	**+0.0418**	**+0.0435**	**+0.0165**	**+3.84**	**+5.81**	**+4.52**
		↑8.3%	**↑4.8%**	**↑10.6%**	**↑24.7%**	**↑25.7%**	**↑12.6%**	**↑5.6%**	**↑12.0%**	**↑7.4%**

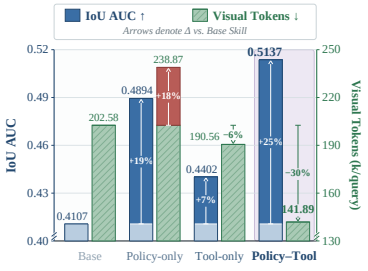

Figure 3: **Policy–tool coevolution ablation** on the VUE-LVTR held-out set. Arrows show relative changes from the base skill.

Image Obs.	Video Obs.	Qwen3.5-27B	Precision AUC ↑	Recall AUC ↑	IoU AUC ↑	Visual Tokens ↓	Image / Video Tokens
✓	✗	+ Base Skill	0.4903	0.4698	0.4173	185.49	185.49 / —
		+ Evolved Skill	0.4917	0.5124	0.4374	194.60	194.60 / —
✗	✓	+ Base Skill	0.3474	0.3092	0.2457	693.52	— / 693.52
		+ Evolved Skill	0.5680	0.5653	0.4772	221.98	— / 221.98
✓	✓	+ Base Skill	0.4706	0.4789	0.4107	202.58	162.01 / 40.57
		+ Evolved Skill	**0.5918**	**0.5773**	**0.5137**	**141.89**	**141.37 / 0.52**
		Δ vs. Image-only	**+0.1001**	**+0.0649**	**+0.0763**	**−52.71**	**−53.23** / —
		Δ vs. Video-only	**+0.0238**	**+0.0120**	**+0.0365**	**−80.09**	— / **−221.46**

Table 4: **Image–video coordination ablation** on the VUE-LVTR held-out set. **Bold** marks results with the evolved image+video skill; blue and green denote its performance gains and reductions in visual token cost relative to evolved single-modality skills. Tokens: k/query.

overall accuracy on LVBench and LSDBench improves by **8.65** and **7.98** points, respectively, demonstrating that the coevolved policies and tools can transfer effectively to broader long-video understanding scenarios.

4.3 Ablations and Analysis

4.3.1 Ablation on Policy–Tool Coevolution

To assess the benefit of policy–tool coevolution and each component's contribution, we compare policy-only and tool-only variants that freeze the tools and policies during evolution, respectively. Both use the same evolution set, evolution configuration, and VUE-LVTR held-out evaluation protocol as the full method. As shown in Figure 3, both variants improve grounding accuracy over the base skill, confirming policies and tools as effective evolution targets. Joint evolution achieves the best accuracy–cost combination, exceeding policy-only and tool-only by **0.0243** and **0.0735** in IoU AUC while reducing visual token cost by **40.6%** and **25.5%**, respectively. This reveals that more efficient evidence acquisition from ultra-long videos relies on both executable media tools to expand the available observation capabilities and high-level policies to select and orchestrate them. In particular, freezing the policies during evolution leads to a larger accuracy loss than freezing the tools,

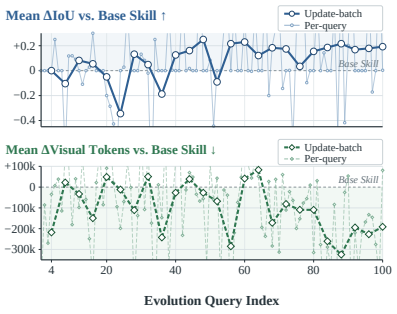

Figure 4: **Online performance–cost dynamics during skill evolution.** Blue and green show changes in IoU and visual tokens relative to the base skill on matched queries.

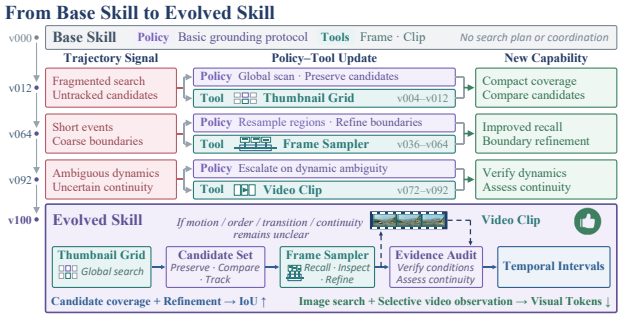

Figure 5: **Coevolution of high-level policies and executable media tools.** Trajectory feedback drives coordinated policy and tool updates, forming an evolved skill that combines image-based global search with selective video verification.

suggesting that stronger observation capabilities can yield substantial gains in ultra-long video temporal grounding when strategically organized into an efficient and accurate evidence search process.

4.3.2 Ablation on Image–Video Coordination

We further conduct image-only and video-only ablations, each restricting the skill's observations to a single modality during evolution and evaluation, to isolate each modality's contribution and assess coordinated image–video observation. As shown in Table 4, both single-modality variants improve grounding performance through evolution, and image–video coordination yields further benefits: the evolved image+video skill outperforms both single-modality skills across all grounding metrics while using fewer visual tokens. These empirical results underscore that image-based and video-based observations are suited to different contexts in ultra-long video temporal grounding, and that strategically coordinating the two makes better use of their complementary strengths, achieving accuracy gains and cost reductions in tandem.

4.3.3 Analysis of Skill Evolution

Figure 4 illustrates online performance–cost dynamics during skill evolution. As evolution proceeds, the skill achieves higher grounding accuracy with lower visual token cost more consistently. Figure 5 further shows how candidate omissions, boundary errors, and dynamic ambiguities exposed in trajectories drive coordinated adjustments to observation capabilities and their orchestration: image-based tools with candidate-retention and local-refinement policies improve evidence coverage and boundary judgment, while video-based observation targets action verification, order discrimination, and continuity assessment. Together, these advances suggest that policy–tool coevolution distills execution feedback into mutually adapted observation capabilities and policies, letting the VLM allocate its visual budget according to evidence needs for more accurate yet cheaper grounding. Qualitative case studies are provided in Appendix D.

5 Conclusion

We propose CoEvoWhen, a policy–tool coevolution framework that jointly evolves high-level policies and executable media tools from agentic reasoning trajectories into a reusable external skill for ultra-long video temporal grounding. Equipped with the evolved skill, the VLM autonomously orchestrates tools under policy guidance, coordinating image-based and video-based observations without relying on a stronger external planner. Comprehensive experiments show that policy–tool coevolution improves grounding accuracy while reducing visual token cost, with gains generalizing across VLMs and transferring to long-video QA without further task-specific evolution. We believe that coevolving what a VLM can observe with how it decides to observe offers a promising path toward seeing less yet understanding more in ultra-long videos.

AI USE STATEMENT

In this work, we used Codex (GPT-5.5, `xhigh`) as the external skill updater to analyze execution trajectories and task feedback, revise policy documents, and synthesize or modify executable media-preparation tools during evolution. We additionally used generative AI to assist with translation and edit parts of the manuscript to improve readability. We have not used generative AI tools for proposing or refining hypotheses, or for supporting qualitative and thematic data analysis, and tasks such as formulating mathematical claims or providing critical ingredients for proving mathematical claims are not applicable to this work. We have reviewed all AI-assisted work, ensuring that it accurately reflects the intended technical meaning. We take responsibility for the final content of this work.

REFERENCES

Shyamal Buch, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. Flexible frame selection for efficient video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29071–29082, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5277–5285, 2017.

Tanveer Hannan, Md Mohaiminul Islam, Jindong Gu, Thomas Seidl, and Gedas Bertasius. ReVisionLLM: Recursive vision-language model for temporal grounding in hour-long videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19012–19022, 2025.

Wenyi Hong, Yean Cheng, Zhuoyi Yang, Weihan Wang, Lefan Wang, Xiaotao Gu, Shiyu Huang, Yuxiao Dong, and Jie Tang. MotionBench: Benchmarking and improving fine-grained video motion understanding for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8450–8460, 2025.

Zhijian Hou, Wanjun Zhong, Lei Ji, Difei Gao, Kun Yan, W.K. Chan, Chong-Wah Ngo, Mike Zheng Shou, and Nan Duan. CONE: An efficient coarse-to-fine alignment framework for long video temporal grounding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8013–8028, 2023.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual Sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*, 37:139348–139379, 2024.

Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. VTimeLLM: Empower LLM to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14271–14280, 2024.

Jing Huang, Luyuan Chen, Zhijie Xu, Yadong Li, Xingzhong Xu, Siye Chen, Jie Liu, Ming Kong, and Qiang Zhu. META: META evolution of tool trajectory adaptation for long-video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9837–9846, 2026.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R. Fung. XSkill: Continual learning from experience and skills in multimodal agents. *arXiv preprint arXiv:2603.12056*, 2026.

Minjoon Jung, Byoung-Tak Zhang, and Lorenzo Torresani. EvoGround: Self-evolving video agents for video temporal grounding. *arXiv preprint arXiv:2605.13803*, 2026.

Jie Lei, Tamara L. Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22195–22206, 2024.

Zeqian Li, Shangzhe Di, Zhonghua Zhai, Weilin Huang, Yanfeng Wang, and Weidi Xie. Universal video temporal grounding with generative multi-modal large language models. *Advances in Neural Information Processing Systems*, 38:64426–64455, 2025.

Ye Liu, Kevin Qinghong Lin, Chang-Wen Chen, and Mike Zheng Shou. VideoMind: A chain-of-LoRA agent for temporal-grounded video reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 57481–57506, 2026.

WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23023–23033, 2023.

Fangzhou Mu, Sicheng Mo, and Yin Li. SnAG: Scalable and accurate video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18930–18940, 2024.

OpenAI. Codex CLI. Software repository, 2025. URL `https://github.com/openai/codex`. Accessed September 26, 2026.

OpenAI. GPT-5.5 System Card. Technical report, OpenAI, 2026a. URL `https://openai.com/index/gpt-5-5-system-card/`.

OpenAI. GPT-5.6 System Card. Technical report, OpenAI, 2026b. URL `https://deploymentsafety.openai.com/gpt-5-6`.

Yulin Pan, Xiangteng He, Biao Gong, Yiliang Lv, Yujun Shen, Yuxin Peng, and Deli Zhao. Scanning only once: An end-to-end framework for fast temporal grounding in long videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13767–13777, 2023.

Tianyuan Qu, Longxiang Tang, Bohao Peng, Senqiao Yang, Bei Yu, and Jiaya Jia. Does your vision-language model get lost in the long video sampling dilemma? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20889–20899, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL `https://qwen.ai/blog?id=qwen3.5`.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026b. URL `https://qwen.ai/blog?id=qwen3.6-27b`.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14313–14323, 2024.

Sukmin Seo and Geewook Kim. Natural-language temporal grounding in hour-long videos is a search problem: A benchmark and empirical decomposition. *arXiv preprint arXiv:2606.12300*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. MAD: A scalable dataset for language grounding in videos from movie audio descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5026–5035, 2022.

Vidi Team, Celong Liu, Chia-Wen Kuo, Dawei Du, Fan Chen, Guang Chen, Jiamin Yuan, Lingxi Zhang, Lu Guo, Lusha Li, et al. Vidi: Large multimodal models for video understanding and editing. *arXiv preprint arXiv:2504.15681*, 2025.

Vidi Team, Chia-Wen Kuo, Chuang Huang, Dawei Du, Fan Chen, Fanding Lei, Feng Gao, Guang Chen, Haoji Zhang, Haojun Zhao, et al. Vidi2.5: Large multimodal models for video understanding and creation. *arXiv preprint arXiv:2511.19529*, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. LVBench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967, 2025a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. VideoAgent: Long-form video understanding with large language model as agent. In *Computer Vision – ECCV 2024*, pp. 58–76, 2025c.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. VideoTree: Adaptive tree-based video representation for LLM reasoning on long videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3283, 2025d.

Ziyang Wang, Honglu Zhou, Shijie Wang, Junnan Li, Caiming Xiong, Silvio Savarese, Mohit Bansal, Michael S. Ryoo, and Juan Carlos Niebles. Active video perception: Iterative evidence seeking for agentic long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9088–9099, 2026.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024.

Yangbo Wei, Zhen Huang, Shaoqiang Lu, Junhong Qian, Qifan Wang, Chen Wu, and Lei He. SkillSmith: Co-evolving skills and tools for self-improving agent systems. *arXiv preprint arXiv:2606.01314*, 2026.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Yongliang Wu, Xinting Hu, Yuyang Sun, Yizhou Zhou, Wenbo Zhu, Fengyun Rao, Bernt Schiele, and Xu Yang. Number it: Temporal grounding videos like flipping manga. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13754–13765, 2025.

Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, et al. Re-thinking temporal search for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8579–8591, 2025.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6543–6554, 2020.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. AFlow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025a.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, and Limin Wang. TimeLens: Rethinking video temporal grounding with multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10419–10429, 2026.

Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Deep video discovery: Agentic search with tool use for long-form video understanding. *Advances in Neural Information Processing Systems*, 38:89863–89895, 2025b.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19632–19642, 2024.

Yuanhao Zou, Arthad Kulkarni, Lucas Toñanez, Lincoln Spencer, Guangyu Sun, Tianxingjian Ding, Andong Deng, Yi Li, Shuangjun Liu, Yuan Li, et al. Conditional multi-event temporal grounding in long-form video. *arXiv preprint arXiv:2606.15320*, 2026.

A Appendix Overview

The appendix is organized as follows:

- **Section B: Implementation Details**
  - **Section B.1: Details on Evolution**
    Evolution set construction, skill initialization, and evolution settings.
  - **Section B.2: Details on Evaluation**
    Data splits, evaluation metrics, inference settings, and baseline evaluation protocols.
- **Section C: More Results**
  - **Section C.1: Results on the Evolution Set**
    Grounding results on the evolution set across three VLMs.
  - **Section C.2: Visual Token Cost on Qwen3.6-27B**
    Cumulative visual token cost and its image and video components.
  - **Section C.3: Results Across Multiple Dimensions**
    Results across multiple query categories.
- **Section D: Case Study**
  Two cases comparing execution trajectories with the base and evolved skills.
- **Section E: Limitations and Future Work**
  Current limitations and future research directions.
- **Section F: Prompts**
  - **Section F.1: Skill Prompts**
    System prompt and the base skill's `SKILL.md`.
  - **Section F.2: Evaluation Task Prompts**
    Prompts for temporal grounding and long-video QA.
  - **Section F.3: Codex Updater Prompt**
    Prompt for Codex to update policies and tools from execution feedback.

B Implementation Details

B.1 Details on Evolution

B.1.1 Evolution Set Construction

We select 100 queries from the purely visual temporal retrieval instances of VUE-TR-V2 (Vidi Team et al., 2026) to form the evolution set. Candidate instances are required to have an available source video of at least 30 minutes, spanning both long and ultra-long (at least 60 minutes) videos.

Selection jointly considers video duration, the number and length of target intervals, target sparsity, and whether the query involves action order, visual detail, or compound conditions. Table 5 summarizes the main characteristics of the evolution set. Among these queries, 80 have sparse target events and 88 contain at least one target interval no longer than 10 seconds. Together, these instances allow the limited set of execution trajectories to cover observation needs ranging from long-range search to short-event localization and multi-candidate handling.

Table 5: **Composition and difficulty characteristics of the evolution set.** The 100 queries come from 100 distinct videos. Sparse-target queries have a total target duration of at most 30 seconds, or at most 1% of the video duration. Boundary-sensitive queries contain at least one target interval no longer than 10 seconds. Video-edge queries have targets within the first or last 30 seconds of the video.

Property	Value
Video Duration (Range / Mean)	30.35–118.61 min / 52.90 min
Keyword / Phrase / Sentence	18 / 34 / 48
Single-Interval / Multi-Interval	56 / 44
Sparse-Target Queries	80
Boundary-Sensitive Queries	88
Video-Edge Queries	30

B.1.2 BASE SKILL

The base skill $\mathcal{S}^{(0)}$ comprises policy documents, tool descriptions and interfaces, and the source code of the local media tools. The policies are organized into `SKILL.md` and three supporting documents: `policy-notes.md` carries guidance for task planning, `tool-selection.md` specifies how tools and observation modes are selected, and `tool-notes.md` records the input–output semantics and usage notes of each tool. Within this structure, the initial policies specify only a basic grounding protocol and elementary descriptions of the observation modes, without prescribing sophisticated strategies for global search, candidate management, or orchestration across modalities.

Table 6 lists the six tools provided by the base skill. Media preparation tools read source-video metadata or produce images and short video clips, which observation tools place in the same VLM's visual context. The `final_answer` tool submits the final prediction. Each tool declares its name, description, and parameter interface in `tool.json`. Local media tools additionally provide an editable implementation in `tool.py`.

Table 6: **Tools in the base skill and their functions.** Image and video observations are performed by the same VLM that executes the task. Local media tools prepare the inputs without calling additional visual models.

Tool	Role	Function
`probe_media`	Media Preparation	Read metadata such as video duration, resolution, and frame rate
`extract_frames_at`	Media Preparation	Extract frames at specified source-video timestamps
`extract_video_clip`	Media Preparation	Extract a video clip without audio from a specified time window
`inspect_images`	Image Observation	Add images to the VLM's visual context
`inspect_video`	Video Observation	Add video to the VLM's visual context
`final_answer`	Prediction	Submit the final answer in the format required by the current task

B.1.3 EVOLUTION SETTINGS

In the main experiments, we use Qwen3.5-27B (Qwen Team, 2026a) for temporal grounding and Codex (GPT-5.5, `xhigh`) (OpenAI, 2025; 2026a) as the external skill updater. Evolution makes a single pass over the 100 queries, forming a feedback batch after every four completed queries to update the policies and media tools. Table 8 lists the VLM configuration and decoding settings. For the cross-VLM experiments, we evolve a separate skill on each of Qwen3.5-9B, Qwen3.5-27B, and Qwen3.6-27B (Qwen Team, 2026b), and compare the base and evolved skills on the corresponding model. For the cross-task experiments, we directly apply the frozen skill evolved for grounding to long-video QA, without additional task-specific evolution.

B.2 DETAILS ON EVALUATION

B.2.1 DATA SPLITS

Table 7 summarizes the query counts of the three temporal grounding benchmarks and their splits for evolution and evaluation. VUE-LVTR is constructed from VUE-TR (Vidi Team et al., 2025) and VUE-TR-V2 (Vidi Team et al., 2026), which contain 1,598 and 1,600 queries, respectively, of which 1,514 and 1,590 have their corresponding source videos available. We select purely visual queries whose corresponding video is at least 30 minutes long, remove 19 duplicate queries shared between the two sources and 6 queries with unavailable source videos, and obtain 407 VUE-LVTR queries, with VUE-TR and VUE-TR-V2 contributing 115 and 292 queries, respectively. Following the procedure described in Appendix B.1.1, we select 100 of the VUE-TR-V2 queries for skill evolution, and the remaining 307 queries, disjoint from the evolution set, form the held-out evaluation set.

We evaluate on the full official test set of ExtremeWhenBench (Seo & Kim, 2026), which comprises 2,273 queries. For CoMET-Bench (Zou et al., 2026), among its 2,789 official queries, we retain those with a corresponding video of at least 30 minutes, yielding an evaluation set of 1,599 queries that covers both target-present and target-absent cases.

Table 7: **Query counts and data splits for evolution and evaluation on the three temporal grounding benchmarks.**

Benchmark	Total queries	Evolution queries	Evaluation queries
VUE-LVTR	407	100	307
ExtremeWhenBench	2,273	0	2,273
CoMET-Bench	2,789	0	1,599

B.2.2 Evaluation Metrics

We follow the evaluation protocol of each benchmark. Let N denote the number of queries in the evaluation set under consideration. Following the notation introduced in Section 3.1, $\mathcal{Y}_i^*$ and $\widehat{\mathcal{Y}}_i$ denote the ground-truth and predicted interval sets, respectively, with N_i and $\widehat{N}_i$ denoting their corresponding interval counts. Here, $\mathbf{1}[\cdot]$ denotes the indicator function.

VUE-LVTR. We follow the VUE-TR evaluation protocol (Vidi Team et al., 2025; 2026). Before scoring, predicted start and end times are rounded down and up to integer seconds, respectively, and overlapping or adjacent intervals are then merged. Let $\widehat{U}_i$ and U_i^* denote the time sets covered by the predicted and ground-truth intervals, respectively, and let $\mu(\cdot)$ denote the total duration of a time set. Temporal precision, recall, and IoU for each query are

$$\mathrm{Prec}_i = \frac{\mu(\widehat{U}_i \cap U_i^*)}{\mu(\widehat{U}_i)}, \quad \mathrm{Rec}_i = \frac{\mu(\widehat{U}_i \cap U_i^*)}{\mu(U_i^*)}, \quad \mathrm{IoU}_i = \frac{\mu(\widehat{U}_i \cap U_i^*)}{\mu(\widehat{U}_i \cup U_i^*)}.$$

A score of zero is assigned whenever the corresponding denominator is zero. These three metrics respectively measure the precision of the predicted intervals, the coverage of the ground-truth event, and the temporal overlap between the two.

AUC is computed from the hit-rate curve over thresholds, using all queries in the evaluation set. For $m \in \{\mathrm{Prec}, \mathrm{Rec}\}$, the hit rate at threshold t is

$$C_m(t) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[m_i \geq t].$$

For IoU AUC, we follow the strict comparison in the official implementation: $C_{\mathrm{IoU}}(t) = N^{-1} \sum_i \mathbf{1}[\mathrm{IoU}_i > t]$. With $t_k = k/100$, trapezoidal integration over $[0, 1]$ at a step size of 0.01 gives

$$\mathrm{AUC}(m) = 0.01 \sum_{k=0}^{99} \frac{C_m(t_k) + C_m(t_{k+1})}{2}.$$

Precision AUC, Recall AUC, and IoU AUC thus summarize the respective hit rates across the full threshold range. At a fixed threshold, IoU@x is defined as

$$\mathrm{IoU@x} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[\mathrm{IoU}_i \geq x], \qquad x \in \{0.3, 0.5, 0.7\}.$$

ExtremeWhenBench. Each query corresponds to a single target interval (Seo & Kim, 2026). Temporal IoU is computed from the intersection and union of the predicted and ground-truth intervals and then aggregated as

$$\mathrm{mIoU} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{IoU}_i, \qquad \mathrm{Recall@x} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[\mathrm{IoU}_i \geq x],$$

where $x \in \{0.3, 0.5, 0.7\}$. Predictions that cannot be parsed as a valid single interval receive zero IoU and remain in the averages. Results for Action, Environment, Object, Reaction, and Scene report mIoU within each query category.

CoMET-Bench. This benchmark covers multi-event grounding, event counting, and queries with absent targets (Zou et al., 2026). Event counting is evaluated on all queries. Given the predicted

event count $\widehat{N}_i$ and ground-truth count N_i, we compute the mean absolute error (MAE) and off-by-one accuracy (OBO), the fraction of queries whose count error is at most one:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\widehat{N}_i - N_i|, \qquad \mathrm{OBO} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}[|\widehat{N}_i - N_i| \le 1].$$

Pearson correlation measures the correlation between the predicted and ground-truth event counts. Denoting their respective means by $\overline{\widehat{N}}$ and $\overline{N}$, we have

$$\rho = \frac{\sum_i(\widehat{N}_i - \overline{\widehat{N}})(N_i - \overline{N})}{\sqrt{\sum_i(\widehat{N}_i - \overline{\widehat{N}})^2}\sqrt{\sum_i(N_i - \overline{N})^2}}.$$

Grounding metrics are computed for positive queries, indexed by $\mathcal{D}_+ = \{i : N_i > 0\}$, and macro-averaged over this set. For each ground-truth interval y^*_{in}, let its highest IoU with any predicted interval be

$$b_{in} = \max_{\widehat{y}\in\widehat{\mathcal{Y}}_i}\mathrm{IoU}(\widehat{y}, y^*_{in}).$$

We set $b_{in} = 0$ when the prediction set is empty. The mIoU and Recall@0.5 for query i are then

$$\mathrm{mIoU}_i = \frac{1}{N_i}\sum_{n=1}^{N_i} b_{in}, \qquad \text{Recall@0.5}_i = \frac{1}{N_i}\sum_{n=1}^{N_i}\mathbf{1}[b_{in} \ge 0.5].$$

F1@0.5 uses one-to-one matching. Predicted intervals are processed in order, with each matched to the first unmatched ground-truth interval whose IoU is at least 0.5. If TP_i is the number of successful matches,

$$\text{F1@0.5}_i = \frac{2\mathrm{TP}_i}{\widehat{N}_i + N_i}.$$

For negative queries, correct rejection requires an empty predicted interval set. We denote the index set of queries with no ground-truth events by $\mathcal{D}_- = \{i : N_i = 0\}$. The correct rejection rate r_- and positive-query coverage are defined as

$$r_- = \frac{1}{|\mathcal{D}_-|}\sum_{i\in\mathcal{D}_-}\mathbf{1}[\widehat{N}_i = 0], \qquad \mathrm{PosCoverage} = \frac{1}{|\mathcal{D}_+|}\sum_{i\in\mathcal{D}_+}\mathbf{1}[\widehat{N}_i > 0].$$

The false positive rate is $\mathrm{FPR} = 1 - r_-$. Rejection-F1 jointly measures correct rejection and positive-query coverage:

$$\mathrm{RejF1} = 100 \cdot \frac{2r_-\,\mathrm{PosCoverage}}{r_- + \mathrm{PosCoverage}}.$$

This metric is set to zero when the denominator is zero. Following the official implementation, r_- is rounded to four decimal places before computing Rejection-F1. Rejection-F1 is reported on a 0–100 scale, while the other proportion-based metrics use a 0–1 scale.

Visual token cost. For query i, let C_i denote the number of model calls. In call c, the model actually receives $T^{\mathrm{img}}_{i,c}$ image tokens and $T^{\mathrm{vid}}_{i,c}$ video tokens. The average cumulative visual cost is defined as

$$\overline{T}^m = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C_i}T^m_{i,c}, \quad m \in \{\mathrm{img}, \mathrm{vid}\}, \qquad \overline{T}^{\mathrm{vis}} = \overline{T}^{\mathrm{img}} + \overline{T}^{\mathrm{vid}}.$$

When an image or video re-enters a later request as part of the interaction history, its visual tokens are counted again for that request. In the main experiments, token counts are computed from the actual resizing and sampling parameters of each request, following the grid used by the Qwen3.5-27B visual processor, rather than simply counting media files. All queries, including those without a valid final prediction, contribute to the denominator of the average. Media that are generated but never passed to the model incur no visual token cost.

Interaction statistics are likewise averaged per query, with separate counts recorded for model calls, local tool calls, and image and video observations. The total number of visual observations is the sum of the image and video observation counts.

B.2.3 Inference Settings

The base and evolved skills remain fixed throughout evaluation. For a given VLM, both use the same set of queries, task prompts, decoding configuration, interaction budget, and scorer. Policy documents are incorporated into the model's system context, and tools are registered as callable functions.

Table 8 summarizes the model configuration for the main Qwen3.5-27B experiments. Task execution during evolution uses the same VLM configuration. Image and video observations are not subject to separate call limits but share an interaction budget of at most 24 inference rounds per query.

Table 8: **Model configuration for Qwen3.5-27B.** The base and evolved skills use identical runtime settings.

Parameter	Setting
Serving Engine	vLLM
Numerical Precision	bfloat16
Maximum Context Length	262,144 tokens
Thinking Mode	Enabled
Decoding	Greedy
Temperature / Top-p / Top-k	0 / 1.0 / 0
Maximum Generated Tokens per Call	4,096

All experiments use visual inputs only, without audio, speech recognition, or transcripts. Predictions for VUE-LVTR and CoMET-Bench are sets of source-video time intervals, whereas ExtremeWhenBench requires a single interval. When transferring to LVBench (Wang et al., 2025a) and LSDBench (Qu et al., 2025), the model continues to search for evidence using the policies and tools evolved for grounding. Task adaptation only supplies the QA instructions and answer options while changing the final output to a single option letter, and the skill is not updated on QA data.

B.2.4 Baseline Evaluation Protocols

We evaluate seven VLM-based and agent-based baselines on the three ultra-long video temporal grounding benchmarks. Among open-source VLMs, Qwen3.5-27B (Qwen Team, 2026a) receives video input with 768 frames uniformly sampled across the full duration, InternVL3.5-8B (Wang et al., 2025b) uses a 128-frame image sequence, and TimeLens-7B (Zhang et al., 2026) uses 384 video frames with textual timestamps. For the closed-source models Gemini 2.5 Flash (Comanici et al., 2025) and GPT-5.6 Luna (OpenAI, 2026b), the input consists of 128 uniformly sampled frames arranged chronologically in timestamped image grids. Agent-based baselines include VideoMind-7B (Liu et al., 2026), in which grounding and verification roles collaborate for multi-step video reasoning, and EvoGround-7B (Jung et al., 2026), which uses feedback between its Proposer and Solver for self-evolving training to improve temporal grounding.

C More Results

C.1 Results on the Evolution Set

Table 9 reports the results of applying the fixed base and evolved skills to the 100 evolution queries after evolution has completed. All reported metrics improve across the three VLMs, with IoU AUC gains of **0.0882** for Qwen3.5-9B, **0.0651** for Qwen3.5-27B, and **0.0523** for Qwen3.6-27B. These results characterize how well each skill fits the evolution data.

C.2 Visual Token Cost on Qwen3.6-27B

Table 10 reports visual token cost for Qwen3.6-27B. After evolution, the average cumulative visual token cost decreases by **42.2%** on the VUE-LVTR held-out set, **28.1%** on ExtremeWhenBench, and **15.8%** on CoMET-Bench. On CoMET-Bench, image tokens increase from 83.60k to 100.19k

Table 9: **Grounding gains from policy–tool coevolution on the evolution set.** Results are obtained by evaluating the fixed skills after evolution. **Bold**: better results; blue: gains over the base skill. Evolution Gain reports absolute changes, with arrows indicating relative changes.

VLM	Method	Precision AUC ↑	Recall AUC ↑	IoU AUC ↑	IoU@0.3 ↑	IoU@0.5 ↑	IoU@0.7 ↑
Qwen3.5-9B	+ Base Skill	0.1759	0.1961	0.1142	0.1600	0.1300	0.0500
	+ Evolved Skill	**0.3065**	**0.3545**	**0.2024**	**0.3000**	**0.2000**	**0.1100**
	Evolution Gain	**+0.1306** ↑74.2%	**+0.1584** ↑80.8%	**+0.0882** ↑77.2%	**+0.1400** ↑87.5%	**+0.0700** ↑53.8%	**+0.0600** ↑120.0%
Qwen3.5-27B	+ Base Skill	0.3620	0.3047	0.2556	0.3400	0.2800	0.1700
	+ Evolved Skill	**0.4285**	**0.3916**	**0.3207**	**0.4300**	**0.3500**	**0.2400**
	Evolution Gain	**+0.0665** ↑18.4%	**+0.0869** ↑28.5%	**+0.0651** ↑25.5%	**+0.0900** ↑26.5%	**+0.0700** ↑25.0%	**+0.0700** ↑41.2%
Qwen3.6-27B	+ Base Skill	0.3341	0.3307	0.2459	0.3600	0.2800	0.1400
	+ Evolved Skill	**0.3732**	**0.4137**	**0.2982**	**0.4100**	**0.3300**	**0.2200**
	Evolution Gain	**+0.0391** ↑11.7%	**+0.0830** ↑25.1%	**+0.0523** ↑21.3%	**+0.0500** ↑13.9%	**+0.0500** ↑17.9%	**+0.0800** ↑57.1%

while video tokens decrease from 55.36k to 16.82k. The reduction in total cost is thus accompanied by a reallocation of the visual budget between the two observation modalities.

Table 10: **Visual token cost of the base and evolved skills on Qwen3.6-27B. Bold**: lower cost. Green (↓) and purple (↑) indicate decreases and increases relative to the base skill. Δ denotes evolved cost minus base cost, with arrows indicating relative changes. Tokens: k/query.

Benchmark	Method	Visual Tokens ↓	Image Tokens	Video Tokens
VUE-LVTR held-out	Base Skill	241.21	214.98	26.23
	Evolved Skill	**139.36**	**134.10**	**5.26**
	Δ (Evolved − Base)	**−101.85** ↓42.2%	**−80.88** ↓37.6%	**−20.97** ↓79.9%
ExtremeWhenBench	Base Skill	220.29	195.24	25.05
	Evolved Skill	**158.33**	**152.64**	**5.69**
	Δ (Evolved − Base)	**−61.96** ↓28.1%	**−42.60** ↓21.8%	**−19.36** ↓77.3%
CoMET-Bench	Base Skill	138.96	**83.60**	55.36
	Evolved Skill	**117.01**	100.19	**16.82**
	Δ (Evolved − Base)	**−21.95** ↓15.8%	+16.59 ↑19.8%	**−38.54** ↓69.6%

C.3 Results Across Multiple Dimensions

ExtremeWhenBench. Table 11 reports mIoU by query category. Qwen3.5-9B and Qwen3.5-27B improve across all five categories, while Qwen3.6-27B improves on Action, Object, Reaction, and Scene, with a slight decrease on Environment. This variation across categories indicates that the distribution of evolution gains depends on both the model and the query type.

CoMET-Bench. We examine the evolution gains in terms of event counting, rejection behavior on target-absent queries, and grounding performance across different event counts. Table 12 shows that counting MAE decreases and both OBO and Pearson correlation increase for all three VLMs, indicating that the gains also extend to the accuracy of event counting.

Table 13 further reports Rejection-F1, the false positive rate on negative queries, and positive-query coverage. All three VLMs achieve higher Rejection-F1 and positive-query coverage after evolution. For Qwen3.5-9B and Qwen3.5-27B, these gains are accompanied by a lower false positive rate.

Grouped by the ground-truth event count, the results in Table 14 reveal differences in multi-event grounding. Qwen3.5-9B and Qwen3.5-27B improve across all groups, while Qwen3.6-27B improves in four of the five groups. Queries with more events still exhibit lower F1@0.5, indicating that precisely detecting and localizing multiple target events in long videos remains challenging.

Table 11: **Grounding results by query category on ExtremeWhenBench.** Each column reports mIoU (↑) for the corresponding category. **Bold**: better results; blue: gains over the base skill. Evolution Gain reports absolute changes, with arrows indicating relative changes.

VLM	Method	Action	Environment	Object	Reaction	Scene
Qwen3.5-9B	+ Base Skill	0.0543	0.0986	0.0877	0.0439	0.0570
	+ Evolved Skill	**0.1092**	**0.1485**	**0.1518**	**0.1218**	**0.1246**
	Evolution Gain	**+0.0549** ↑101.1%	**+0.0499** ↑50.6%	**+0.0641** ↑73.1%	**+0.0779** ↑177.4%	**+0.0676** ↑118.6%
Qwen3.5-27B	+ Base Skill	0.1446	0.2040	0.1862	0.1167	0.1377
	+ Evolved Skill	**0.2459**	**0.2884**	**0.3427**	**0.1872**	**0.2664**
	Evolution Gain	**+0.1013** ↑70.1%	**+0.0844** ↑41.4%	**+0.1565** ↑84.0%	**+0.0705** ↑60.4%	**+0.1287** ↑93.5%
Qwen3.6-27B	+ Base Skill	0.1565	**0.2162**	0.2289	0.1259	0.1583
	+ Evolved Skill	**0.2122**	0.2004	**0.2797**	**0.1681**	**0.1975**
	Evolution Gain	**+0.0557** ↑35.6%	−0.0158 ↓7.3%	**+0.0508** ↑22.2%	**+0.0422** ↑33.5%	**+0.0392** ↑24.8%

Table 12: **Event counting results on CoMET-Bench. Bold**: better results; blue: gains over the base skill. Evolution Gain reports absolute changes, with arrows indicating relative changes.

VLM	Method	MAE ↓	OBO ↑	Pearson ↑
Qwen3.5-9B	+ Base Skill	3.7104	0.5378	0.3305
	+ Evolved Skill	**3.6929**	**0.5516**	**0.6614**
	Evolution Gain	**−0.0175** ↓0.5%	**+0.0138** ↑2.6%	**+0.3309** ↑100.1%
Qwen3.5-27B	+ Base Skill	3.6717	0.5760	0.0775
	+ Evolved Skill	**3.4290**	**0.6116**	**0.1258**
	Evolution Gain	**−0.2427** ↓6.6%	**+0.0356** ↑6.2%	**+0.0483** ↑62.3%
Qwen3.6-27B	+ Base Skill	3.6473	0.5716	0.0864
	+ Evolved Skill	**3.5616**	**0.5816**	**0.1214**
	Evolution Gain	**−0.0857** ↓2.3%	**+0.0100** ↑1.7%	**+0.0350** ↑40.5%

Table 13: **Rejection and positive-query coverage results on CoMET-Bench.** Rejection-F1 is reported on a 0–100 scale, while FPR and PosCoverage use a 0–1 scale. **Bold**: better results; blue: gains over the base skill. Evolution Gain reports absolute changes, with arrows indicating relative changes.

VLM	Method	Rejection-F1 ↑	FPR ↓	PosCoverage ↑
Qwen3.5-9B	+ Base Skill	67.51	0.0989	0.5397
	+ Evolved Skill	**68.80**	**0.0968**	**0.5556**
	Evolution Gain	**+1.29** ↑1.9%	**−0.0021** ↓2.1%	**+0.0159** ↑2.9%
Qwen3.5-27B	+ Base Skill	66.13	0.1183	0.5291
	+ Evolved Skill	**77.06**	**0.1032**	**0.6755**
	Evolution Gain	**+10.93** ↑16.5%	**−0.0151** ↓12.8%	**+0.1464** ↑27.7%
Qwen3.6-27B	+ Base Skill	68.53	**0.0989**	0.5529
	+ Evolved Skill	**72.37**	0.1054	**0.6076**
	Evolution Gain	**+3.84** ↑5.6%	+0.0065 ↑6.6%	**+0.0547** ↑9.9%

Table 14: **Grounding results by target-event count on CoMET-Bench.** Groups are defined by the ground-truth event count, and each column reports macro-averaged F1@0.5 over the corresponding subset of positive queries. **Bold**: better results; blue: gains over the base skill. Evolution Gain reports absolute changes, with arrows indicating relative changes.

VLM	Method	1 event ↑	2–3 ↑	4–7 ↑	8–15 ↑	≥16 ↑
Qwen3.5-9B	+ Base Skill	0.1055	0.0673	0.0460	0.0376	0.0124
	+ Evolved Skill	**0.1314**	**0.1296**	**0.0574**	**0.0622**	**0.0239**
	Evolution Gain	**+0.0259** ↑24.5%	**+0.0623** ↑92.6%	**+0.0114** ↑24.8%	**+0.0246** ↑65.4%	**+0.0115** ↑92.7%
Qwen3.5-27B	+ Base Skill	0.2058	0.1846	0.1257	0.0860	0.0226
	+ Evolved Skill	**0.2615**	**0.2101**	**0.1498**	**0.0939**	**0.0529**
	Evolution Gain	**+0.0557** ↑27.1%	**+0.0255** ↑13.8%	**+0.0241** ↑19.2%	**+0.0079** ↑9.2%	**+0.0303** ↑134.1%
Qwen3.6-27B	+ Base Skill	0.2085	0.1597	0.1342	**0.0990**	0.0322
	+ Evolved Skill	**0.2578**	**0.1800**	**0.1378**	0.0901	**0.0377**
	Evolution Gain	**+0.0493** ↑23.6%	**+0.0203** ↑12.7%	**+0.0036** ↑2.7%	−0.0089 ↓9.0%	**+0.0055** ↑17.1%

D CASE STUDY

We analyze the execution trajectories of the base and evolved skills on two ExtremeWhenBench (Seo & Kim, 2026) queries using the same Qwen3.5-27B. Figures 6 and 7 illustrate how policy–tool coevolution changes evidence acquisition for localizing an action and a scene, respectively. Both skills remain fixed during evaluation, and visual token costs are computed following the definitions in Appendix B.2.2.

In Figure 6, the query asks when fries are scooped with a metal skimmer into a holding trough in a 50.6-minute video. The base skill selects a candidate near 900 s from 15 sparse frames and inspects four overlapping clips in this region. Although the initial clips do not confirm the queried action, the model continues to inspect the same region and ultimately mistakes a chute transfer for the requested action. In contrast, the evolved skill uses timestamped contact sheets for global comparison, followed by dense frame inspection over 550–590 s and a single 15-second clip for verification. The prediction matches the ground-truth interval $[570, 582]$ s, raising IoU from 0 to 1. Meanwhile, video tokens drop from 168.98k to 11.88k, leading to a **51.27%** net reduction in visual token cost. This trajectory illustrates how better candidate localization allows video observation to focus on verifying query-specific motion, reducing repeated inspection of an incorrect region.

Figure 7 concerns an 11-second scene of a couple walking along a tree-lined path in golden light within a 100.1-minute video. The base skill performs six batches of sparse sampling across the long timeline but misses the brief target scene. Subsequent frame and video inspection focuses on a visually similar candidate, yielding the incorrect interval $[1868, 1915]$ s. The evolved skill instead packs 192 frames into eight contact sheets, allowing a broad search within one image observation. After identifying a candidate near 5708 s, it uses a local contact sheet over 5680–5740 s and targeted boundary frames to determine when the scene begins and ends. These image-based observations localize the target to $[5703, 5714]$ s without video observation, increasing IoU from 0 to 1 while reducing visual token cost by **45.78%**. This case highlights the importance of discovering a brief target during global search before investing in local boundary refinement.

Together, these two cases illustrate the connection between tool capabilities and observation orchestration: the evolved tools provide compact temporal coverage and explicit temporal correspondence, while the policies organize subsequent observations around the evidence still needed. In the base trajectories, the additional local evidence gathered remains confined to an incorrect candidate. The evolved trajectories instead improve candidate localization before resolving the remaining temporal uncertainty, using video to verify action progression in one case and images to determine scene boundaries in the other. This contrast illustrates how coordinated image–video observation adapts to the evidence needs of different queries. By coupling the form in which evidence is presented with decisions about where to search and what to verify, the evolved skill enables the VLM to achieve more precise grounding at a lower visual token cost.

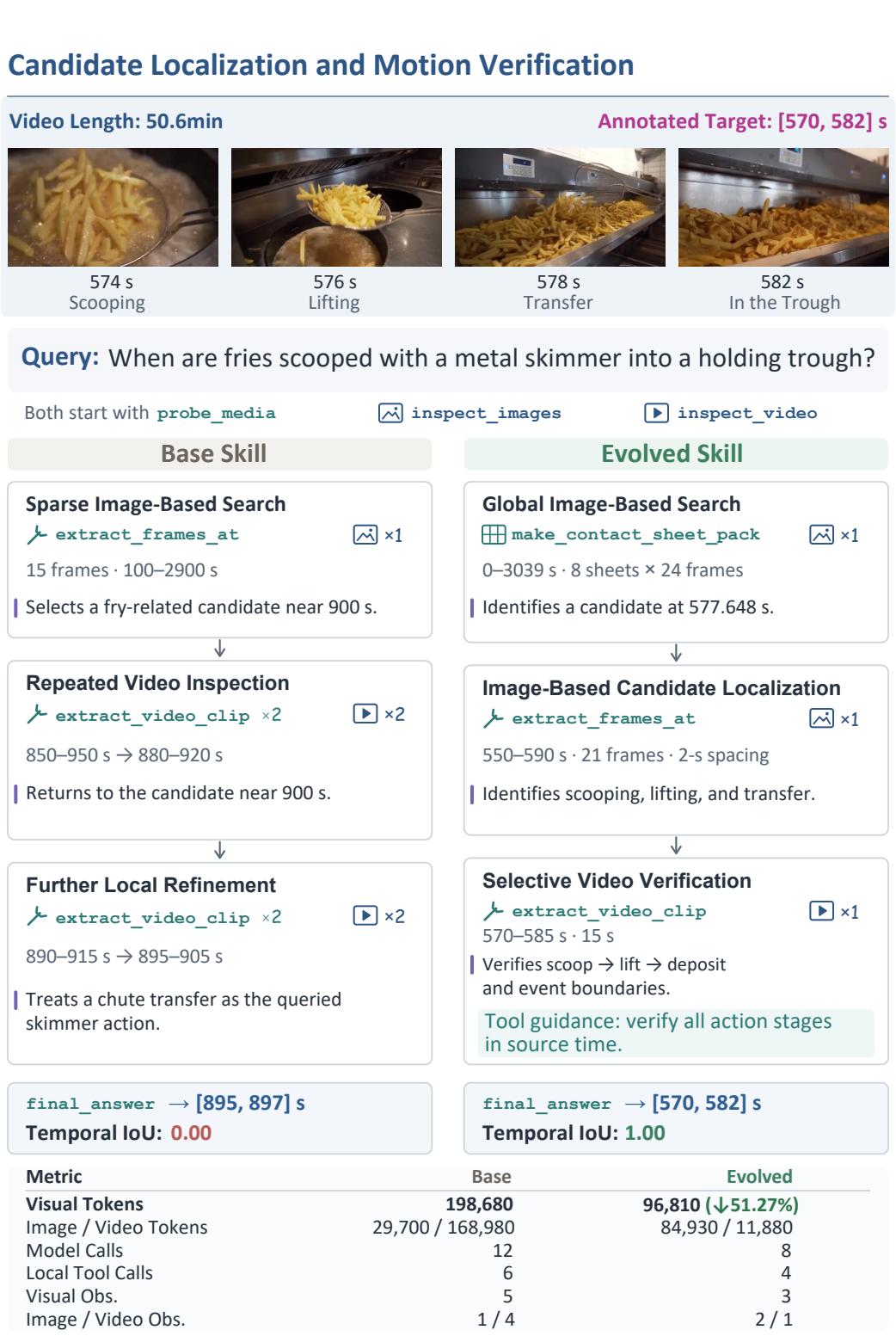

Figure 6: **Candidate localization and motion verification.** The base skill repeatedly inspects an incorrect candidate, whereas the evolved skill combines global contact-sheet search with local frame inspection and a single short video to verify the complete action, precisely localizing the target interval while reducing visual token cost by **51.27%**.

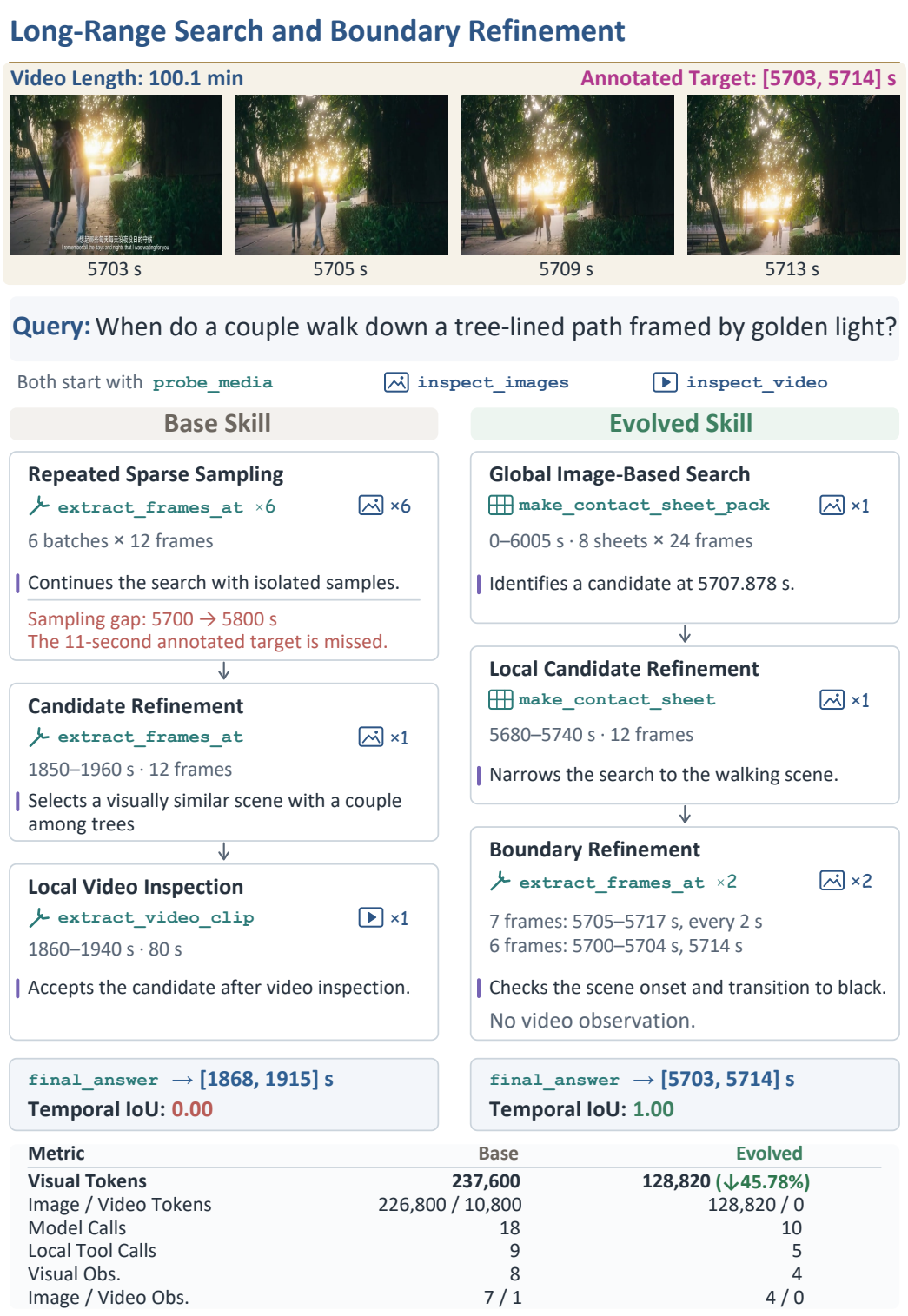

Figure 7: **Long-range search and boundary refinement.** Sparse sampling misses the brief target, leading the base skill to mistakenly focus on a visually similar scene. The evolved skill instead combines global and local contact sheets with boundary frames to precisely localize the target interval, reducing visual token cost by **45.78%**.

E LIMITATIONS AND FUTURE WORK

At present, our policy–tool coevolution is conducted on a limited set of temporal grounding queries from a single data source with labeled feedback. Although the evolved skill improves performance across ultra-long video temporal grounding and long-video QA tasks, broader data sources, larger evolution scales, and evolution without labeled feedback remain to be further explored. All visual observations in the current framework are performed directly by the VLM acting as the main agent. A meaningful next step is to extend coevolution to determine when the main agent should observe directly or delegate observation to subagents, and how they can coordinate efficiently. Since we currently focus only on the visual modality, extending the framework to more general omni-modal settings also merits investigation.

F PROMPTS

F.1 SKILL PROMPTS

System Prompt

You are a video-task solving agent.

Active evolvable skill:
<SKILL.md, tool-selection.md, tool-notes.md, policy-notes.md>

Current task adapter:
<optional task-specific adapter>

Base SKILL.md

This skill supports ultra-long video temporal grounding with image and video observations.

Evidence acquisition

1. Use local tools to read video metadata and prepare visual evidence. `probe_media` reads metadata, `extract_video_clip` prepares video clips, and `extract_frames_at` prepares timestamped frames.
2. Choose the observation tool for the current evidence need:
   - `inspect_video` for native video observation of motion, event order, state changes, fine details, or temporal boundaries.
   - `inspect_images` for native image observation of static detail, readable text, object identity, or boundary frames.
3. Use source-video seconds in the final temporal answer.

Fixed constraints

- Local tools prepare visual evidence and metadata. The VLM interprets the evidence and predicts temporal intervals.
- Do not call external models or APIs from local tools.
- Do not use audio, speech recognition, transcripts, hidden labels, filenames, or ground truth as evidence.
- Submit the final prediction with `final_answer` only after collecting sufficient visual evidence.

No observation orchestration policy has been learned through evolution yet.

F.2 Evaluation Task Prompts

VUE-LVTR Prompt

Task: <text query>
Required answer format: a JSON array containing all matching [start_seconds, end_seconds] intervals.
Declared video duration: <duration> seconds.
Use source-video seconds for temporal answers.

ExtremeWhenBench Prompt

You are watching a video of <duration> seconds. Please find the moment described by the following question, determining its starting and ending times.
Question: <question>
Required answer format: a single [start_seconds, end_seconds] interval in source-video seconds.

CoMET-Bench Prompt

Watch the provided video and answer the following question:
<query>
Return ONLY a JSON array of temporal intervals where the event occurs, each formatted as [start_seconds, end_seconds]. Return [] if the event does not occur. No explanation.

Long-Video QA Prompt

The current task is multiple-choice video question answering. Reuse the active skill to locate and verify the relevant visual evidence. Any temporal-interval final-output convention in the skill does not apply to this task. Finish by returning exactly one option letter: A, B, C, or D.
<question>
<answer options>

F.3 CODEX UPDATER PROMPT

Codex Updater Prompt

You are optimizing a compact reusable skill for ultra-long video temporal grounding.

Inputs

- Update context with trajectory summaries, task feedback, and evaluation metrics: <update context path>
- Execution trajectories in the current batch:
  - Episode <episode index>: <trajectory path>
  - ...
- Batch feedback and aggregate metrics: <batch feedback path>
- Previous skill (read-only): <previous skill directory>
- Candidate skill (a copy of the previous skill to update in place): <candidate skill directory>

Evolution rules

- Edit only the candidate skill. Keep the previous skill, trajectories, and feedback unchanged. You may edit `SKILL.md`, `tools/`, and the three reference documents listed below. All editable paths are relative to the candidate skill directory.
- First analyze all execution trajectories in the batch: execution steps, tool choices, visual observations, model–tool interactions, common failures, and successful patterns.
- Use the update context for evaluation metrics and final outcomes. Do not copy sample identifiers, video or file names, question text, scene facts, predictions, ground-truth intervals, dataset-specific labels or groupings, dataset-specific patterns, or content-type examples into the skill. Express reusable lessons as task-general evidence workflows, tool contracts, validation checks, or guidance for model–tool interactions.
- Keep local tools deterministic and local. They prepare visual evidence and metadata for the VLM, which interprets the evidence and predicts temporal intervals. Do not call external models or APIs, or use audio, speech recognition, or transcripts.
- Record reusable lessons in the appropriate skill files:
  - `references/policy-notes.md`: task planning, evidence acquisition, and reasoning strategies.
  - `references/tool-notes.md`: tool contracts, failure modes, and usage details.
  - `references/tool-selection.md`: selection of local tools and observation modes.

  Consider the trade-offs between native image and video observation, including static detail, readable text, object identity, and boundary-frame verification with `inspect_images`.
- Ground updates in the listed execution trajectories: tool choices, visual readability, temporal coverage, interaction failures, token usage, and repeated patterns where a better reusable tool or tool-selection rule would help.
- Analyze token usage as part of skill quality, not just task score. Prefer updates that improve accuracy while keeping token use stable or lower. Add broad video inspection, longer clips, or extra observation calls only when trajectories support a reusable accuracy benefit that justifies the cost.
- Use code synthesis for reusable improvements. Add or improve local media-preparation tools for clipping, frame sampling, resizing, timestamp labeling, compact visual summaries, motion- or scene-aware media preparation, and validation. Merge or delete weak or redundant tools. Store each tool's code and interface in `tools/<tool_name>/tool.py` and `tools/<tool_name>/tool.json`.
- Consolidate or replace weak guidance. Leave the candidate unchanged if there is no reusable improvement.