

Watch and Learn: Learning to Use Computers from Online Videos

Chan Hee Song^{3*} Yiwen Song¹ Palash Goyal¹ Yu Su³
 Oriana Riva^{2†} Hamid Palangi^{1†} Tomas Pfister^{1†}

¹Google Cloud AI Research ²Google DeepMind ³The Ohio State University

Abstract

Computer-using agents (CUAs) must plan task workflows across diverse and evolving applications, yet progress is limited by the lack of large-scale, high-quality training data. Existing datasets are narrow, static, and costly to annotate, while synthetic data often yields oversimplified or misaligned behaviors. We present Watch & Learn (W&L), a framework that converts readily available Internet videos of human computer use into executable UI trajectories at scale. Instead of directly generating actions or relying on handcrafted heuristics, we cast trajectory annotation as an inverse dynamics problem that predicts user actions from consecutive screen states, which simplifies learning and generalizes across domains. Through a task-aware retrieval and labeling pipeline, W&L yields over 53K high-quality trajectories that enhance CUAs both as in-context exemplars and as supervised training data. On OSWorld, it consistently improves general-purpose and specialized CUAs, while on WindowsAgentArena it achieves state-of-the-art performance among 7B-scale models under the 15-step limit. These results show that web-scale human demonstration videos can serve as a practical and scalable foundation for advancing real-world CUAs.

Project page: <https://chanh.ee/wandl/>.

1. Introduction

Computer-using agents (CUAs) [2, 3, 11, 12, 19, 25, 29, 36, 39, 42, 45] are emerging as a powerful paradigm for automating software and web interactions. From everyday productivity tools to enterprise workflows, CUAs must both *plan* multi-step task sequences that incorporate domain knowledge and *ground* these plans into concrete UI actions across diverse, constantly changing interfaces. Despite rapid progress, their scalability remains constrained by

*This work was partly done during the first author’s internship at Google Cloud AI Research.

†Co-last authors.

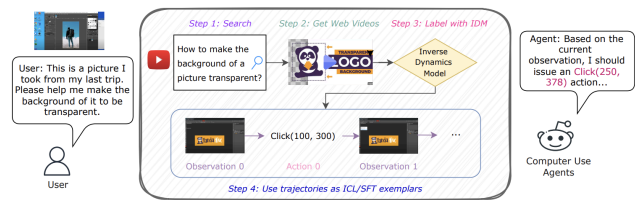


Figure 1. W&L converts web-scale human demonstration videos into executable UI trajectories, providing scalable supervision and in-context exemplars for computer-using agents.

the difficulty of obtaining large-scale, high-quality demonstrations of human-computer interaction [36].

Collecting such trajectories remains extremely costly and time-consuming. For instance, OpenCUA’s AgentNet dataset [36], which spans three operating systems and more than 200 applications and websites, required six months of dedicated annotation effort to label 22K tasks, with a total cost exceeding \$32,000 (about \$1.45 per task). Despite its impressive coverage, AgentNet underscores the scalability bottleneck of manual annotation: extending to the millions of trajectories needed for robust generalization would require substantial additional time and cost well over half a million dollars, even for well-funded labs. Such high costs and long turnaround times make it difficult to build datasets that capture the full diversity and complexity of real-world computer use, limiting progress in developing scalable computer-using agents.

Meanwhile, the web already contains an enormous volume of human demonstration videos, including tutorials, screencasts, and walkthroughs, which naturally encode rich task workflows across applications. If we could reliably convert these videos into structured action trajectories, CUAs could learn directly from human expertise at unprecedented scale. However, existing trajectory generation methods have struggled to achieve this vision.

Prior work falls into three broad categories. *Offline synthesis* reconstructs trajectories from recorded videos using multi-stage pipelines that couple multimodal large language models (MLLMs) with UI element detectors and transition parsers. Despite substantial engineering, systems such

as MONDAY [18] and TongUI [43] report modest accuracies (around 70% for MONDAY) due to cascading errors in heuristic pipelines. *Online synthesis* collects trajectories through random exploration and retrofits them with task instructions [22, 33], but these data are low in task complexity and expensive to gather. *Hybrid approaches*, such as Explorer [27], mix proposal generation with online refinement, yet still depend on MLLMs for action grounding, retaining many of the same limitations.

To address these limitations, we introduce **Watch & Learn (W&L)**, a framework that formulates trajectory recovery as an *inverse dynamics* prediction problem. Given two consecutive screen observations (O_t, O_{t+1}), the model predicts the intermediate action a_t that produced the transition. This one-step formulation is substantially easier to learn than generating entire trajectories, avoids hand-crafted parsing heuristics, and generalizes across applications. In robotics, inverse dynamics modeling has long been used to infer actions from state transitions (e.g., VPT [6], DreamGen [17]); here, we adapt this principle to computer use, showing it can recover accurate, executable actions from raw visual transitions.

We train an inverse dynamics model (IDM) on a large state-transition corpus containing more than 600K automatically labeled transitions from real-world web interactions. Each sample records (O_t, a_t, O_{t+1}), enabling direct supervision from interaction logs without manual annotation. We then pair this model with a task-aware retrieval module that locates relevant YouTube videos, either those specific to target benchmarks or general instructional content, and automatically converts them into structured UI trajectories. This pipeline yields 53,125 high-quality trajectories spanning a broad range of applications and task categories across three operating systems (Ubuntu, macOS, and Windows).

Beyond scalable data collection, W&L reveals a dual role for such trajectories. They can be used not only for supervised fine-tuning, but also as *in-context exemplars* at inference time, allowing CUAs to dynamically retrieve demonstrations as planning and grounding priors. This enables flexible integration with both open-source CUAs and general-purpose multimodal agents.

We evaluate W&L on two challenging benchmarks, OSWorld [38] and WindowsAgentArena [7], which require complex reasoning and precise UI grounding. On OSWorld, trajectories extracted from web videos yield consistent gains: in-context use improves general-purpose models and agentic frameworks by up to 3 percentage points, while supervised fine-tuning provides larger boosts for open-weight models (up to 11 points). On WindowsAgentArena, fine-tuning with our trajectories achieves leading performance among 7B-scale models under the 15-step evaluation limit, demonstrating that large-scale web demonstrations can translate effectively into high-quality training

data. These benefits are obtained without any manual annotation, highlighting the potential of web-scale human workflows as a practical and scalable foundation for advancing CUAs toward real-world deployment. In summary, this work makes the following contributions:

- We introduce a **inverse dynamics model** that recovers UI actions directly from visual transitions, removing the need for heuristic parsing or manual labeling.
- We develop a **task-aware retrieval framework** that converts web-scale human demonstration videos into 53K executable UI trajectories.
- We show that these trajectories **enhance CUAs** both as in-context exemplars and as supervised training data, achieving strong gains on OSWorld and WindowsAgentArena.

2. Related Work

2.1. Data Synthesis for Computer-Using Agents

Human-curated UI datasets [8, 20, 21, 31] remain limited in scale and diversity, constraining progress in CUAs. Recent research therefore explores synthesizing data through exploration, tutorials, or self-play.

Exploration-based synthesis methods such as BAGEL [22], NNetNav [23], Explorer [27], and OS-Genesis [33] let agents explore websites and retroactively label interactions with task instructions. This yields large-scale data but often introduces noise, since quality depends heavily on heuristics or multimodal LLM labeling.

Tutorial-driven synthesis leverages online instructional resources. Synatra [26] and AgentTrek [40] convert textual tutorials into executable trajectories, while TongUI [43] aggregates multimodal tutorials (text and screencast videos) into GUI data. These works show that web-scale content can broaden coverage across applications, though reliance on off-the-shelf LLMs often leads to brittleness and misalignment.

Self-improving agents integrate data synthesis directly into training. OpenWebVoyager [15] improves via online exploration and feedback, WebRL [28] creates new instructions from failures, SCA [28] generates and verifies code-based tasks, and ZeroGUI [41] trains GUI agents entirely through automated task and reward generation. These systems enable continual learning without human data but often yield narrow or repetitive task distributions and require expensive iterative retraining.

Our approach, *Watch & Learn*, also mines web videos like TongUI but differs fundamentally in supervision. Rather than labeling videos with MLLMs, we train an inverse dynamics model that infers user actions directly from screen transitions. This produces accurate UI trajectories that strengthen both supervised training and in-context reasoning. By combining large-scale video mining with precise action labeling, our framework complements prior syn-

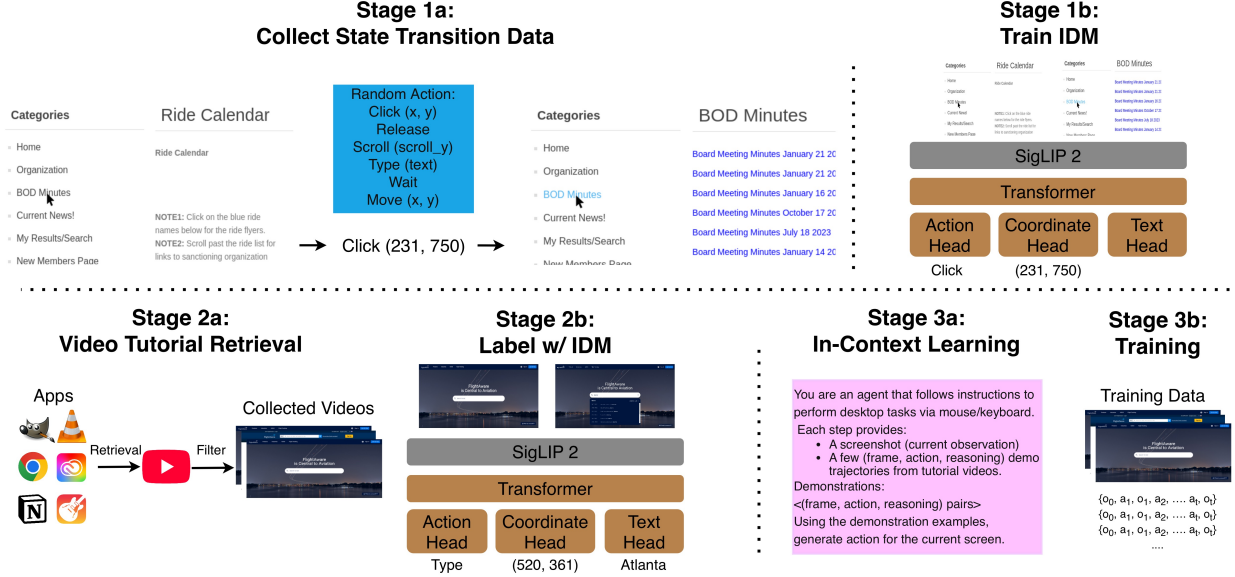


Figure 2. **Method overview.** Our framework converts web-scale human demonstration videos into executable trajectories for CUAs. We first collect a large-scale state-transition dataset of screen observations and user actions, and train an inverse dynamics model (IDM) to recover actions from consecutive screenshots. This IDM is then applied to tutorial videos to extract step-by-step trajectories. A retrieval module selects task-relevant or general demonstrations, which are used in two ways: (i) as in-context exemplars that provide application-specific knowledge at inference time, and (ii) as supervised training data to improve open-source CUAs.

thesis efforts and emphasizes the value of reliable video-based supervision for CUAs.

2.2. In-context Learning for Agents

In-Context Learning (ICL) has emerged as a pivotal test-time scaling paradigm for large language models, enabling them to adapt to new tasks without explicit parameter updates [9]. This approach is particularly useful for enhancing LLM-powered agentic systems [32].

Despite being generally helpful, the effectiveness of ICL is heavily influenced by the scale of the LLMs and the size of their context window, particularly for long-horizon, multi-step tasks. While including more ICL examples usually brings performance gains [1], this method incurs significant computational overhead and latency with long demonstration trajectories. Therefore, efficiently selecting demonstration sequences [14] or abstracting them in high-level workflows [37, 46] has become a promising research direction. For computer-using agents, where tasks are often long and complex, one major challenge is the model’s inability to plan effectively. Several pieces of work have leveraged ICL to address this specific problem [2, 3, 11, 16, 44].

Another important direction is to develop data-centric frameworks to adapt LLM agents to any given environments without human annotations [32]. However, such methods require generating large amounts of synthetic data, and the potential for using publicly available web-scale video data as ICL examples still remains underexplored.

3. Method

Computer use agents must operate the user interface of many diverse and ever-changing applications where internal UI representations such as HTML or accessibility trees are often incomplete, inconsistent, or unavailable. To maximize generality and scalability, we focus on a *vision-only* setting: models observe raw screen pixels and output structured user actions. This mirrors how humans interact with computers, by visually perceiving the interface and deciding where to click or what to type, while avoiding brittle dependencies on application-specific APIs or noisy UI representations.

At a high level, our framework works in three stages (see Figure 2). First, we construct a large-scale state-transition corpus from diverse computer interaction data and use it to train an inverse dynamics model (IDM), enabling the system to recover the underlying actions from consecutive screen observations. Second, we apply this IDM to web-scale tutorial videos, paired with a retrieval component that identifies either task-relevant videos (for inference-time use) or general tutorials (for training). This process automatically produces executable UI trajectories without manual labeling. Finally, we leverage these trajectories in two complementary ways: as *in-context exemplars*, which provide CUAs with planning and grounding priors as well as application-specific knowledge at inference time; and as *supervised training data*, which can be used to fine-tune models and improve their general knowledge.

3.1. Inverse Dynamics Model

A key component of our framework is an inverse dynamics model (IDM) that predicts the user action given two consecutive screen observations. Since large-scale state-transition data are scarce, we construct a new corpus by synthesizing interactions at scale.

State-transition data. To obtain large-scale supervision, we built an automated pipeline that interacts with live web pages and records transitions. Inspired by Web-Dreamer [13], we sample entry points from the March 2025 Common Crawl index and launch browsing sessions that execute actions drawn from the above schema. The sampling policy favors frequent actions (e.g., `click`) while ensuring sufficient diversity. This process yields roughly 600K (O_t, a_t, O_{t+1}) transitions. Further implementation details are provided in the Appendix.

Action space. The IDM models six atomic primitives that together capture the majority of real-world computer interactions: `Click`, `Release`, `Scroll`, `Type`, `Wait`, and `Move`. Each action is parameterized by coordinates, text, or scalar values as appropriate. The `Click` primitive corresponds to a press event and includes normalized screen coordinates (x, y) , whereas `Release` denotes the corresponding button-up event and requires no positional arguments. This separation simplifies supervision and aligns with low-level GUI event semantics, where drag-and-drop gestures can be represented compositionally as a sequence of `Click` $\rightarrow$ `Move` $\rightarrow$ `Release`. Modeling these events as distinct atomic classes allows the IDM to predict a single action per transition while maintaining the flexibility to represent both short and prolonged clicks. Overall, this action space balances coverage and learnability, providing sufficient expressiveness for all interaction types required by OS-level benchmarks such as OSWorld and WindowsAgentArena.

Model architecture. The IDM takes as input two consecutive observations (O_t, O_{t+1}) and outputs the action a_t that caused the transition. We adopt a vision-only architecture consisting of a SigLIP-2 vision encoder [34] followed by four Transformer [35] layers. On top of this backbone, we attach three specialized prediction heads:

- **Action classification head:** a categorical predictor over six supported primitives: `click`, `release`, `scroll`, `type`, `wait`, and `move`.
- **Coordinate head:** for location-based actions (`click`, `move`, and `type`), the model predicts normalized (x, y) coordinates discretized into integers from 0 to 999. This formulation converts coordinate regression into a classification problem, which proved more stable during training.
- **Language head:** for text-entry actions, the model predicts the typed string autoregressively using a lightweight GPT-2 decoder [30] conditioned on the visual features of

(O_t, O_{t+1}) . This design enables variable-length text generation while maintaining grounding in the observed state transition.

`Scroll`, `wait`, and `release` actions require no additional arguments; the model simply predicts their occurrence.

Training. The IDM predicts three types of outputs depending on the action: (1) an action primitive, (2) discretized pixel coordinates (x, y) for location-based actions, and (3) token sequences for text-entry. Given a transition (O_t, O_{t+1}) , the model outputs a distribution over action primitives, optional coordinate distributions over 1000 bins per axis (integers 0–999), and an autoregressive distribution over text tokens. In practice, we discretize coordinates instead of regressing in continuous space, which stabilizes training and yields more accurate click and typing locations.

We train the model with a multi-task objective that activates only the losses relevant to the ground-truth action. Concretely, we apply a classification loss for the primitive, a coordinate loss for location-based actions, and a language-modeling loss for text-entry. Each head uses a standard cross-entropy term, and all losses are weighted equally unless otherwise specified. Full loss definitions are in the Appendix.

Evaluation. We evaluate the IDM on a held-out subset of the state-transition corpus containing 100 transitions per action type. This evaluation set spans diverse real-world web interactions collected using the same automated pipeline described earlier. We assess performance along two dimensions: *ActionType accuracy*, which measures the correctness of the predicted action primitive, and *Action accuracy*, which additionally includes the precision of the predicted coordinates and generated text. As shown in Section 4.4.1, the IDM trained solely on state-transition supervision achieves strong action classification accuracy, outperforming off-the-shelf foundation models and confirming its effectiveness as the labeling backbone of our framework.

3.2. Trajectory Data Generation from Videos

Once the IDM is trained, we retrieve suitable tutorial videos and use the IDM to convert them into UI trajectories.

Video retrieval. We build a retrieval framework that searches and downloads tutorial videos from large video platforms such as YouTube. The retrieval procedure differs depending on whether the goal is inference-time support or large-scale training data collection. *Inference-time retrieval.* Given a task description and the target application, we form a natural language search query. To refine the query, we prompt Gemini 2.5 Flash [10] with both the task instruction and the initial screen, asking it to generate a more specific query. We then use the YouTube Search API to retrieve the top 15 videos. For example, a task instruction "Can you increase the max volume of

Table 1. Distribution of collected videos across 69 applications in 7 main categories.

Category	# Apps	# Videos
Productivity	11	8,691
Programming	12	12,829
Design	9	7,948
Screen Editing	8	7,808
Audio Production	8	5,206
System Utilities	11	4,601
Science & Data	10	6,042
Total	69	53,125

the video to the 200% of the original volume in VLC?" becomes the search query "vlc increase max volume". Each retrieved video is paired with its title, which we treat as the candidate task description. *Training-time retrieval.* To construct a broad training dataset, we curate a list of 69 applications spanning productivity, programming, design, screen editing, audio production, system utilities, and science/data domains (Table 1). For each application, we prompt Gemini 2.5 Flash to generate plausible task queries and use them to search on video platforms, downloading the corresponding tutorial videos.

Filtering. Retrieved videos vary widely in quality and content, so we first sample frames at 1 frame per second and discard unusable segments. Segments that are not screen-casts (e.g., talking-head clips), are zoomed or cropped, or are blurred due to scene transitions are automatically removed by prompting Gemini 2.5 Flash. At inference time, we retain only the top 3 videos that pass this filter to reduce noise, whereas for training data collection we keep all qualified videos.

Trajectory labeling. After filtering, we segment each video into a sequence of frames $\{O_0, O_1, \dots\}$ and apply the IDM to every consecutive pair (O_t, O_{t+1}) , predicting the intermediate action a_t and assembling a trajectory $\tau = (O_0, a_0, O_1, a_1, \dots, O_T, a_T, O_{T+1})$. In this way, raw human demonstration videos are transformed into structured, executable trajectories without manual annotation. For inference-time usage, these trajectories are aligned with the task description and used as exemplars; for training-time usage, they are aggregated into a large corpus for supervised fine-tuning.

3.3. Applications of Trajectories

The extracted trajectories are utilized in two settings. For in-context learning (ICL), each trajectory is converted into a set of (*observation*, *action*) pairs augmented with concise natural-language reasoning traces generated by Gemini 2.5 Flash. These (*observation*, *action*, *reasoning*) exemplars are formatted as few-shot demonstrations in the model context, enabling general-purpose agents to leverage age planning and grounding priors without additional train-

ing. For supervised fine-tuning (SFT), the labeled trajectories are aggregated into a large corpus of (*state*, *action*) sequences and used to optimize multimodal LLMs with a standard sequence modeling objective. We fine-tune UI-TARS-1.5 [29], a specialized computer-using agent, and Qwen 2.5-VL [5], a general multimodal model, demonstrating that the proposed data serve as a scalable and transferable supervision source across diverse models.

4. Experiments and Analysis

4.1. Experimental Setup

Our experiments evaluate whether IDM-labeled trajectories improve interactive computer-using agents in two settings: (1) *in-context learning (ICL)* for general-purpose multimodal models and agentic frameworks, and (2) *supervised fine-tuning (SFT)* for open-source models. We also compare the resulting trajectories against TongUI [43], an existing automatic labeling pipeline.

All agents operate under a unified interface where each step provides only the current screen image and the history of executed actions. In the ICL setting, each model receives a fixed number of task-relevant trajectories retrieved from our video-derived dataset as exemplars. In the SFT setting, open-source CUAs are trained directly on the same IDM-labeled trajectories.

We evaluate on two complementary interactive computer-use benchmarks. OSWorld-Verified [38] provides full desktop and operating-system (Ubuntu) environments across productivity, programming, design, and system utilities, and we apply a 50-step interaction limit. To assess robustness across different interface conventions and OS-specific workflows, we additionally evaluate on WindowsAgentArena [7], a benchmark of Windows tasks that follows a 15-step interaction limit. Together, these benchmarks allow us to test whether the benefits of IDM-labeled trajectories generalize across heterogeneous operating systems and UI distributions.

4.2. Datasets

State-transition corpus. To train the IDM, we gather autonomous state transitions from real web interactions, resulting in over 600k (O_t, a_t, O_{t+1}) triples.

Video-derived trajectories. After training the IDM, we apply it to a curated collection of 53,125 instructional YouTube videos, producing one trajectory per video and yielding 53,125 trajectories across 69 applications spanning productivity, programming, design, audio production, system utilities, and scientific/data domains (Table 1). Additional details are provided in the Appendix.

Baseline labeler. TongUI [43] generates action annotations by prompting the UI-TARS-1.5-7B agent [29], while our method applies the trained IDM to the same source videos.

Table 2. **Main results on OSWorld-Verified [38]** under the 50-step evaluation limit. IDM-labeled video exemplars from W&L improve both general multimodal models (Gemini, Claude, o3, Jedi) via ICL and open-source CUAs via SFT, achieving up to +11.1 absolute gains over TongUI-labeled and baseline models.

Category	Base Model	Method	Success Rate (%)
In-Context Learning			
General Models	Gemini 2.5 Flash [10]	Base (w/o video)	19.0
		w/ W&L labeled videos	22.0 (+3.0)
	OpenAI o3 [24]	Base (w/o video)	21.8
		w/ TongUI labeled videos	21.1 (-0.7)
		w/ W&L labeled videos	24.3 (+2.5)
	Claude 4 Sonnet [4]	Base (w/o video)	43.9
w/ TongUI labeled videos		43.4 (-0.5)	
w/ W&L labeled videos		45.5 (+1.6)	
Agentic Framework	Jedi [39]	Base (w/o video)	50.6
		w/ W&L labeled videos	52.8 (+2.2)
Supervised Fine-Tuning			
Open-Source Models	Qwen 2.5VL 7B [5]	Base (No SFT)	1.9
		SFT w/ TongUI labeled	5.4 (+3.5)
		SFT w/ W&L (IDM labeled)	13.0 (+11.1)
	UI-TARS-1.5-7B [29]	Base (No SFT)	27.3
		SFT w/ TongUI labeled	23.8 (-3.5)
		SFT w/ W&L (IDM labeled)	31.1 (+3.8)

Table 3. **WindowsAgentArena results under the 15-step evaluation limit.** Comparison of open-source 7B-sized CUAs. W&L achieves the highest success rate among 7B-sized models, surpassing both prior automatic labeling methods and zero-shot baselines.

Model	Setting / Training Data	Success Rate (%)
UI-TARS-1.5-7B [29]	Zero-shot (no SFT)	18.1
	SFT w/ TongUI [43]	12.9 (-5.2)
	SFT w/ W&L (IDM-labeled)	24.0 (+5.9)
OpenCUA-7B [36]	Base	13.5
UltraCUA-7B [42]	Base	21.7

To ensure a fair comparison focused only on labeling quality, we use the identical set of tutorial videos retrieved by our framework and apply TongUI’s labeling pipeline, differing only in the model used to annotating actions.

4.3. Models

We evaluate three categories of models covering general multimodal model, agentic framework, and open-source CUAs.

General-purpose multimodal models. Gemini 2.5 Flash [10], OpenAI o3 [24], and Claude 4 Sonnet [4] are evaluated in the ICL setting using retrieved IDM-labeled trajectories as exemplars.

Agentic framework. We use Jedi [39], a vision-only agentic framework for OSWorld that combines an MLLM planner (OpenAI o3) with the Jedi-7B grounding model that maps natural language steps to executable UI actions. Its

modular design allows straightforward integration of our ICL exemplars, so we evaluate Jedi both with and without retrieved IDM-labeled trajectories.

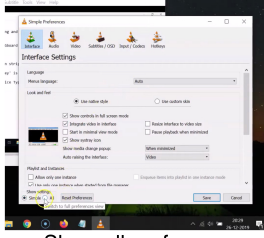
Open-source CUAs. UI-TARS-1.5-7B [29] and Qwen 2.5-VL 7B [5] are trained with supervised fine-tuning on our 53,125 video-derived trajectories, enabling a controlled comparison against TongUI-labeled data and the exemplar-based performance of multimodal models.

4.4. Main Results

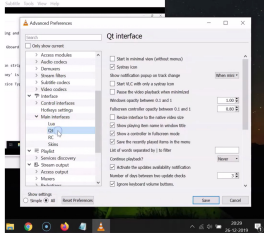
Table 2 summarizes our main results on OSWorld across both in-context learning and supervised fine-tuning. We observe consistent improvements across all model categories. For general-purpose multimodal models (Gemini 2.5 Flash, OpenAI o3, Claude 4 Sonnet), adding our W&L exemplars improves performance by +1.6 to +3.0 points. This shows that trajectories distilled from web tutorials provide useful domain-specific priors that even strong foundation models can leverage at inference time. For the Jedi agentic framework [39], which couples the o3 planner with Jedi-7B grounding model, W&L yields a +2.2 point gain, demonstrating that our trajectories can complement structured planning pipelines by enriching them with exemplars that support both planning and grounding. For open-source CUAs, supervised fine-tuning on our 53k video-derived trajectories yields even larger gains. UI-TARS-7B improves by +3.8 points, while Qwen 2.5-VL sees the largest im-

Task: Current max volume is 125%. Can you increase it to 200% in VLC?

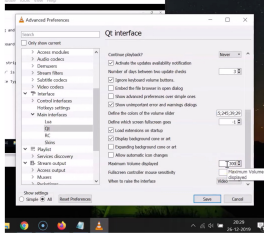
Retrieved & IDM Labeled Video



Show all preferences
Click (931, 638)

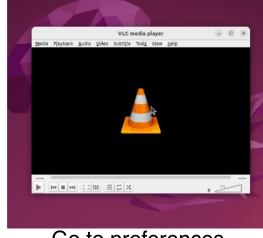


Go to QT section
Click (578, 695)

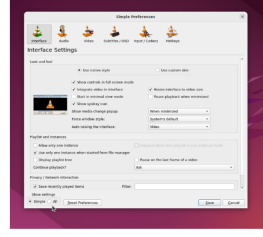


Type 300 for Maximum Volume
Type (948, 890, "300")

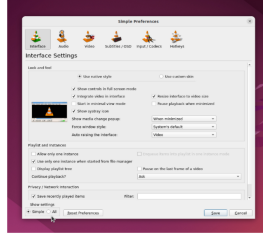
OpenAI o3



Go to preferences
Click (330, 574)

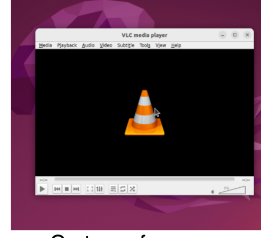


Show all preferences
Click (867, 411)

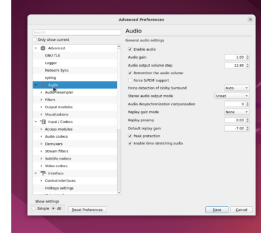


Show all preferences
Click (867, 411)

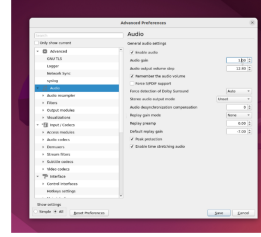
Jedi (o3)



Go to preferences
Click (334, 575)

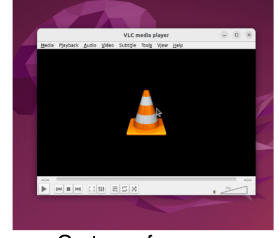


Expand Audio option
Click (388, 392)

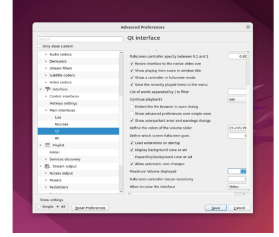


Modify Audio gain
Type (264, 622, "2.00")

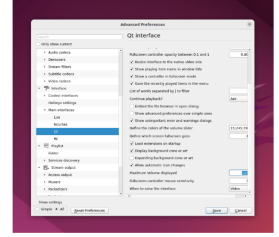
ICL w/ Watch & Learn



Go to preferences
Click (332, 576)



Expand QT option
Click (346, 622)



Type 200 for Maximum Volume
Type (721, 833, "200")

✗ Grounding Error

✗ Planning Error

✓ Task Success

Figure 3. **Qualitative examples on OSWorld.** On the left, the video-derived trajectory that W&L generates for the task. On the right: (i) the o3 agent makes a grounding error by selecting a wrong UI element; (ii) the Jedi (o3) agent makes a planning error by entering the wrong submenu without recovering; (iii) using the video-derived trajectory, W&L agent completes the task successfully. Images are cropped for visibility, and the action coordinates correspond to the original full-resolution screenshots. More trajectory examples are in Appendix.

provement, from 1.9 to 13.0 (+11.1). This larger jump is expected because Qwen is a general-purpose multimodal model not originally trained for computer use, so it benefits disproportionately from our dataset, which provides task-specific supervision that was previously missing.

We further evaluate on WindowsAgentArena [7], which focuses on Windows OS tasks under a standardized 15-step evaluation limit. As shown in Table 3, W&L achieves the state-of-the-art success rate among 7B-scale models with 15 step limit, surpassing both prior automatic labeling method and specialized CUA baselines. Specifically, fine-tuning UI-TARS-1.5-7B on IDM-labeled trajectories improves performance from 18.1 to 24.0 (+5.9), whereas training on TongUI-labeled data leads to a degradation in performance (12.9, -5.2). These results confirm that IDM-based

labeling produces more accurate and consistent supervision, translating to stronger downstream generalization.

Overall, across both OSWorld and WindowsAgentArena, W&L serves as a scalable and high-quality supervision signal that consistently enhances models and agentic frameworks in both in-context and fine-tuning settings.

4.4.1. Effect of Label Accuracy on Performance

Action label accuracy is central to training CUAs: noisy annotations not only fail to help but can actively degrade performance. We first compare our dedicated IDM against Gemini 2.5 Flash and the TongUI labeling pipeline (based on UI-TARS-1.5-7B) on the held-out state-transition test set (Table 4). Our IDM achieves the strongest results, substantially outperforming both baselines. TongUI offers modest

gains over Gemini, especially for structured actions such as `scroll` and `click`, but still falls short of our IDM.

These differences in labeling accuracy directly translate into downstream performance. TongUI, although it follows the same prompt format, relies on labels generated by UI-TARS-1.5-7B, whose own performance is notably weaker on WindowsAgentArena (Table 3) than on Ubuntu-based OSWorld (Table 2). As a result, labeling quality degrades more severely in the Windows setting, leading to noisier supervision. As shown in Table 3, this effect is amplified on WindowsAgentArena, where fine-tuning with TongUI-labeled data causes a drop in success rate (a decrease of 5.2 points) compared with consistent gains from IDM-labeled trajectories. In contrast, our IDM-derived labels remain reliable across both Ubuntu-style (OSWorld) and Windows-style (WindowsAgentArena) interfaces, underscoring that accurate and platform-robust supervision is essential for effective action grounding and cross-OS generalization.

4.4.2. Effect of Trajectories on In-Context Learning

We next analyze the contribution of accurate video labeling to in-context learning (ICL). Our framework provides structured action annotations and natural language reasoning for each step. To isolate the effect of each component, we compare three variants: (i) consecutive frames only, (ii) frames paired with predicted actions, and (iii) frames with both actions and reasoning generated by Gemini 2.5 Flash.

Ablations on OSWorld (Table 5) show that adding action labels provides a substantial boost over using frames alone, and further gains are achieved when natural language reasoning is included. This pattern holds consistently across all tested models. Figure 3 illustrates these effects qualitatively: baseline agents without labeled trajectories often make grounding or planning errors, for example clicking incorrect UI elements or entering the wrong submenu, while the W&L-enhanced agent successfully completes the same tasks by leveraging accurate video-derived exemplars. These observations confirm that labeled trajectories convey procedural and causal knowledge beyond visual context, improving both planning and grounding for complex workflows.

4.4.3. Effect of Video Retrieval on In-Context Learning

We further examine the role of retrieval quality by comparing our method against a random retrieval baseline using o3 (Table 6). Interestingly, random retrieval neither improves nor degrades performance relative to the base model. This suggests that, while carefully retrieved exemplars provide useful signal, even randomly selected exemplars do not introduce significant noise. A likely explanation is that the action labels themselves remain highly accurate regardless of retrieval quality, ensuring that the model is not misled by contradictory supervision. These results indicate that the main benefit of our method lies in providing *targeted* ex-

Table 4. **Action labeling accuracy on the held-out test set.** We evaluate six primitive action types with 100 transitions each. *ActionType Accuracy* reflects correct action classification, while *Action Accuracy* also checks argument correctness. W&L’s IDM outperforms Gemini and TongUI across all action types.

ActionType	Gemini 2.5 Flash	TongUI	W&L IDM
click(x, y)	68%	72%	95%
release	71%	67%	90%
scroll(scroll.y)	55%	75%	93%
type(text)	77%	71%	86%
wait(500ms)	92%	88%	97%
move(x, y)	65%	61%	89%
ActionType Accuracy	81.5%	84.3%	95.8%
Action Accuracy	70.5%	72.3%	91.7%

Table 5. **Ablation of action labeling and reasoning** within ICL exemplars on OSWorld. Structured trajectories lead to steady gains compared with raw frame exemplars.

	Gemini 2.5 Flash	OpenAI o3	Claude 4 Sonnet
Baseline (no exemplars)	19.0	21.8	43.9
+ Frames	18.4	21.8	43.9
+ Frames + Actions	20.1	23.0	44.4
+ Frames + Actions + Reasoning	22.0	24.3	45.5

Table 6. **Ablation of retrieval strategy for ICL on OSWorld.** Random retrieval gives minimal gains, while **task-relevant retrieval** in W&L improves performance.

	o3 (base)	o3 + Random	o3 + W&L
ICL	21.8	21.8	24.3 (+2.5)

emplars that align closely with the task context. Retrieval quality therefore determines the strength of the positive effect, but poor retrieval does not actively harm performance when the underlying labels are still correct.

5. Conclusion and Future Work

We introduced W&L, a framework that converts web-scale human demonstration videos into executable UI trajectories using a vision-only inverse dynamics model (IDM) and a task-aware video retrieval pipeline. With over 53k automatically labeled trajectories, W&L improves both in-context learning and supervised fine-tuning across diverse agents, benefiting general-purpose multimodal models as well as open-source CUAs. Experiments on OSWorld and WindowsAgentArena confirm that IDM-labeled trajectories provide high-quality supervision that generalizes across operating systems and interface conventions.

Future work will focus on combining or segmenting tutorials to better capture long-horizon workflows and on exploring reinforcement learning using our trajectories as demonstrations, replay buffers, or priors for reward modeling. These directions aim to further bridge large-scale demonstrations with adaptive policy learning, bringing CUAs closer to robust real-world deployment.

Acknowledgments

We thank the Google Cloud AI Research team for their valuable feedback, with special thanks to Yale Song for his constructive input. This work was supported in part by the National Science Foundation under NSF CAREER Award #2443149 and by an Alfred P. Sloan Foundation Fellowship.

References

- [1] Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zahoor Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966, 2024. 3
- [2] 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 *International Conference on Learning Representations (ICLR)*, 2025. 1, 3
- [3] Saaket Agashe, Kyle Wong, Vincent Tu, Jiachen Yang, Ang Li, and Xin Eric Wang. Agent s2: A compositional generalist-specialist framework for computer use agents. In *Conference on Language Modeling (CoLM)*, 2025. 1, 3
- [4] Anthropic. Claude opus 4 & claude sonnet 4 system card. Technical report, Anthropic, 2025. 6
- [5] 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 preprint arXiv:2502.13923*, 2025. 5, 6
- [6] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampeiro, and Jeff Clune. Video pretraining (VPT): Learning to act by watching unlabeled online videos. In *Advances in Neural Information Processing Systems*, 2022. 2
- [7] Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Buckner, Lawrence Jang, and Zack Hui. Windows agent arena: Evaluating multi-modal os agents at scale. Technical report, Microsoft Research, 2024. 2, 5, 7
- [8] 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 *NeurIPS*, 2023. 2
- [9] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022. 3
- [10] Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. 4, 6
- [11] Gonzalo Gonzalez-Pumariaga, Vincent Tu, Chih-Lun Lee, Jiachen Yang, Ang Li, and Xin Eric Wang. The unreasonable effectiveness of scaling agents for computer use, 2025. 1, 3
- [12] 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 *The Thirteenth International Conference on Learning Representations*, 2025. 1
- [13] Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. Is your LLM secretly a world model of the internet? model-based planning for web agents. *CoRR*, abs/2411.06559, 2025. 4
- [14] Shivanshu Gupta, Sameer Singh, Ashish Sabharwal, Tushar Khot, and Ben Bogin. Leveraging in-context learning for language model agents. *arXiv preprint arXiv:2506.13109*, 2025. 3
- [15] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Hongming Zhang, Tianqing Fang, Zhenzhong Lan, and Dong Yu. OpenWebVoyager: Building multimodal web agents via iterative real-world exploration, feedback and optimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27545–27564, Vienna, Austria, 2025. Association for Computational Linguistics. 2
- [16] Samuel Holt, Max Ruiz Luyten, Thomas Pouplin, and Michaela van der Schaar. Improving llm agent planning with in-context learning via atomic fact augmentation and lookahead search. *arXiv preprint arXiv:2506.09171*, 2025. 3
- [17] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Björck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705v2*, 2025. 2
- [18] Yunseok Jang, Yeda Song, Sungryull Sohn, Lajanugen Logeswaran, Tiange Luo, Dong-Ki Kim, Kyunghoon Bae, and Honglak Lee. Scalable Video-to-Dataset Generation for Cross-Platform Mobile Agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2
- [19] Jihyung Kil, Chan Hee Song, Boyuan Zheng, Xiang Deng, Yu Su, and Wei-Lun Chao. Dual-view visual contextualization for web navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14445–14454, 2024. 1
- [20] Wei Li, William E Bishop, Alice Li, Christopher Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on UI control agents. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. 2
- [21] Xing Han Lù, Zdeněk Kasner, and Siva Reddy. WeBLINX: Real-world website navigation with multi-turn dialogue, 2024. 2
- [22] Shikhar Murty, Christopher Manning, Peter Shaw, Mandar Joshi, and Kenton Lee. BAGEL: Bootstrapping agents by guiding exploration with language. 2024. 2
- [23] Shikhar Murty, Hao Zhu, Dzmitry Bahdanau, and Christopher D Manning. Nnetnav: Unsupervised learning of browser agents through environment interaction in the wild. *arXiv preprint arXiv:2410.02907*, 2025. 2

- [24] OpenAI. Openai o3 and o4-mini system card. Technical report, OpenAI, 2025. 6
- [25] OpenAI. Introducing operator. <https://openai.com/index/introducing-operator/>, 2025. Accessed: April 12, 2025. 1
- [26] Tianyue Ou, Frank F. Xu, Aman Madaan, Jiarui Liu, Robert Lo, Abishek Sridhar, Sudipta Sengupta, Dan Roth, Graham Neubig, and Shuyan Zhou. Synatra: Turning indirect knowledge into direct demonstrations for digital agents at scale. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2
- [27] Vardaan Pahuja, Yadong Lu, Corby Rosset, Boyu Gou, Arindam Mitra, Spencer Whitehead, Yu Su, and Ahmed Hassan Awadallah. Explorer: Scaling exploration-driven web trajectory synthesis for multimodal web agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6300–6323, Vienna, Austria, 2025. Association for Computational Linguistics. 2
- [28] Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiada Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. 2
- [29] Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025. 1, 5, 6
- [30] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 4, 16
- [31] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy P Lillicrap. AndroidInTheWild: A large-scale dataset for android device control. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 2
- [32] Hongjin Su, Ruoxi Sun, Jinsung Yoon, Pengcheng Yin, Tao Yu, and Sercan Ö Arık. Learn-by-interact: A data-centric framework for self-adaptive agents in realistic environments. *arXiv preprint arXiv:2501.10893*, 2025. 3
- [33] Qiushi Sun, Kanzhi Cheng, Zichen Ding, Chuanyang Jin, Yian Wang, Fangzhi Xu, Zhenyu Wu, Chengyou Jia, Liheng Chen, Zhoumianze Liu, Ben Kao, Guohao Li, Junxian He, Yu Qiao, and Zhiyong Wu. OS-genesis: Automating GUI agent trajectory construction via reverse task synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5555–5579, Vienna, Austria, 2025. Association for Computational Linguistics. 2
- [34] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 4, 15
- [35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4
- [36] 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, Zheng Boyuan, LI PEIHANG, Fangyu Lei, Ruisheng Cao, Yeqiao Fu, Dongchan Shin, Martin Shin, Hu Jiarui, Yuyan Wang, Jixuan Chen, Yuxiao Ye, Danyang Zhang, Yipu Wang, Heng Wang, Diyi Yang, Victor Zhong, Y.Charles, Zhilin Yang, and Tao Yu. OpenCUA: Open foundations for computer-use agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 1, 6
- [37] Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent Workflow Memory. 2024. 3
- [38] 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 *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. 2, 5, 6
- [39] Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, Yiheng Xu, Junli Wang, Doyen Sahoo, Tao Yu, and Caiming Xiong. Scaling Computer-Use Grounding via User Interface Decomposition and Synthesis. *arXiv preprint arXiv:2505.13227*, 2025. 1, 6
- [40] Yiheng Xu, Dunjie Lu, Zhennan Shen, Junli Wang, Zekun Wang, Yuchen Mao, Caiming Xiong, and Tao Yu. Agenttrek: Agent trajectory synthesis via guiding replay with web tutorials. In *The Thirteenth International Conference on Learning Representations*, 2025. 2
- [41] Chenyu Yang, Shiqian Su, Shi Liu, Xuan Dong, Yue Yu, Weijie Su, Xuehui Wang, Zhaoyang Liu, Jinguo Zhu, Hao Li, Wenhai Wang, Yu Qiao, Xizhou Zhu, and Jifeng Dai. Zerogui: Automating online gui learning at zero human cost, 2025. 2
- [42] Yuhao Yang, Zhen Yang, Zi-Yi Dou, Anh Nguyen, Keen You, Omar Attia, Andrew Szot, Michael Feng, Ram Ramakrishna, Alexander Toshev, Chao Huang, Yinfei Yang, and Zhe Gan. Ultracua: A foundation model for computer use agents with hybrid action, 2025. 1, 6
- [43] Bofei Zhang, Zirui Shang, Zhi Gao, Wang Zhang, Rui Xie, Xiaojian Ma, Tao Yuan, Xinxiao Wu, Song-Chun Zhu, and Qing Li. Tongui: Building generalized gui agents by learning from multimodal web tutorials. *arXiv preprint arXiv:2504.12679*, 2025. 2, 5, 6
- [44] Xinran Zhao, Hanie Sedghi, Bernd Bohnet, Dale Schuurmans, and Azade Nova. Improving large language model planning with action sequence similarity. *arXiv preprint arXiv:2505.01009*, 2025. 3
- [45] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v(ision) is a generalist web agent, if grounded. In *Forty-first International Conference on Machine Learning*, 2024. 1

- [46] Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. Synapse: Trajectory-as-exemplar prompting with memory for computer control. 2024. [3](#)