VIDEO AS EXPERIENCE: WATCH, ACT, AND REWATCH FOR COMPUTER-USE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models have enabled increasingly capable computer-use agents that perceive graphical user interfaces, reason over instructions, and execute actions in interactive environments. However, existing agents largely operate on information available within the current interaction, while procedural knowledge from demonstrations is typically incorporated through separately extracted action trajectories. We introduce VEX (Video as EXperience), a framework that treats video as a unified representation of experience for both learning from external demonstrations and reflecting on an agent's own behavior. VEX follows a simple watch-act-rewatch paradigm: an agent extracts task-relevant guidance from tutorial videos, acts in the target environment, and subsequently verifies its progress by rewatching its execution as video. We instantiate VEX with video-SALMONN-VEX, an 8B-scale audiovisual computer-use agent built on video-SALMONN 2+, and introduce Action-Grounded Guide Optimization (AGGO), a reinforcement learning method that trains the model to generate procedural guidance according to its utility for downstream action prediction. We further augment AgentNet, OSWorld, and CUA-Gym with more than 24K real-world tutorial videos to support learning and evaluation from video experience. Experiments on OSWorld and a complementary subset of CUA-Gym show that VEX consistently improves computer-use performance, with video-SALMONN-VEX achieving a 17% relative improvement in task success rate over its backbone on OSWorld. Moreover, guidance generated by video-SALMONN-VEX improves Gemini-3.5-flash by 10% relatively on OSWorld. These results demonstrate the potential of video as a common experiential interface for acquiring procedural knowledge and verifying interactive behavior in computer-use agents.

1 INTRODUCTION

Recent advances in multimodal large language models (LLMs) (Li et al., 2024; Zhang et al., 2024; Lin et al., 2024; Bai et al., 2025b; Zhang et al., 2025a; Tang et al., 2025; Sun et al., 2025; Bai et al., 2025a; Sun et al., 2026) have enabled increasingly capable agents for computer use, where models perceive graphical user interfaces (GUIs), reason over task instructions, and execute actions in interactive environments (Koh et al., 2024; He et al., 2024; Cheng et al., 2024; Rawles et al., 2025). Most existing computer-use agents (CUAs), however, are primarily optimized to act from the information available in the current interaction, such as screenshots, textual instructions, and previously executed actions (Agashe et al., 2025; Zhang et al., 2025b; Gou et al., 2025; Xu et al., 2025; Qin et al., 2025; Yang et al., 2026a; Fu et al., 2025; Li et al., 2025; Xue et al., 2026; Wang et al., 2025). In contrast, humans often acquire new procedural skills by observing demonstrations from experienced individuals. To this end, online tutorial videos provide an abundant record of such demonstrations, jointly capturing visual states, user operations, and spoken explanations.

This has motivated a growing line of work that leverages instructional videos to improve computer-use agents, either at training time or as additional knowledge at inference time. Recent approaches (Song et al., 2026; Lu et al., 2026; Xiong et al., 2026; Liu et al., 2026) have demonstrated that internet-scale videos can indeed provide valuable information for computer use, typically by recovering executable interaction trajectories from observed demonstrations and using the resulting trajectories for training or in-context guidance.

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.

Despite these advances, existing approaches largely use video as a source of trajectory extraction, rather than as a form of experience that an agent can directly understand and reason over. Raw videos are commonly passed through dedicated processing pipelines (Liu et al., 2026) that construct and filter trajectories before they can be consumed by the main agent. These representations are effective and provide explicit access to actions and intermediate states, but constructing them often requires dedicated processing pipelines and additional models. More fundamentally, demonstrations and an agent's own interactions are often represented differently, with external experience originating as video while self-generated experience is recorded as interleaved screenshots, actions, and textual traces. A single audiovisual model could instead use the same perceptual interface to extract task-relevant knowledge from observed demonstrations and revisit its own execution, unifying learning from others and reflecting on its own behavior as reasoning over videos.

To achieve such agents, we introduce VEX (Video as EXperience), a framework for computer-use agents that use video as a common representation of experience for both acquisition of procedural knowledge and self-verification on task trajectories. At inference time, VEX follows a simple watch-act-rewatch loop, where the agent uses tutorial-derived guidance when needed and verifies its progress by rewatching its own execution as video. We instantiate VEX with video-SALMONN-VEX, a single 8B-scale agent built on video-SALMONN 2+ that can understand audio-visual inputs while performing computer-use tasks. In addition, we introduce Action-Grounded Guide Optimization (AGGO) to train video-SALMONN-VEX to extract procedural guidance from tutorial videos according to their utility for downstream action prediction.

The VEX framework is evaluated on both OSWorld (Xie et al., 2024) and a complementary subset of CUA-Gym (Wang et al., 2026) focusing on tasks that are not covered by OSWorld. Consistent improvements were obtained using VEX for video-SALMONN-VEX compared to its backbone Qwen3-VL and video-SALMONN 2+ models, with a notable 17% relative task success rate improvement on OSWorld. Moreover, using the VEX framework together with task guidance generated from video-SALMONN-VEX on Gemini-3.5-flash, a relative 10% improvement was achieved on OSWorld. The main contributions of this paper are summarized as follows:

- We propose VEX, a light-weight single-agent framework using video as the unified representation of experience for computer use tasks. We introduce video-SALMONN-VEX as the single agent with audio-visual understanding capabilities that can learn from real video tutorials to perform the task, and can verify its performance through video recordings of the trajectory.
- We introduce AGGO, a reinforcement learning algorithm that trains the agent to extract task-useful guides from tutorial videos by rewarding guides according to their utility for predicting ground-truth action trajectories.
- We augment AgentNet, OSWorld, and CUA-Gym with over 25K real tutorial videos, and evaluate VEX on OSWorld and a complementary subset of CUA-Gym, where video-SALMONN-VEX achieves a **17%** relative improvement in task success rate over its backbone.

2 Related Work

2.1 Computer-Use Agents

Recent progress in multimodal LLMs has driven rapid advances in agents that interact with computers through graphical user interfaces. Early work studied web navigation and grounding from demonstrations and structured webpages, including WebShop, Mind2Web, and WebArena (Yao et al., 2022; Deng et al., 2023; Zhou et al., 2024), while subsequent benchmarks such as VisualWebArena, WebVoyager, AndroidWorld, Windows Agent Arena, and OSWorld extended evaluation to visually grounded websites, mobile devices, and open-ended desktop environments (Koh et al., 2024; He et al., 2024; Rawles et al., 2025; Xie et al., 2024). In parallel, GUI-specialized multimodal models and agent frameworks have improved visual grounding, action prediction, and long-horizon interaction, including CogAgent (Hong et al., 2024), SeeClick (Cheng et al., 2024), OS-ATLAS (Wu et al., 2025), ShowUI (Lin et al., 2025), Agent S (Agashe et al., 2025), UFO (Zhang et al., 2025b), Aguvis (Xu et al., 2025), UI-TARS (Qin et al., 2025), GTA-1 (Yang et al., 2026a), MANO (Fu et al., 2025), DART-GUI (Li et al., 2025), EvoCUA (Xue et al., 2026), and OpenCUA (Wang et al., 2025). These approaches primarily improve computer use through GUI-specific training data,

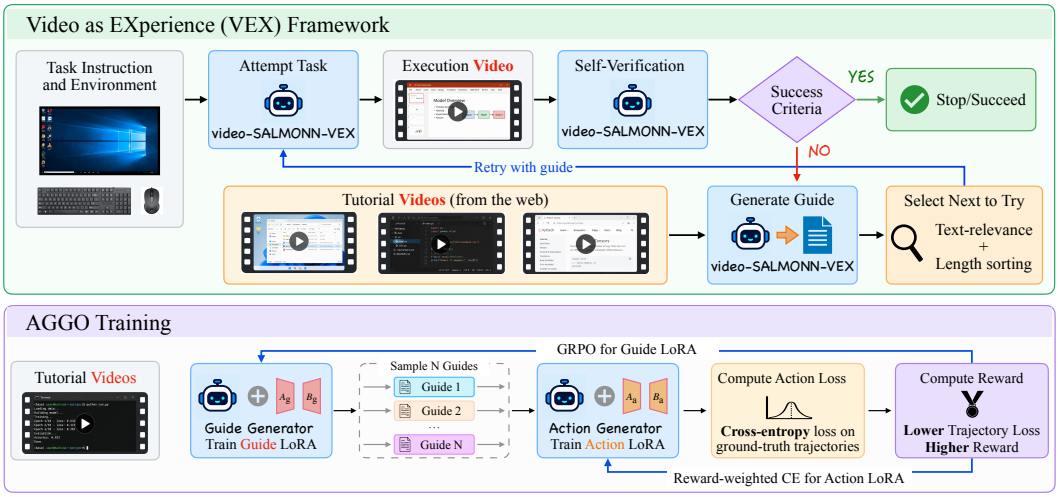


Figure 1: Illustration of the **VEX framework** (upper): We highlight interfaces that employ video as the representation. The VEX framework starts with task execution without video first, followed by the self-verification based on a video recording of its execution. If the execution failed to meet criteria, the agent will fetch the next most informative video tutorial, extract guidance and re-perform the task. The loop is repeated until success or the maximum trial is reached. Illustration of **AGGO** (lower): the guide generation policy is trained with GRPO and the action generation policy is trained with weighted CE loss, as described in Section 3.2.

interaction traces, grounding objectives, planning, or reasoning over the current observation and interaction history. VEX complements these by focusing on how procedural experience and an agent's past execution can be incorporated through a shared video interface.

2.2 Learning from Video Experience

Recent work moved beyond passive video understanding toward acquiring reusable knowledge and skills from prior video experience. SLVMBench (Yang et al., 2026b) studies this setting explicitly by evaluating whether models can extract procedural knowledge from tutorial videos retained in long-form video memory and later apply that knowledge to ongoing tasks, while video-SALMONN S (Sun et al., 2026) provides a complementary architectural perspective by enabling selective retention and retrieval from multi-hour audiovisual streams. In computer use specifically, instructional videos have emerged as a scalable source of procedural experience: Watch & Learn (Song et al., 2026) recovers executable interaction trajectories from online demonstrations for training and in-context guidance, while related approaches convert, segment, retrieve, or filter tutorial videos into action trajectories or procedural guides for downstream GUI interaction (Lu et al., 2026; Xiong et al., 2026; Liu et al., 2026; Zhang et al., 2026). These studies demonstrate that videos contain valuable task knowledge, but generally transform demonstrations into an intermediate representation before the acting agent consumes them. VEX instead treats video itself as the common representation of experience: the same audio-visual agent can watch external demonstrations to acquire task-relevant procedural knowledge and rewatch recordings of its own execution for verification.

3 Methodology

3.1 VEX Framework Overview

The illustration of the VEX framework is shown in Fig. 1. The core idea is to use a single agent, such as video-SALMONN-VEX, to learn from video, to perform actions, and to verify its action trajectory represented as a video. A loop is formed when the verification result does not meet the pass criterion, enabling a second trial with more knowledge from video.

Specifically, starting with the task instruction and environment, the agent performs the task without any further inputs for the first time, and the trajectory is recorded as videos with action descriptions converted to audio via text-to-speech (TTS) synthesis. After re-watching its own actions, a self-verification score is provided by the agent itself (see Section 3.3 for detailed implementations). Then, a threshold check is performed, and when the execution is below the threshold, the agent will do the task again with the help of external video tutorials.

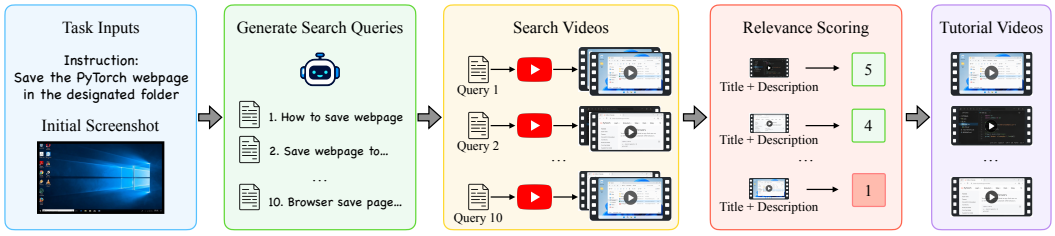


Figure 2: Illustration of the search pipeline to get the candidate pool of tutorial videos. The agent used in this pipeline is the same agent used in the VEX framework. The scoring is based on title and description, with a relatively coarse 1 to 5 score scale.

To learn from video tutorials, a pool of candidate videos is first obtained following the pipeline shown in Fig. 2, which is constructed with the help of the same agent itself. Then, guidance texts are extracted from each video by providing the task description and the initial screenshot to the agent. The agent is prompted to extract as much information as possible, while being honest to respond `no information found` when there is nothing to extract. The detailed prompt design is provided in the Appendix. A. Then, when the first trial fails the threshold, the video with the longest guidance is selected, and the text guidance extracted by the agent from the video is used as the context for the agent to conduct the task. The agent then goes through the act-and-rewatch procedure again, and once failed, the second-longest video is used to provide tutorial knowledge. If the set maximum number of trials is reached and there is still no trajectories above the threshold, the one with the highest self-verification score is used as the final execution of the task.

3.2 VIDEO-SALMONN-VEX AND AGGO TRAINING

We introduce video-SALMONN-VEX, which is an audio-visual LLM capable of understanding both audio and visual inputs in videos. The auditory capability is particularly important since most instructional information is conveyed via speech in demonstrative videos. The model structure inherits from video-SALMONN 2+ (Tang et al., 2025), and we subsequently fine-tune its LLM backbone with low-rank adapters (LoRA) (Hu et al., 2022) via the AGGO algorithm.

The process of AGGO is illustrated in Fig. 1, which leverages data containing ground-truth trajectories to train the guidance from videos via RL. Specifically, we train the model to extract task-relevant textual guidance from tutorial videos and use the generated guidance to improve downstream computer-use action prediction. Each training example consists of a task instruction q, an instructional video V, and an oracle computer-use trajectory $\tau^* = \{(o_1, a_1^*), \ldots, (o_T, a_T^*)\}$, where o_t and a_t^* denote the observation and groundtruth action at step t, respectively. Rather than supervising the generated guidance with a reference, we directly evaluate its usefulness based on how well it supports prediction of the ground-truth trajectory.

Two separate LoRA adapters are used, denoted as ϕ_{g} for the guidance generation policy and ϕ_{a} for the action prediction policy, since they require different capabilities. The action policy is first initialized from supervised fine-tuning (SFT) using ground-truth trajectories without guidance, and the guidance policy is initialized from SFT using descriptions generated by a strong model (e.g. Gemini-3.5-flash). The two adapters have disjoint trainable parameters while sharing the same frozen backbone. For each task-video pair, we sample N candidate step-by-step guides (see prompt in the Appendix A) from the guidance policy,

$$g_i \sim \pi_{\mathrm{g},\phi_{\mathrm{g}}}(\cdot \mid q, V), \qquad i = 1, \ldots, N. \tag{1}$$

Each candidate guidance is then evaluated using the action policy on the *entire* ground-truth trajectory. Specifically, we teacher-force the full trajectory and compute its sequence-level log-likelihood conditioned on g_i as shown in Eqn. (2)

$$r_i = \frac{1}{T} \sum_{t=1}^{T} \log \pi_{\mathrm{a},\phi_\mathrm{a}} \left(a_t^* \mid q, o_{\le t}, a_{<t}^*, g_i \right). \tag{2}$$

Thus, each generated guidance receives a single scalar reward determined by the likelihood of the complete ground-truth trajectory. The reward does not assess whether the guidance improves prediction at any particular interaction step in isolation. Instead, it measures its overall utility for explaining the demonstrated task execution. Guidance that makes the full ground-truth trajectory more likely under the action policy receives a higher reward. When computing this reward, the action policy is used only as a scorer, and gradients are not propagated through it to the guidance policy.

We optimize the guidance LoRA using Group Relative Policy Optimization (GRPO), with rewards normalized using a leave-one-out (RLOO) (Ahmadian et al., 2024) baseline within the N guidance samples generated for the same task–video pair. The advantage of candidate g_i is

$$\hat{A}_i = r_i - \frac{1}{N-1} \sum_{j=1, j\neq i}^{N} r_j. \tag{3}$$

The resulting sequence-level advantage is assigned to the tokens of the corresponding text guide and optimized using the standard clipped GRPO (Shao et al., 2024) objective. Since all candidates are generated for the same task and video, the RLOO baseline provides a direct relative comparison between alternative guidances while reducing sensitivity to differences in task difficulty.

Simultaneously, we train the action LoRA to better utilize the sampled guidances. Rather than treating all generated guidances equally, we use their trajectory-level rewards to assign a normalized importance weight as shown in Eqn. (4).

$$w(g_i) = \mathrm{Softmax}(g_i / T_w), \tag{4}$$

where T_w is a temperature controlling the concentration of the weighting distribution. The action policy is trained with weighted cross-entropy (CE) as shown in Eqn. (5).

$$\mathcal{L}_{\mathrm{action}} = -\sum_{i=1}^{N} w(g_i) \left[\frac{1}{T} \sum_{t=1}^{T} \log \pi_{\mathrm{a},\phi_\mathrm{a}} \left(a_t^* \mid q, o_{\le t}, a_{<t}^*, g_i \right) \right]. \tag{5}$$

This weighting couples the two learning objectives. Consequently, the guidance LoRA is encouraged to generate video-derived information that better supports the complete oracle trajectory, while the action LoRA is preferentially trained on guidances that are already more useful for downstream action prediction. Moreover, instead of using the entire dataset, which has been used in SFT for the action policy, we first perform per-step action prediction to select the top 30% of trajectories that are more error-prone to be used as the training set for RL.

3.3 Video-based Self-verification

To perform self-verification using video as the only interface, we synthesise videos from trajectories by joining screenshots together, where each screenshot occupies a duration determined by the duration of the synthesized speech of the action description[1]. This way, the pipeline simulates the situation where the agent inspects the demonstration video it recorded to judge its performance.

Given each execution video, we prompt the model to give an integer score of 0 to 9, with detailed grades specified in Appendix A. Then, following Kwok et al. (2026), we compute the expected score by averaging the token probabilities with their values as shown in Eqn. (6).

$$s_\tau = \sum_{c=0}^{9} c P(c | V_\tau, q; \theta) \tag{6}$$

where V_τ denotes the video of the execution trajectory τ, and θ denotes the backbone video-SALMONN 2+ without the two LoRA adapters. This score is more robust than simply choosing the score with the highest probability, especially when the boundaries between the grades are inevitably ambiguous, and the model may assign high probabilities to two or more neighbouring scores.

[1]The VEX framework prompt the agent to generate a description of action followed by the code.

4 EXPERIMENTAL SETUP

4.1 VIDEO-AUGMENTED DATASETS

The video-SALMONN-VEX agent is trained on AgentNet (Wang et al., 2025). As part of our contribution, we augment AgentNet with 20k tutorial videos covering 7k tasks. These videos are selected using the video selection pipeline illustrated in Fig. 2, with relevance scoring omitted. For each video-task pair, we collect reference guidance text generated by Gemini-3.5-flash, which can be used for cold-start fine-tuning of the guide-generation LoRA adapter. Statistics for the videos and reference guidance texts are provided in Appendix B.

We primarily evaluate agents on OSWorld-Verified (Xie et al., 2024) using all tasks. To assess the generalization of the proposed methods, we additionally evaluate on a subset of 220 tasks from CUA-Gym (Wang et al., 2026) that complements the OSWorld tasks. There are 49 PDF tasks from CUA-Gym incorporated since this requires software not covered by OSWorld-Verified. Details of this subset are provided in Appendix B. Following the video collection pipeline, we obtain 2k tutorial videos for OSWorld and 3k for CUA-Gym from YouTube that are below 10 mins. When infrastructure failure happens, we start the task from beginning.

4.2 MODEL, TRAINING AND INFERENCE CONFIGURATIONS

The action-generation LoRA adapter which is applied only to the LLM backbone in video-SALMONN-VEX is first fine-tuned on the full AgentNet dataset for 2 epochs, using video-SALMONN 2+ as the backbone. We denote the resulting model as *video-SALMONN 2+ SFT* in subsequent results. This stage restores computer-use capabilities degraded by the video-oriented fine-tuning of video-SALMONN 2+. Separately, the guidance-generation LoRA adapter is also fine-tuned with LoRA on the LLM part using the reference guidance texts generated by Gemini-3.5-flash, with the same video-SALMONN 2+ backbone. Bothe th adapters use a rank of 64 and an alpha value of 128. Fine-tuning the two adapters takes 12 hours in total on 32 H100 GPUs.

Next, we generate an action at each step of every training sample, conditioned on the oracle action history, and compute the proportion of steps with incorrect predictions. We select the 30% of training samples with the highest error rates for AGGO training. For each video-task pair, we generate 9 guidance samples and set $T_w = 0.2$. AGGO training runs for 2 epochs on this selected subset and takes 48 hours on 32 H100 GPUs. We take the last model checkpoint at the end of epoch 2 across our experiments. Model checkpoints and code are available online.[2]

The tutorial videos are processed at 1FPS with a maximum number of 768 frames at a resolution of 360p. The trajectory videos are made using Kokoro TTS and `ffmpeg` to glue screenshots and audios together. Unless otherwise specified, VEX uses a maximum of 3 trials and a self-verification threshold of 7.0 during inference. We examine both hyperparameters in the ablation study. If the available tutorial videos are used up before the trial limit is reached and the self-verification score remains below the threshold, the agent uses its remaining trials without tutorial videos. Each trial starts by re-configuring the machine to the beginning status. Execution stops when either the trial limit is reached or the self-verification score meets the threshold. If that happens, the final state of the trail with the highest verification score is used for scoring.

We also apply VEX to Gemini-3.5-flash, a larger and stronger proprietary model, to assess its benefits over the baseline without VEX. Gemini-3.5-flash receives guidance generated either by itself or by the trained video-SALMONN-VEX model, allowing us to evaluate whether AGGO training improves the extraction of actionable skills from videos. For task execution with both video-SALMONN-VEX and Gemini-3.5-flash, we adapt the official OSWorld harness for Qwen.[3]

4.3 BASELINE SETUPS

To evaluate the effectiveness of VEX, we compare it with two single-agent baselines: video-SALMONN 2+ and its underlying Qwen3-VL backbone, which have comparable model sizes and language backbones. We additionally include a three-trial baseline in which each task is attempted

[2] https://anonymous.4open.science/r/VideoCUA-27E4/

[3] https://github.com/xlang-ai/OSWorld/tree/main/mm_agents/qwen

Table 1: Success rate on OSWorld using video-SALMONN-VEX compared to its backbone models including Qwen3-VL, video-SALMONN 2+ and that fine-tuned on AgentNet (SFT). We use average results across 3 trials for video-SALMONN 2+ SFT @ 3 trials, filtering out unfinished trajectories, and use the VEX framework for video-SALMONN-VEX with a maximum 3 trials.

Models	OS	Office	Daily	Professional	Workflow	Overall
Qwen3-VL (Bai et al., 2025a)	35.15	28.57	46.24	65.31	20.82	36.52
vSALMONN 2+ (Tang et al., 2025)	26.70	21.43	38.57	38.78	11.11	25.66
vSALMONN 2+ SFT	39.43	25.71	52.02	61.22	13.67	35.52
vSALMONN 2+ SFT @ 3 trials	39.19	27.38	56.53	53.74	15.17	36.10
vSALMONN 2+ SFT + Traj. (Liu et al., 2026)	35.20	29.28	54.64	58.16	17.35	36.65
video-SALMONN-VEX (ours)	41.67	31.43	63.94	61.54	16.97	**41.65**

Table 2: Success rate on CUA-Gym using video-SALMONN-VEX compared to its backbone models including Qwen3-VL, video-SALMONN 2+ and that fine-tuned on AgentNet (SFT). We use average results across 3 trials for video-SALMONN 2+ SFT @ 3 trials, filtering out unfinished trajectories, and use VEX framework for video-SALMONN-VEX with maximum 3 trials.

Models	Office	Daily	Professional	Workflow	PDF	Overall
Qwen3-VL (Bai et al., 2025a)	8.94	22.00	25.51	14.81	2.04	11.87
vSALMONN 2+ (Tang et al., 2025)	9.99	17.00	15.38	12.50	6.12	11.05
vSALMONN 2+ SFT	6.76	25.50	25.64	6.25	11.22	12.70
vSALMONN 2+ SFT @ 3 trials	8.61	21.17	23.38	10.00	8.20	12.45
vSALMONN 2+ SFT + Traj. (Liu et al., 2026)	10.29	26.00	27.19	15.21	13.16	15.72
video-SALMONN-VEX (ours)	12.86	29.00	26.92	15.62	11.84	**17.37**

three times, reporting the average success rate across attempts. This reflects the absence of an oracle success signal during execution, which prevents selecting a successful attempt in real time. Finally, following Liu et al. (2026), we use video-SALMONN 2+ SFT to convert tutorial videos into step-by-step textual trajectories and include these trajectories in the agent's context. This provides a stronger baseline that uses experience extracted from online videos. Ablation studies were performed in Section 5.2 where: (i). the isolated effect of AGGO training, and (ii). same multiple trial budgets without video tutorials were examined with their respective baselines.

5 REUSLTS

5.1 VIDEO-SALMONN-VEX MAIN RESULTS

Tables 1 and 2 summarize the main results on OSWorld and CUA-Gym, respectively. The categorization of tasks is based on the OSWorld official split (Xie et al., 2024). video-SALMONN-VEX achieves the highest overall success rate among the evaluated methods on both benchmarks, reaching 41.65% on OSWorld and 17.37% on CUA-Gym. Compared with video-SALMONN 2+ SFT, these results represent relative success rate gains of 17% and 37%, respectively. VEX also outperforms the underlying Qwen3-VL backbone by 14% relatively on OSWorld and 46% on CUA-Gym. The improvements across both benchmarks support the effectiveness of VEX for computer-use tasks, including the complementary tasks evaluated in CUA-Gym.

Comparison with repeated execution and textual trajectories. The expected succes rate across three attempts yields limited gains over single-attempt SFT on OSWorld, increasing success from 35.52% to 36.10%, and slightly reduces performance on CUA-Gym from 12.70% to 12.45%. Incorporating video-derived trajectories improves the overall success rate to 36.65% and 15.72%, respectively, suggesting that tutorial videos provide useful procedural information. VEX further improves over this trajectory-based baseline by 5.00% on OSWorld and 1.65% on CUA-Gym, while largely exceeding the three-trial baseline. These comparisons demonstrate the benefit of the complete VEX framework. The ablation studies further isolate the contributions of each component in Section 5.2. The results across different tasks indicate that VEX improves performance across a range of tasks, while leaving some room to improve in Workflow and Professinal categories.

Table 3: Ablation study on key design of video-SALMONN-VEX on OSWorld. When video tutorial is not used but self-verification is turned on, we randomly sample 3 trajectories and perform verification to select the best performing trajectory. When using video tutorial for video-SALMONN 2+ SFT, we directly use tutorial video as the context without extracting guidance.

Model	Video Tutorial	Self-Verification	Success Rate	Difference
video-SALMONN-VEX	✓	✓	41.65	0.00
video-SALMONN-VEX	✓	✗	36.65	-5.00
video-SALMONN-VEX	✗	✓	37.47	-4.18
video-SALMONN-VEX	✗	✗	35.52	-6.13
video-SALMONN 2+ SFT	✓	✗	30.76	-10.89
video-SALMONN 2+ SFT	✗	✗	35.52	-6.13

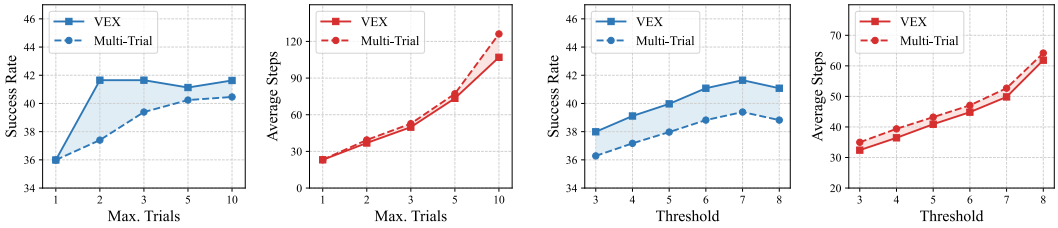


Figure 3: Variation of success rate and average steps per task for VEX and a multi-trial counterpart, under different maximum number of trial settings and threshold values. The multi-trial counterpart does not use tutorial videos, only keep trying until threshold is met or maximum trials is reached.

5.2 Ablation Studies

Different functionality of VEX components are shown in Table 3, including the video tutorials and the self-verification on OSWorld. The full VEX framework achieves a success rate of 41.65%. Removing self-verification reduces performance to 36.65%, while removing video tutorials degrades it to 37.47%. Disabling both components further reduces success to 35.52%, a decline of an absolute 6.13% compared to the full VEX framework. Both components therefore contribute to performance, with their combination outperforming either component alone.

Importantly, directly providing tutorial videos as context to video-SALMONN 2+ SFT decreases success from 35.52% to 30.76%. This contrast suggests that access to videos alone is insufficient: extracting actionable guidance is important for making effective use of tutorial content, while self-verification provides an additional improvement through trajectory selection.

Video guidance improves the effectiveness and efficiency of repeated trials. Fig. 3 compares VEX with a multi-trial baseline that uses the same self-verification loop without tutorial videos, varying the trial limit and verification threshold. Increasing the trial limit from 1 to 2 substantially improves VEX's success rate, achieving substantially better performance compared to merely trying multiple times, which demonstrate the effectiveness of the video guidance. The gain becomes smaller since the tutorial videos run out, and the rest of the trials are bootstrapped by repeated trials without additional information. Moreover, VEX framework always achieves a lower average number of steps per task compared to simple multi-trial baseline. Specifically at a maximum of 3 trials, VEX framework requires 40 steps on average, whereas the multi-trial baseline requires 43 steps. Similarly, increasing the verification threshold improves both methods up to 7.0, but further increasing it to 8.0 reduces success and requires more steps. Across all tested thresholds, VEX achieves higher success rates with fewer steps.

Video-based self-verification yields higher task success than screenshot-and-text verification. The comparison between video-based verification in VEX, and trajectory-based (screenshot + text) verification, is shown in Table 4. We deliberately choose Qwen3-VL instead of video-SALMONN 2+ for screenshot + text verification, since it performs better in understanding screenshots, and any improvements compared to Qwen3-VL, will be more salient when compared against video-SALMONN 2+. Overall, video-based verification achieves better success rate with an abso-

Table 4: Comparison between video and trajectory (screenshot plus text description) as the representation of the trial for self-verification. Pearson Correlation Coefficient (PCC) is between the predicted verification score and actual per-task actual score, and AUROC is computed by converting per-task actual score into binary with threshold of 0.5.

Representation	Model for Verification	PCC	AUROC	Success Rate
Video	video-SALMONN-VEX	0.4503	0.7984	41.65
Video	Gemini-3.5-flash	**0.6247**	**0.8327**	**41.68**
Screenshot + Text description	Qwen3-VL	0.3240	0.7848	39.77

Table 6: Success rate on OSWorld and GUA-Gym using **Gemini-3.5-flash** with VEX framework. No tutorial means we perform multiple trials with VEX framework without tutorial videos.

Guidance Generation Model	Method	OSWorld	CUA-Gym	Overall
Gemini-3.5-flash	Standard	61.69	47.13	56.07
Gemini-3.5-flash	3 trials	63.60	45.07	56.45
Gemini-3.5-flash	VEX (no tutorial)	67.72	50.12	60.93
Gemini-3.5-flash	VEX	67.38	**52.70**	61.71
video-SALMONN-VEX	VEX	**68.67**	52.34	**62.37**

lute 1.88% improvement than trajectory-based counterparts. The video-based configuration also achieves higher Pearson correlation with task scores (0.4503 vs. 0.3240) and AUROC (0.7984 vs. 0.7848). Using Gemini-3.5-flash as the video verifier further improves these metrics to 0.6247 and 0.8327, respectively, but yields only a marginal increase in task success to 41.68%.

Table 5: Effect of dual-LoRA AGGO compared with single-LoRA AGGO and guidance SFT using Gemini-generated descriptions.

Training Setting	Success Rate	Difference
AGGO	41.65%	0.00%
AGGO (single LoRA)	37.06%	−4.59%
Guidance SFT	39.39%	−2.26%

Dual-LoRA AGGO outperforms both single-LoRA training and guidance SFT. Table 5 shows that AGGO achieves a success rate of 41.65%, exceeding guidance SFT with Gemini-generated descriptions by 2.26%. This improvement indicates that AGGO provides benefits beyond supervised imitation of reference guidance. Replacing the dual-LoRA design with a single adapter reduces success to 37.06%, a degradation of 4.59%. These results support maintaining separate adapters for guidance generation and action generation, suggesting that separating the two roles helps preserve effective task execution during guidance optimization.

5.3 Transferability to Gemini-3.5-flash

Table 6 shows that VEX with Gemini's self-generated guidance improves overall success from 56.07% to 61.71%, exceeding both the three-trial baseline and VEX without tutorials. Replacing this guidance with descriptions generated by video-SALMONN-VEX further increases overall success to 62.37%, demonstrating that its learned guidance can benefit a stronger proprietary agent. Overall, a 10% relative improvement in success rate on both datasets is achieved, demonstrating the effectiveness of VEX framework, and the knowledge gain by learning from tutorial videos. Example guidance generated by Gemini-3.5-flash and video-SALMONN-VEX is shown in Appendix C.

6 Conclusion

We introduced VEX, a watch, act and rewatch framework that uses video as a unified representation for learning from demonstrations and verifying an agent's own behavior. We instantiated it with the 8B-scale video-SALMONN-VEX, proposed AGGO to optimize guidance for downstream action prediction, and augmented three datasets with over 24K tutorial videos. Experiments demonstrate improved computer-use performance and transfer to Gemini-3.5-flash, highlighting the potential of video experience for procedural learning and self-verification.

AI USE STATEMENT

In this work, we used generative AI tools for generating descriptions for tutorial videos in our datasets, and polishing paper writing. We have not used generative AI tools for retrieval and discovery, research ideation or execution or draft sections of the paper, and proving mathematical claims are is applicable to this work. Additionally, we used generative AI tools for critical feedback on the complete draft. We have reviewed all AI-assisted work. For example, we checked the polished texts word by word to ensure they still maintain our original intention and the key facts are correct. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

VEX aims to improve computer-use agents through learning from video demonstrations and reviewing their own execution. Tutorial videos and execution recordings may contain personal information, copyrighted material, or sensitive application content. Their collection, use, and release therefore require attention to privacy, permissions, and applicable usage restrictions. Public availability alone should not be treated as permission to redistribute content. More capable computer-use agents may also facilitate unauthorized actions or harmful automation. Deployment should include appropriate access controls, human oversight for consequential actions, and protections for sensitive data. Finally, self-verification is imperfect. A high verification score does not guarantee successful, safe, or authorized execution. Our benchmark results establish task-performance improvements rather than safety guarantees for unrestricted deployment.

REPRODUCIBILITY STATEMENT

We describe the training procedures, key hyperparameters, evaluation protocols, and baseline configurations in the experimental setup section, with additional dataset details in the appendix. We ensure the experiments are fully reproducible by releasing the experimental code, including data preparation, video data annotation, training script, model checkpoints, and scoring scripts. Experiments involving proprietary models may be affected by API availability and model updates. Internet browser-based tasks may be subject to the change to those website.

REFERENCES

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Eric Wang. Agent S: An open agentic framework that uses computers like a human. In *Proc. ICLR*, 2025.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proc. ACL*, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL Technical Report. *arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report. *arXiv:2502.13923*, 2025b.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proc. ACL*, 2024.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In *Proc. NeurIPS*, 2023.

Tianyu Fu, Anyang Su, Chenxu Zhao, Hanning Wang, Minghui Wu, Zhe Yu, Fei Hu, Mingjia Shi, Wei Dong, Jiayao Wang, Yuyang Chen, Ruiyang Yu, Siran Peng, Menglin Li, Nan Huang, Haitian Wei, Jiawei Yu, Yi Xin, Xilin Zhao, Kai Gu, Ping Jiang, Sifan Zhou, and Shuo Wang. Mano technical report. *arXiv:2509.17336*, 2025.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *Proc. ICLR*, 2025.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. WebVoyager: Building an end-to-end web agent with large multimodal models. In *Proc. ACL*, 2024.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents. In *Proc. CVPR*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *Proc. ICLR*, 2022.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proc. ACL*, 2024.

Jacky Kwok, Shulu Li, Pranav Atreya, Yuejiang Liu, Yixing Jiang, Chelsea Finn, Marco Pavone, Ion Stoica, and Azalia Mirhoseini. Llm-as-a-verifier: A general-purpose verification framework. *arXiv:2607.05391*, 2026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Pengxiang Li, Zechen Hu, Zirui Shang, Jingrong Wu, Yang Liu, Hui Liu, Zhi Gao, Chenrui Shi, Bofei Zhang, Zihao Zhang, Xiaochuan Shi, Zedong YU, Yuwei Wu, Xinxiao Wu, Yunde Jia, Liuyu Xiang, Zhaofeng He, and Qing Li. Efficient multi-turn rl for gui agents via decoupled training and adaptive data curation. *arXiv:2509.23866*, 2025.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. VILA: On pre-training for visual language models. In *Proc. CVPR*, Seattle, 2024.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent. In *Proc. CVPR*, 2025.

Yujian Liu, Ze Wang, Hao Chen, Ximeng Sun, Xiaodong Yu, Jialian Wu, Jiang Liu, Emad Barsoum, Zicheng Liu, and Shiyu Chang. Learning from online videos at inference time for computer-use agents. *Transactions on Machine Learning Research*, 2026.

Dunjie Lu, Yiheng Xu, Junli Wang, Haoyuan Wu, Xinyuan Wang, Zekun Wang, Junlin Yang, Hongjin Su, Jixuan Chen, Junda Chen, Yuchen Mao, Jingren Zhou, Junyang Lin, Binyuan Hui, and Tao Yu. Videoagenttrek: Computer use pretraining from unlabeled videos. In *International Conference on Learning Representations (ICLR)*, 2026.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, Xiaojun Xiao, Kai Cai, Chuang Li, Yaowei Zheng, Chaolin Jin, Chen Li, Xiao Zhou, Minchao Wang, Haoli Chen, Zhaojian Li, Haihua Yang, Haifeng Liu, Feng Lin, Tao Peng, Xin Liu, and Guang Shi. UI-TARS: Pioneering automated GUI interaction with native agents. *arXiv:2501.12326*, 2025.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William E Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Kenji Toyama, Robert James Berry, Divya Tyamagundlu, Timothy P Lillicrap, and Oriana Riva. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In *Proc. ICLR*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*, 2024.

Chan Hee Song, Yiwen Song, Palash Goyal, Yu Su, Oriana Riva, Hamid Palangi, and Tomas Pfister. Watch and learn: Learning to use computers from online videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5421–5431, 2026.

Guangzhi Sun, Yudong Yang, Jimin Zhuang, Changli Tang, Yixuan Li, Wei Li, Zejun MA, and Chao Zhang. video-SALMONN-o1: Reasoning-enhanced audio-visual large language model. In *Proc. ICML*, 2025.

Guangzhi Sun, Yixuan Li, Xiaodong Wu, Yudong Yang, Wei Li, Zejun Ma, and Chao Zhang. video-salmonn s: Memory-enhanced streaming audio-visual llm. In *Proc. ICML*, 2026.

Changli Tang, Yixuan Li, Yudong Yang, Jimin Zhuang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. video-salmonn 2: Captioning-enhanced audio-visual large language models. *arXiv:2506.15220*, 2025.

Bowen Wang, Dunjie Lu, Junli Wang, Tianyi Bai, Shixuan Liu, Zhipeng Zhang, Haiquan Wang, Hao Hu, Tianbao Xie, Shuai Bai, Dayiheng Liu, Que Shen, Junyang Lin, and Tao Yu. Cua-gym: Scaling verifiable training environments and tasks for computer-use agents. *arXiv:2605.25624*, 2026.

Xinyuan Wang, Bowen Wang, Dunjie Lu, Junlin Yang, Tianbao Xie, Junli Wang, Jiaqi Deng, Xiaole Guo, Yiheng Xu, Chen Henry Wu, Zhennan Shen, Zhuokai Li, Ryan Li, Xiaochuan Li, Junda Chen, Boyuan Zheng, Peihang Li, Fangyu Lei, Ruisheng Cao, Yeqiao Fu, Dongchan Shin, Martin Shin, Jiarui Hu, Yuyan Wang, Jixuan Chen, Yuxiao Ye, Danyang Zhang, Dikang Du, Hao Hu, Huarong Chen, Zaida Zhou, Haotian Yao, Ziwei Chen, Qizheng Gu, Yipu Wang, Heng Wang, Diyi Yang, Victor Zhong, Flood Sung, Y. Charles, Zhilin Yang, and Tao Yu. OpenCUA: Open foundations for computer-use agents. *arXiv:2508.09123*, 2025.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. Os-atlas: A foundation action model for generalist gui agents. In *Proc. ICLR*, 2025.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Proc. NeurIPS*, 2024.

Weimin Xiong, Shuhao Gu, Bowen Ye, Zihao Yue, Lei Li, Feifan Song, Sujian Li, and Hao Tian. Video2gui: Synthesizing large-scale interaction trajectories for generalized gui agent pretraining. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2605.14747.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. Aguvis: Unified pure vision agents for autonomous GUI interaction. In *Proc. ICML*, 2025.

Taofeng Xue, Chong Peng, Mianqiu Huang, Linsen Guo, Tiancheng Han, Haozhe Wang, Jianing Wang, Xiaocheng Zhang, Xin Yang, Dengchang Zhao, Jinrui Ding, Xiandi Ma, Yuchen Xie, Peng Pei, Xunliang Cai, and Xipeng Qiu. Evocua: Evolving computer use agents via learning from scalable synthetic experience. *arXiv:2601.15876*, 2026.

Yan Yang, Dongxu Li, Yutong Dai, Yuhao Yang, Ziyang Luo, Zirui Zhao, Zhiyuan Hu, Junzhe Huang, Amrita Saha, Zeyuan Chen, Ran Xu, Liyuan Pan, Silvio Savarese, Caiming Xiong, and Junnan Li. GTA1: Gui test-time scaling agent. In *Proc. ICLR*, 2026a.

Yudong Yang, Guangzhi Sun, Yixuan Li, and Chao Zhang. Slvmbench: Skill learning from video memory. *arXiv:2607.11312*, 2026b.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Proc. NeurIPS*, 2022.

Bofei Zhang, Zirui Shang, Zhi Gao, Wang Zhang, Rui Xie, Xiaojian Ma, Tao Yuan, Xinxiao Wu, Song-Chun Zhu, and Qing Li. Tongui: Internet-scale trajectories from multimodal web tutorials for generalized gui agents. In *Proc. AAAI*, 2026.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. VideoLLaMA 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025a.

Chaoyun Zhang, Liqun Li, Shilin He, Xu Zhang, Bo Qiao, Si Qin, Minghua Ma, Yu Kang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. UFO: A UI-focused agent for Windows OS interaction. In *Proc. NAACL*, 2025b.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic Data. *arXiv preprint arXiv:2410.02713*, 2024.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *Proc. ICLR*, 2024.

A Prompt Templates

A.1 Video Guidance Extraction and Apply

Video Guidance Extraction Prompt

<tutorial video input>

You are given the above video tutorial teaching you how to perform some computer use tasks.

You will also be given a task description.

Use ONLY the knowledge from the video, describe how to perform the task step by step.
The video may not contain the entire task. It may only contain a part of the task. Just describe the part of the task that is shown in the video.

For example, for a long task where part of it requires you to login with username and password that is included in the video, you should output the following:
When you want to log in, you should first click the login button on the left menu bar. Then, this will direct you to the login page. You need to enter your username and password. You should first click on the textbox with a word "Username" and then enter your username. Then, click on the textbox with a word "Password" and enter your password.

Rules
1. Do not use your own knowledge. Just describe what you can learn from the video.
2. Always output step by step, and at the end of each step, describe what screen status you are expected to get.
3. Try to gather as much information/procedural knowledge as possible from the video.
4. If the video contains the entire task, describe the entire task.
5. If the video really does not contain anything meaningful, output "No information available".

Task description:

{task}

Initial screenshot:
<image of the initial screenshot>

Output format
{
"description": "The detailed knowledge you learned from the video about the given task."
}

Action Prediction with Video Guidance Prompt

{tool_definitions}

{output_format_definitions}

Task Instruction
{task}

The following describes the knowledge that supports this task from a tutorial video:
{guidance}
You can use it as the guide to perform the task.

History action sequence:
{history}

A.2 VIDEO-BASED VERIFICATION PROMPT

Video-based Verification Prompt

You are an expert GUI agent reviewer. You will watch the following video of an agent performing a task while describing its action every step.

The task is:
{task}

You need to evaluate the completion of this task, with the following score schemes:
0-1 — No meaningful progress; wrong app, crash, infinite repetition, or immediate failure.
2-3 — Correct environment but core action not executed; loops or early abandonment.
4-5 — Main action attempted; deliverable exists but is flawed or missing key constraints.
6-7 — Task complete with minor errors (e.g., slight formatting, naming) or inefficient path.
8-9 — Fully correct final state, all constraints satisfied, clean and robust execution.

You should focus on the following aspects in your evaluation:
- Final frames: Is the exact deliverable visible and correct (saved file, sent email, correct UI state)?
- Constraints: Were ALL specific requirements from the instruction fulfilled?
- Coherence: Penalize unresolved dialogs, unsaved files, and infinite loops.

Carefully analyze the video, then directly provide your final scores in the following format:
{
"score": "your predicted score"
}

A.3 Video Search and Relevance Scoring Prompt

The prompt used to generate query and the coarse filtering of videos based on simple text-based queries. The description required is just downloaded description made by the content creators themselves from YouTube.

Video Guidance Extraction Prompt

You are given the following computer use task instruction:
{task}

Generate 10 search queries to search tutorial videos online.

Output format:
```
{
  "queries": [
    "query 1",
    "query 2"
  ]
}
```

Video Relevance Filtering Prompt

You are given the following computer use task instruction:
{task}

You are given the following video title and description:

Title:
{title}

Description:
{description}

Determine the relevance of the video in a scale of 0 to 5, where 0 means completely irrelevant, and 5 means exact task.

B Data Statistics

B.1 CUA-Gym Task Distribution

Task distribution of the CUA-Gym subset is shown in Fig. 4. The tasks are selected randomly from the pool of tasks complementary to OSWorld tasks.

B.2 OSWorld and CUA-Gym Tutorial Video Statistics

The length distribution in terms of number of words for guidance text generated by Gemini-3.5-flash is shown in Fig. 5. The video durations are kept under 10 minutes.

B.3 AgentNet Tutorial Video Statistics

The duration of tutorial videos, and the length distribution in number of words are shown in Fig. 6 for AgentNet used in SFT.

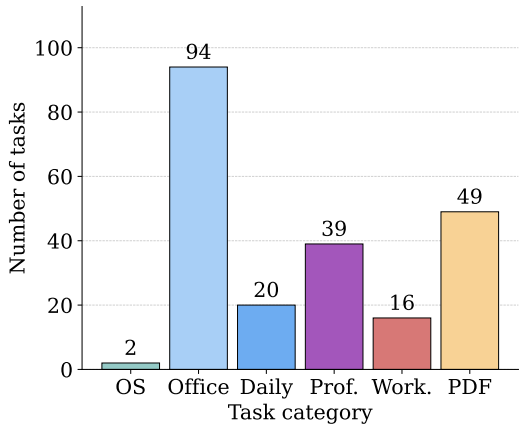


Figure 4: Task count distribution on CUA-Gym subset.

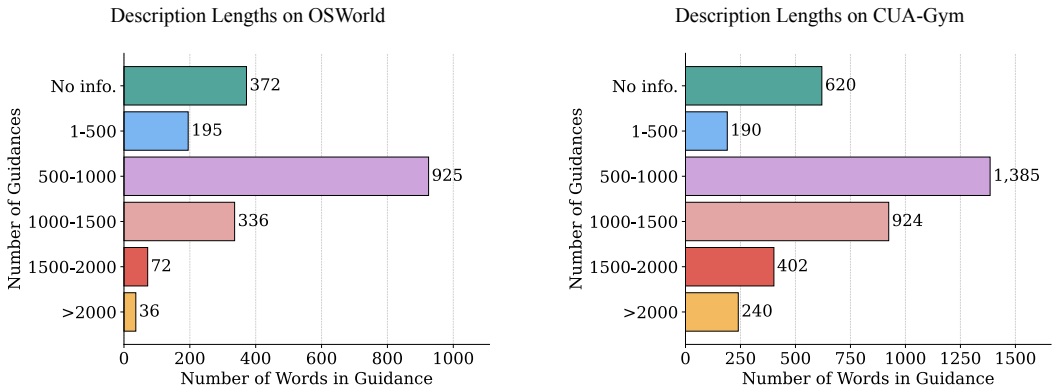


Figure 5: Length distributions in terms of the number of words per guidance on OSWorld (left) and CUA-Gym (right). No info. refers to the case where Gemini-3.5-flash is unable to obtain useful information from the video to help the task.

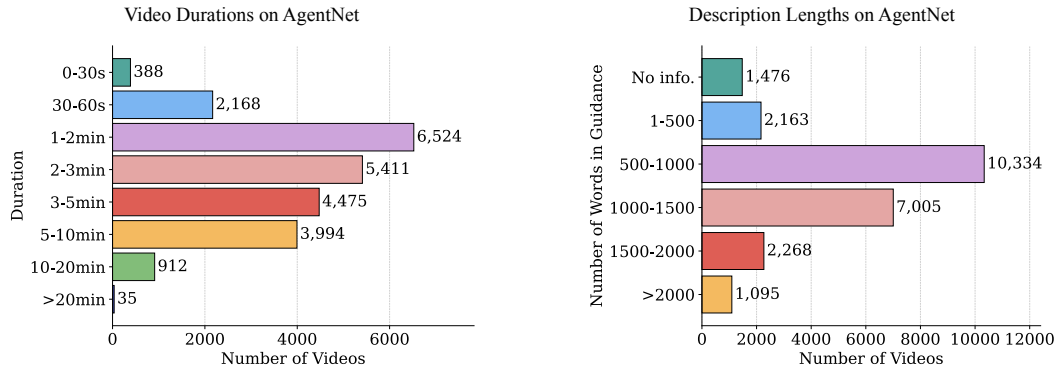
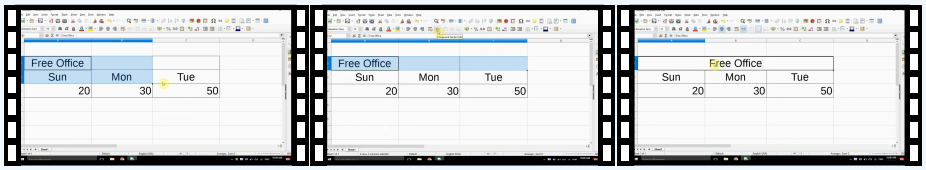


Figure 6: Duration distribution of tutorial videos (left) and word count for Gemini-3.5-flash guidance texts (right) in AgentNet used for training.

Task Instruction:

Create a new sheet named "Sheet2" and merge cells A1:C1 to write the header "Investment Summary". Beneath that, merge cells A2:B2 to write "High Interest Rate" and merge cells C2:D2 to form "Low Interest Rate".

Tutorial Video:



Gemini Guidance:

Based on the video tutorial, you can learn how to merge cells in LibreOffice Calc:

First, select the range of cells you want to merge by clicking and dragging across them (for example, cells A2 to C2).

Click the "Merge and Center Cells" button on the formatting toolbar.

This button is located next to the text alignment options and will combine the selected cells into a single cell, centering any text within it.

If you want to unmerge the cells at any point, select the merged cell and click the "Merge and Center Cells" button again.

video-SALMONN-VEX Guidance:

Based on the video tutorial, here is how you can perform the task step by step:

First, click on the "+" icon next to the "Sheet1" tab at the bottom of the screen to create a new sheet. This will create a new sheet named "Sheet2".

To merge cells A1:C1, click and drag to select cells A1, B1, and C1. Then, click the "Merge and Center Cells" icon on the toolbar.

Type "Investment Summary" into the merged cell A1:C1.

To merge cells A2:B2, click and drag to select cells A2 and B2. Then, click the "Merge and Center Cells" icon on the toolbar.

Type "High Interest Rate" into the merged cell A2:B2.

To merge cells C2:D2, click and drag to select cells C2 and D2. Then, click the "Merge and Center Cells" icon on the toolbar.

Type "Low Interest Rate" into the merged cell C2:D2.

Figure 7: Example guidance generated from Gemini-3.5-flash and video-SALMONN-VEX for one OSWorld task. Screenshot of the video is shown and audio of the video is also part of the input.

C Example Guidance Output

Two examples of output guidance text from Gemini-3.5-flash and video-SALMONN-VEX are shown in Fig. 7 and 8. The knowledge output from Gemini-3.5-flash are more faithful towards the video itself without paying much attention to the actual task setting. On the contrary, the output from video-SALMONN-VEX is more adapted to the specific task that we are performing. Both are effective with video-SALMONN-VEX guidance slightly outperforming Gemini-3.5-flash ones by 0.6% using a Gemini-3.5-flash-based agent as shown in Table 6.

Task Instruction:
Convert all Word documents (.docx files) in `/home/user/Documents/Reports` to PDF, saving them in folder `Reports_PDF` with same filenames.

Tutorial Video:



Gemini Guidance:
Based on the video, you can convert Word documents (.docx or .doc files) to PDF using the command-line tool `lowriter`, which is part of LibreOffice.
Open your terminal application.
You can batch convert files using the following command shown in the video, adapted for `.docx` files:
lowriter --convert-to pdf *.docx
Alternatively, to safely process multiple files without crashing LibreOffice, you can use a bash loop with the `--headless` flag as shown in Solution 3:
for f in *.docx; do lowriter --headless --convert-to pdf "$f"; done

video-SALMONN-VEX Guidance:
Based on the video, you can convert Word documents to PDF using the `lowriter` command-line tool.
Open the terminal by clicking on the terminal icon on the left dock.
Navigate to the directory containing your Word documents:
cd /home/user/Documents/Reports
Create the destination folder for the PDFs:
mkdir Reports_PDF
Use the `lowriter` command to convert all `.docx` files in the current directory to PDFs and save them in the `Reports_PDF` folder:
lowriter --headless --convert-to pdf *.docx --outdir Reports_PDF
This command will batch-convert all `.docx` files in the directory to PDFs, saving them in the `Reports_PDF` folder with the same filenames

Figure 8: Example guidance generated from Gemini-3.5-flash and video-SALMONN-VEX for one OSWorld task. Screenshot of the video is shown and audio of the video is also part of the input.