MMADP: UNIFIED MULTIMODAL AGENT TRAJECTORIES FOR OPEN AGENTIC TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Many open-weight models learn to use tools through agentic midtraining, but the datasets used at this stage are rarely released. Open datasets could support research on this stage. Existing efforts to combine them focus on text-only trajectories or individual domains. We introduce the Multimodal Agent Data Protocol (MMADP) to support multimodal agent training across coding, web, and computer-use tasks. MMADP builds on the Agent Data Protocol (ADP) and adds support for screenshots and annotations of screen elements. Using MMADP, we combine 16 datasets into a corpus with an estimated 14.7M trajectories and 366B tokens, the largest open corpus of agentic trajectories to our knowledge. Our experiments show that a small sample of this corpus can teach tool use across domains. We perform supervised fine-tuning of InternVL2.5-MPO on a 250M-token sample, starting from models with no prior tool-calling training. Both the 8B and 26B models improve on coding, web, and computer-use benchmarks. At 8B, the absolute gains are +3.0% on SWE-bench Verified, +18.2% on MiniWoB++, and +6.0% on AndroidWorld. We also find that mixing domains helps within each domain: with 250M tokens sampled for each model, the mixed-domain 8B model outperforms single-domain models on all three benchmarks. For each domain, the mixed-domain model uses only about one-third of the training tokens used by the corresponding single-domain model. We release the schema, converters, and full corpus to support open research on multimodal agentic midtraining.

1 INTRODUCTION

Recent models, including GLM-4.5, GLM-5, and Qwen3-Coder-Next, incorporate agentic trajectories into midtraining (GLM-4.5 Team et al., 2025; GLM-5 Team et al., 2026; Cao et al., 2026). By training on these sequences of actions and observations, models learn to use tools to complete tasks. Reinforcement learning (RL) alone can fail to teach reliable tool use, making supervised training on trajectories a useful starting point (Hong et al., 2026; Wei et al., 2025). The datasets used in these midtraining efforts are not publicly available. Open access to data like this would greatly help researchers study which trajectories teach useful agent behaviors and how to combine data from different domains, as open data and ablations have done for vision-language models (Deitke et al., 2025).

Although these datasets are not public, researchers can still study agentic midtraining with public trajectories, such as SWE-smith, AgentNet, and Mind2Web (Yang et al., 2025b; Wang et al., 2025d; Deng et al., 2023). Public datasets span coding, web, and computer use. Each uses its own format for tool calls and observations (see Appendix A). The Agent Data Protocol (ADP) combines 13 text-only datasets in a common format for training (Song et al., 2026). Other open corpora combine trajectories from different GUI platforms but contain no coding-agent data (Section 2.2).

Training on coding and GUI trajectories together requires a format that supports both text and image observations. For example, a recorded click on "button 3" must be paired with the screenshot and annotations that identify that button.

We therefore introduce the Multimodal Agent Data Protocol (MMADP), a shared format for joint training on coding and GUI trajectories. ADP records each trajectory as a sequence of agent actions and observations. MMADP extends this format to include screenshots and annotations of screen elements. Existing ADP trajectories can be reused in MMADP without modification. Figure 1

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

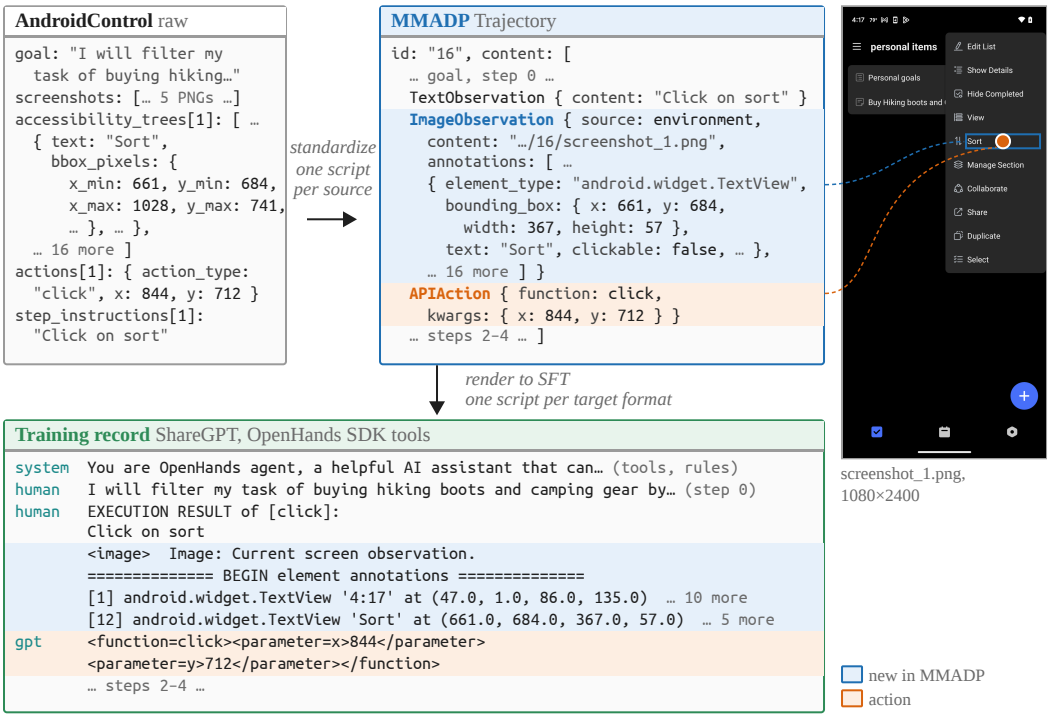


Figure 1: An AndroidControl (Li et al., 2024) step in its raw release, as an MMADP trajectory, and as a training record. Fields that MMADP adds to ADP are marked in blue, and agent actions in orange.

shows how we convert a screenshot and click action from AndroidControl into the MMADP format and then into a training record. Each dataset needs one converter to MMADP, and each target agent framework needs one converter from MMADP to its training format. Each converter also runs the quality checks its dataset needs. These checks find and fix a source's defects once, at conversion. New datasets can therefore use the existing training converters, and new frameworks can use the existing data.

Using MMADP, we combine 16 datasets into a corpus spanning coding, web, computer use (desktop and mobile), and general tasks. The corpus contains around 14.7M trajectories and 366B tokens (Table 1). To our knowledge, it is the largest open corpus of agentic trajectories and the first to combine coding-agent trajectories with web and computer-use trajectories in one schema.

We evaluate the corpus by fine-tuning InternVL2.5-MPO (Chen et al., 2024; Wang et al., 2024a) at 8B and 26B on a 250M-token sample containing fewer than 20,000 training records. Neither model has prior tool-calling training. After supervised fine-tuning (SFT), both models solve more tasks on SWE-bench Verified (Jimenez et al., 2024; Chowdhury et al., 2024), MiniWoB++ (Liu et al., 2018), and AndroidWorld (Rawles et al., 2025) (Section 5.1). On SWE-bench Verified, the share of valid tool calls from the 8B model rises from 4.0% to 99.99%. Models trained on one domain learn the same format and still score near zero outside it (Section 5.2). These experiments use SFT; a full midtraining run remains future work.

The corpus also lets us test whether data from other domains helps a model on its target domain. Prior work reports both gains from mixing data and negative transfer between tasks (Song et al., 2026; Zeng et al., 2024; Song et al., 2024; Yang et al., 2025a). We compare mixed-domain and single-domain training at 8B, sampling 250M tokens for each model. The mixed-domain model outperforms each single-domain model on that domain's benchmark (Section 5.2). For each domain, it uses only about one-third of the training tokens used by the corresponding single-domain model.

Lastly, we release the schema, converters, and full corpus publicly.

2 Related Work

2.1 Agentic Midtraining

Midtraining is a training stage that sits between general pretraining and specialized post-training (Mo et al., 2025; Tu et al., 2025). GLM-4.5, GLM-5, and Qwen3-Coder-Next introduce agentic trajectories in this stage and credit it with improved tool use and long-context capability (GLM-4.5 Team et al., 2025; GLM-5 Team et al., 2026; Cao et al., 2026). Controlled ablations show that the stage helps downstream agents: with the SFT recipe fixed, agentic continual pretraining raises agent benchmark scores (Su et al., 2026), tool-use data outperforms a generic corpus at the same budget (Jiang et al., 2026), and synthesized GUI data improves GUI agents (Wu et al., 2026).

Reinforcement learning is the other common route to agentic capability and also raises GUI-agent scores (Lu et al., 2025a; Wang et al., 2025a). In interactive multimodal environments, RL faces sparse or delayed rewards, costly rollouts, and fragile environments (Lu et al., 2025a; Wang et al., 2025a). RL alone also does not reliably induce tool use (Wang et al., 2025b): DeepEyesV2 finds that direct RL fails to establish robust tool-use behavior and adds a cold-start stage on trajectories (Hong et al., 2026), and WebAgent-R1 finds that RL without behavior cloning does not improve much over the initial model (Wei et al., 2025).

Text trajectories are relatively abundant. Most multimodal trajectory data is unreleased. UI-TARS, GUI-Owl-1.5, UI-Venus-2, Qwen3-VL, and GLM-4.5V each describe a GUI trajectory pipeline but release only the model (Qin et al., 2025; Xu et al., 2026; Venus Team et al., 2026; Bai et al., 2025; GLM-V Team et al., 2025). The few reports that do release trajectories describe other GUI training data as proprietary (Wang et al., 2025d; Liu et al., 2026). Unlike text or code corpora, screen trajectories are scarce and costly to collect (Liu et al., 2026; Wang et al., 2025a).

2.2 Open Agentic Corpora

ADP argues that the bottleneck for text agents is not a lack of data but its fragmentation across formats, tools, and interfaces (Song et al., 2026). In multimodal data, action spaces, observation formats, and platform coverage also differ between datasets (Xu et al., 2025; Zhang et al., 2025a; Yang et al., 2026). GUI trajectory sets such as AgentNet (Wang et al., 2025d) and ScaleCUA-Data (Liu et al., 2026) each cover different platforms in their own action spaces, while grounding sets such as OS-Atlas (Wu et al., 2025) hold single steps. Coding-agent trajectories are text-only and likewise differ in action space (Yang et al., 2025b; Pan et al., 2025). We list several other open releases in Appendix A for reference. Several efforts unify this data. None of them covers multiple domains with screenshots. ADP, AgentOhana, and AgentBank unify text-only trajectories (Song et al., 2026; Zhang et al., 2024a; Song et al., 2024), and ADP in particular introduces the schema and pipeline that we extend. Aguvis, OS-Atlas, and ScaleCUA define one action space across GUI platforms but cover GUI tasks only (Xu et al., 2025; Wu et al., 2025; Liu et al., 2026). Orchard releases coding and web trajectories under separate schemas for separate models (Peng et al., 2026). To our knowledge, no open corpus joins coding-agent trajectories with web and computer-use trajectories in one schema.

2.3 Mixing Domains

A mixed-domain corpus generally outperforms a single-domain one (Song et al., 2026), and adding general instruction data to agent trajectories improves held-out agent tasks (Zeng et al., 2024). Transfer between domains is less settled. GUIMid finds that midtraining on math and code data improves mobile device use (Zhang et al., 2025b). AgentBank, in contrast, finds that fine-tuning on embodied trajectories yields lower held-out reasoning, math, programming, and web scores than fine-tuning on generalist instruction and code data (Song et al., 2024). Neither compares domains at a fixed token budget with screenshot trajectories as we do in Section 5.2.

3 Multimodal Agent Data Protocol

MMADP has two parts: a representation format that extends the Agent Data Protocol (ADP) to screen-based trajectories, and a corpus of 16 datasets converted into that format. We use the term

corpus for the full output of the conversion pipeline; the smaller *training data* we train on is defined in Section 3.6.

3.1 Background: Agent Data Protocol

The Agent Data Protocol (Song et al., 2026) represents agentic trajectories with `Trajectory` objects, each of which contains an `id`, an ordered list of actions and observations (`content`), and metadata. The three action types are API actions (`APIAction`), representing function calls and tool use; code actions (`CodeAction`), representing the output and immediate execution of a code fragment; and message actions (`MessageAction`), representing natural language outputs such as responses to the user. ADP has two observation types: text observations (`TextObservation`), which represent any text, and web observations (`WebObservation`), which represent the state of a web page. Web observations in ADP include an image field that no ADP v1 dataset populates.

To standardize trajectories, ADP requires each dataset to define an `api.py`, a file that declares the functions that its API actions can call. A three-stage pipeline then converts the raw datasets to the standard format, converts the standard format to the training format of a specific agent, and runs quality checks on the resulting files. ADP v1 targeted three agent formats: OpenHands (Wang et al., 2025c), AgentLab (Le Sellier de Chezelles et al., 2025), and SWE-agent (Yang et al., 2024).

The ADP v1 release consists of 1.3M trajectories from 13 text-only datasets, primarily in the coding and web-browsing domains.

3.2 Challenges

Two properties of the sources shaped the design in Sections 3.3 and 3.5:

Data verification. Many datasets carry errors that are undocumented by their releases. Examples are undocumented placeholder values and links to live websites that have since changed. An independent re-annotation of AITW (Rawles et al., 2023) kept 5,147 of 7,180 sampled episodes (Zhang et al., 2024b). Datasets differ so much in structure and composition that general scripts cannot find these errors. A model trained on these errors can learn them, for example, clicks on elements that the observation does not contain. Examples in our sources include WebLINX (Lù et al., 2024) actions that target elements with no identifying attribute (1.7% of its actions) and trajectories that claim success without performing any action.

Screen representation. Datasets differ in what the agent sees of the screen and in how actions point at it (Xu et al., 2025; Zhang et al., 2025a). Datasets represent the screen as pixels, element boxes with OCR text or interactivity flags, accessibility trees, or HTML, and point actions at it by element ID, element index, absolute pixel coordinates, or normalized coordinates. The two are coupled because each action points at the observation before it. A trajectory therefore cannot be reformatted without also changing its tool usage and its observations together.

3.3 MMADP Format

We complete ADP's `ImageObservation` schema, add fields for element annotations, and add screenshot support to `WebObservations` (Figure 1). We keep the actions unchanged. This design maintains compatibility with ADP: datasets already converted to ADP remain valid MMADP.

To support the variety of data sources, we extend the schema as follows:

Image observations represent arbitrary images, which are most often screenshots of a desktop, a mobile device, or a browser viewport. Each observation includes: (1) content: a path to the screenshot, (2) annotations: an optional list of element annotations, and (3) source: whether the image came from the user, the agent, or the environment. Image observations are used standalone by non-web datasets and nested inside web observations for web datasets.

Element annotations describe the elements in a screenshot. These elements are usually interactive. Each annotation includes: (1) element type, (2) bounding box: position and size in the image, (3) text: the element's OCR text, if any, (4) content description: a label, resource identifier, or hint for elements without text, and (5) clickable and editable: two interactivity flags. Every field except the type and box is optional because sources record different subsets of them.

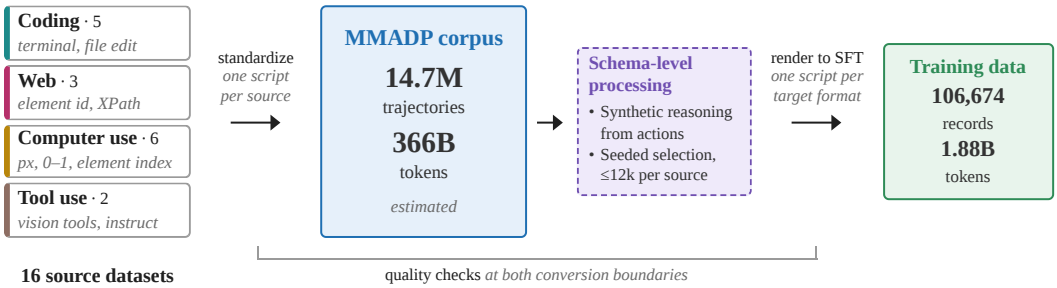


Figure 2: The MMADP pipeline. Sources from multiple domains convert to a unified schema through one converter each. Schema-level processing and a seeded selection render trajectories into the hosted training data. Quality checks run at both conversion boundaries (Section 3.5).

Web observations keep the five ADP fields (HTML, accessibility tree, URL, viewport size, and image observation) and add (6) last action error: the browser's error message from the previous step, if any. The last action error field keeps the browser's error message inside the next observation. Observations and actions still alternate strictly. We also populate the image observation slot, which was previously unused.

Actions are unchanged from ADP: message, code, and API actions, with each dataset declaring its API surface in its `api.py`. Actions carry their target in the API action's arguments, in the format of their dataset. Each target form from Section 3.2 resolves against the observation before the action: element and set-of-marks (Yang et al., 2023) IDs and XPaths against its accessibility tree or HTML, element indices against its image annotations, and coordinates against its screenshot.

3.4 Standardization Pipeline

To convert the raw datasets into a standard format, we use a pipeline derived from the Agent Data Protocol. This updated pipeline has the following stages, which run for each dataset (Figure 2):

1. **Raw to Standardized.** This stage follows ADP's design: one converter per dataset maps the raw release onto the standard schema, now including image observations and their annotations. At the end of this stage, a typical web-browsing trajectory may look like alternating `WebObservation` and `APIAction` objects, with API tool calls like `click(bid="3")`. Exact tool calls remain defined by the dataset, not by our pipeline.
2. **Standardized to SFT.** Also from ADP, this stage maps the standard schema to the training format of a specific harness. Each target format requires only a single script; we describe ours in Section 3.6. A deterministic selection sub-stage then draws the training data from the converted records (Section 3.6).
3. **Quality Assurance.** We expand the quality assurance (QA) stage of ADP (Song et al., 2026) with checks at both conversion boundaries, described with the corpus in Section 3.5.

Appendix B details each stage.

This design decreases the engineering needed to extend the corpus. Adding WildGUI (Xiong et al., 2026), a 12.7M-trajectory source, and ScaleCUA, which spans five platforms, took one converter each: 418 and 532 lines, respectively. Other conversion script lengths can be found in Table 1. Because part of the QA stage runs on standardized trajectories, both passed the same checks as the other sources on their first conversion. Adding a training signal is similar: our training set was preprocessed by a script that splits certain action explanations into reasoning and action components, for models that support reasoning. This processing script is around 500 lines long and is compatible with any dataset that populates a standard field on MMADP's Action objects. We ran it without modification on AgentNet, Go-Browse (Gandhi & Neubig, 2026), SWE-smith, and SWE-Gym, four datasets with different action spaces. It added 98,322 reasoning turns.

Table 1: Source datasets of the MMADP corpus by domain: coding (C), web (W), computer use (U), and general (G). *Axtree* and *Annot.* mark sources with accessibility trees and element annotations (boxes, OCR text); *Target* is how actions are grounded: element id (id), pixel (px) or normalized (0–1) coordinates, or element index (idx). *Code* is the length of each converter in lines. Token counts use the InternVL2.5 tokenizer; other models tokenize differently.

		Source-specific						
Dataset	Domain(s)	Axtree	Annot.	Target	Code	Trajectories	Turns/traj.	Tokens (B)
SWE-smith	C				188	17,378	27.2	0.50
SWE-Gym	C				203	491	21.4	0.01
OpenHands	C				800	118	19.0	0.01
Code-Feedback	C				107	30,213	4.5	0.26
CodeActInstruct	C				120	7,109	4.0	0.06
Go-Browse	W	•		id	300	6,594	5.0	0.27
WebLINX	W	•		id	459	969	25.6	0.13
Multimodal-Mind2Web	W	•	•	id	338	1,009	6.3	0.03
AgentNet	U			px	207	16,057	29.6	0.83
OmniACT	U		•	px	104	6,789	1.0	0.04
AndroidControl	U		•	px/idx	280	13,594	6.5	0.47
AITW	U		•	0–1	260	711,961	8.0	13.50
WildGUI	W, U			px	418	12.70M	9.7	340.06
ScaleCUA	W, U			px	532	37,382	5.3	0.65
LLaVA-Plus	G				193	117,039	2.3	1.00
Orca AgentInstruct	G				77	1.05M	1.1	8.53
Total					4,586	14.71M	9.0	366.35

3.5 Corpus Construction

We construct the MMADP corpus by selecting 16 datasets across the four domains (Appendix C). Of these, seven are shared with ADP, primarily in the coding domain: SWE-smith, SWE-Gym, OpenHands (All Hands AI, 2024), Code-Feedback (Zheng et al., 2024b), CodeActInstruct (Wang et al., 2024b), Go-Browse, and Orca AgentInstruct (Mitra et al., 2024). We keep them so that those domains stay represented, and we apply to them quality checks found in neither the original releases nor ADP. The other nine datasets are new, and all of them include images: WebLINX, Multimodal-Mind2Web (Deng et al., 2023; Zheng et al., 2024a), AgentNet, OmniACT, AndroidControl, AITW, LLaVA-Plus (Liu et al., 2024), WildGUI, and ScaleCUA. They cover the web, computer-use, and general domains. A full breakdown can be found in Table 1, and the platform, collection process, source, and license of each dataset in Appendix C.

3.5.1 Quality Checks

The converters also run quality checks, which Appendix B lists. At the trajectory level, we drop trajectories that perform no actions, except in instruct sets, where the response to the user is the answer. At the action level, we drop every action whose referenced element is absent from the observation the model sees, which removes 24.3% of Multimodal-Mind2Web actions. Every check runs at conversion time. A source's defects are found and fixed once, when the dataset joins the corpus, and not once per model trained on it. Some defects are visible only through a trained model. An early set of fixes, including indentation handling on code blocks, reduced the share of empty patches among the SWE-bench Verified attempts of an early trained model from 42.8% to 21.4%.

3.6 From Standard Trajectories to Training Records

ADP (Song et al., 2026) targeted OpenHands V0; MMADP targets OpenHands SDK V1 (Wang et al., 2026), in the ShareGPT format read by LlamaFactory (Zheng et al., 2024c). The SDK provides one default system prompt, whose tool list does not match the tools of every dataset. Our converter therefore contains one prompt per environment, with common dataset tool names mapped onto the

SDK's tool layer but otherwise passed as-is. For multimodal training, we add image observations and render their annotations as pseudo-accessibility trees next to the image tokens.

After corpus construction, we select the training data: at most 12,000 trajectories from each of 14 sources, 106,674 in total, which we release with MMADP (Appendix A). WildGUI and ScaleCUA were not fully converted at the time of the draw and were not sampled. Selection is a seeded draw up to a record cap, after which each record is truncated to the longest even-turn prefix under a token cap set by its predicted peak training memory (Appendix B). The corpus holds long multi-turn records: SWE-smith and AgentNet trajectories average 27.2 and 29.6 turns, respectively (Table 1), and 34% and 84% of their training records reach the token cap (Appendix A).

4 Experimental Setup

4.1 Training

We test whether the corpus teaches tool use across domains, one of the two capabilities credited to agentic midtraining (Tu et al., 2025; Cao et al., 2026). We select InternVL2.5-MPO at 8B and 26B as our base models, for their vision input and their lack of tool-calling or agentic post-training. The InternVL2.5 fine-tuning mix holds GUI data in question-answer form (Chen et al., 2024, Table 5) but no tool-calling or coding-agent data, and its data format has no tool or action role.

The corpus holds 14.7M trajectories in datasets of varying sizes. We subsample in different ratios to balance domains and to keep any one dataset from dominating: sampling the 14 sources of the training data in proportion to their size would give computer use 58% of the tokens, and AITW alone 53% (Table 1). The 250M-token mixture is therefore drawn from the training data (Section 3.6) with hand-set per-dataset shares (Appendix A) and a fixed seed. Datasets smaller than their share are upsampled by at most a factor of 2. Our mixed-domain run has its tokens split roughly equally across coding, web, and computer use; general datasets are included to limit forgetting (Zeng et al., 2024; Luo et al., 2025). To test whether domains help each other (Section 5.2), we also train three single-domain models, each filling the same 250M-token budget from its domain only. Each one scales up its domain's shares from the mixture, draws no dataset more than $2\times$ its size, and gives the remaining budget to the larger datasets. A fifth model trains on AgentNet alone, drawn from its full conversion at $1.13\times$ upsampling, as a test of existing corpora. All runs use LlamaFactory (Zheng et al., 2024c) for full fine-tuning of the language model and projector, with the vision tower frozen and images downscaled to 1 MP. We take the best checkpoint by eval loss on a held-out set with 6.25M tokens, drawn in the same way. Appendix D states how these runs differ from a midtraining run.

4.2 Evaluation

We evaluate on SWE-bench Verified (Jimenez et al., 2024; Chowdhury et al., 2024), MiniWoB++ (Liu et al., 2018), and AndroidWorld (Rawles et al., 2025), one benchmark each for coding, web, and computer use, matching the domains of the training data. SWE-bench Verified asks the agent to write a patch that resolves a real repository issue. MiniWoB++ asks it to follow an instruction on a small web interface, such as filling a form or selecting items in a list. AndroidWorld asks it to complete a task in an Android app, such as recording a piece of audio or adding a calendar event. A manual check of every source dataset against the three benchmarks found no contamination. We exclude WebArena (Zhou et al., 2024) because Go-Browse trajectories are collected on its sites. Every model runs in the same harness, the OpenHands SDK agent loop (Wang et al., 2026) adapted to each benchmark, with the XML tool-call format of the training records. Appendix A gives the tools and step limits of each benchmark. As reference points, two open models we did not train, Qwen3-VL-8B-Instruct (Bai et al., 2025) and GUI-Owl-1.5-8B (Xu et al., 2026), run under the same protocol (Appendix A).

Table 2: Percent of tasks solved before and after training on MMADP, with the gain over the base in parentheses for trained rows. Both sizes use the same data mixture and hyperparameters.

Model	SWE-bench Verified	MiniWoB++	AndroidWorld
InternVL2.5-8B-MPO	0.0%	14.1%	0.0%
+ MMADP	3.0% (+3.0%)	32.3% (+18.2%)	6.0% (+6.0%)
InternVL2.5-26B-MPO	0.0%	25.3%	1.7%
+ MMADP	5.4% (+5.4%)	37.4% (+12.1%)	6.0% (+4.3%)

Table 3: Tool calls the harness accepts before and after training on MMADP: valid calls of all calls attempted (SWE-bench Verified, final attempt per instance) and dispatched steps of all steps (AndroidWorld). 8B + Coding is the coding-only model of Table 7.

Model	SWE-bench Verified (valid / attempted)	AndroidWorld (dispatched / steps)
8B base	891 / 22,181 (4.0%)	112 / 2,318 (4.8%)
8B + MMADP	14,337 / 14,338 (99.99%)	769 / 770 (99.9%)
8B + Coding	8,891 / 8,891 (100%)	429 / 443 (96.8%)
26B base	10,937 / 10,937 (100%)	1,844 / 2,324 (79.4%)
26B + MMADP	15,329 / 15,329 (100%)	552 / 560 (98.6%)

5 Results and Analysis

5.1 Training on MMADP

Our primary contribution is the corpus. We read the model results as a measurement of what it teaches a model with no tool-calling training. Table 2 reports the base models at both sizes and the same models after training on the MMADP training data. At 8B, training takes SWE-bench Verified from 0.0% to 3.0%, MiniWoB++ from 14.1% to 32.3%, and AndroidWorld from 0.0% to 6.0%. Fewer than 20,000 trajectories (250M tokens, about 0.1% of the corpus) teach tool surfaces in three domains at once, even though some harness tools appear in none of the source datasets. Scores grow with size on SWE-bench Verified (3.0% to 5.4%) and MiniWoB++ (32.3% to 37.4%), but AndroidWorld stays at 6.0%. We attribute this to grounding: the base locates 1.1% of ScreenSpot-Pro targets (Li et al., 2025), against 55.9% for a newer model, Qwen3-VL-8B-Instruct. The corpus holds no single-step grounding data (Wu et al., 2025; Gou et al., 2025; Xu et al., 2025). We do not expect training on it to close that gap.

Table 3 reports tool-call validity, one of the two capabilities credited to agentic midtraining. Before training, the 8B base scores zero on SWE-bench Verified and AndroidWorld because it cannot act in the harness. On SWE-bench Verified, 96.0% of its tool calls fail validation, almost all by dropping required tags, despite the harness's fallback repairs and its syntax reminders after rejected calls. On AndroidWorld, it writes the parameter name inside the tag body (`<parameter>x=324</parameter>`) instead of in the tag (`<parameter=x>324</parameter>`). Training on our corpus teaches it the same tools: one SWE-bench Verified call in 14,338 fails validation at 8B and none at 26B. AndroidWorld dispatch rises from 4.8% to 99.9%. This is the starting point Section 2.1 describes for RL: a model that emits the tool-call format and runs the agentic loop in each domain. The single-domain models of Section 5.2 learn the same format and still score poorly outside their domain.

5.2 Mixing Domains

To test whether mixing domains helps, we train three single-domain models with the same base, recipe, and 250M-token budget as our 8B model, each filled from one domain of the training data

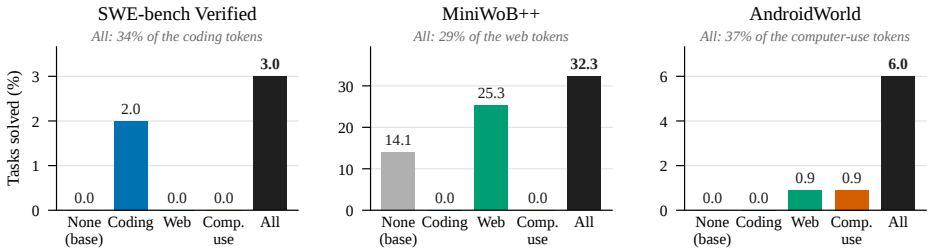


Figure 3: Percent of tasks solved at 8B when the same 250M-token budget is filled from one domain or from all domains. See Appendix A for exact numbers.

(Section 4.1, Appendix A).[1] All four models share the tool-call format and harness and differ only in the composition of their training data. Figure 3 reports them against the mixed-domain model.

The mixed-domain model beats each single-domain model on that model's own benchmark while seeing about a third of that domain's tokens. With a third of the coding data, it resolves 3.0% of SWE-bench Verified instances against 2.0%. With 29% of the web data, it solves 32.3% of MiniWoB++ tasks against 25.3%. With 37% of the computer-use data, it completes 6.0% of AndroidWorld tasks against 0.9%. The single-domain models' tool calls show how they fail outside their domain. The coding-only model ends 83 of the 99 scored MiniWoB++ episodes at step 1 by calling `finish` and never calls a browser tool. The web-only model submits empty SWE-bench Verified patches. The computer-use-only model also calls `finish` immediately on SWE-bench Verified. Those `finish` calls are valid, and on SWE-bench Verified the coding-only model's calls pass validation at the same rate as the mixed-domain model's (Table 3).

A natural question is whether an existing dataset does the same on its own. The AgentNet-only model (Section 4.1) tests this with a source that is large, has long trajectories (Table 1), spans three operating systems and many tasks, and uses a tool set close to that of an evaluation harness. It solves no task on any of the three benchmarks (Appendix A). On AndroidWorld it calls only the PyAutoGUI tool, always with `x` and `y` parameters where the tool takes `code`. Fewer than 1% of its calls dispatch. On MiniWoB++ it calls only `think`. One source, even a large one, appears to teach one surface, which the model then applies everywhere. Mixing domains in this way is practical because MMADP writes every domain in the tool-call format of one harness.

6 Conclusion

We introduce MMADP, an extension of ADP to multimodal data, and use it to build the largest open corpus of agentic trajectories, the only one that joins coding-agent trajectories with web and computer-use trajectories in one schema. Fine-tuning on a 250M-token sample of it teaches multi-domain tool use to a vision-language model with no tool-calling training, at both 8B and 26B. Further, at a fixed token budget, a mix of domains does better on each domain's own benchmark than single-domain models do.

We see three directions for future work: a midtraining run on the corpus from a pretrained checkpoint with the pretraining objective, followed by a shared post-training stage; learned mixture weights instead of the hand-set weights we use; and the addition of other datasets and harnesses. Other harnesses matter because our SFT target writes tool calls as XML text, and one more Standardized-to-SFT script would produce native tool calls for other harnesses from the same corpus.

Reproducibility Statement

Section 3.3 defines the MMADP schema, and Section 3.4 describes the conversion pipeline, and Section 3.5 its quality checks. Table 1 lists every source dataset, and Appendix C gives the release and license of each one. We release the schema and the full pipeline (converters, quality checks, rendering, and selection) as code. The corpus can be reconstructed by running the provided scripts.

[1] We assigned the two general datasets to the single-domain model that their data most resembles.

We also release the training data of Section 3.6 on Hugging Face and the training configurations of every reported model. The training data includes the reasoning turns generated with Gemini 2.5 Flash-Lite (Comanici et al., 2025). Section 4.1 and Appendix A give further details on data mixtures. Appendix A also provides details on replicating the benchmark numbers.

AI USE STATEMENT

We used generative AI tools (large language models, including coding agents and chat assistants) in this work. Among the tasks with required disclosure, we used them to implement methods: the conversion pipeline, its quality checks, and the training and evaluation scripts were partially written by an LLM coding agent under author direction. The reasoning turns in the released training data were generated with Gemini 2.5 Flash-Lite (Section 3.4 and Appendix B); no other released data was generated by a model. To a smaller extent, we used generative AI for feedback on research methodology and experiments, to interpret results, and to support qualitative analysis of results. We did not use generative AI to develop theoretical models or conceptual frameworks, to propose hypotheses, or to clean or reformat dataset records directly. Formulating or proving mathematical claims, and translation, are not applicable to this work. Additionally, we used generative AI throughout the project as a copilot and second opinion, which includes editing code and paper text, rendering figures, and summarizing related literature, among other tasks. We have reviewed all AI-assisted work, including all code, which an author reviewed change by change and which the unit tests and the quality checks of Section 3.5 cover. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

All Hands AI. OpenHands feedback dataset. `https://huggingface.co/datasets/OpenHands/openhands-feedback`, 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 preprint arXiv:2511.21631*, 2025.

Ruisheng Cao, Mouxiang Chen, Jiawei Chen, Zeyu Cui, Yunlong Feng, Binyuan Hui, Yuheng Jing, Kaixin Li, Mingze Li, Junyang Lin, Zeyao Ma, Kashun Shum, Xuwu Wang, Jinxi Wei, Jiaxi Yang, Jiajun Zhang, Lei Zhang, Zongmeng Zhang, Wenting Zhao, and Fan Zhou. Qwen3-Coder-Next technical report. *arXiv preprint arXiv:2603.00729*, 2026.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

Neil Chowdhury, James Aung, Chan Jun Shern, Oliver Jaffe, Dane Sherburn, Giulio Starace, Evan Mays, Rachel Dias, Marwan Aljubeh, Mia Glaese, Carlos E. Jimenez, John Yang, Leyton Ho, Tejal Patwardhan, Kevin Liu, and Aleksander Madry. Introducing SWE-bench verified. `https://openai.com/index/introducing-swe-bench-verified/`, 2024.

Gheorghe Comanici et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and PixMo: Open weights and open data for state-of-the-art vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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 *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Apurva Gandhi and Graham Neubig. Go-Browse: Training web agents with structured exploration. In *International Conference on Learning Representations (ICLR)*, 2026.

GLM-4.5 Team, Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, et al. GLM-4.5: Agentic, reasoning, and coding (ARC) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

GLM-5 Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, et al. GLM-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

GLM-V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, et al. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 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 *International Conference on Learning Representations (ICLR)*, 2025.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. DeepEyesV2: Toward agentic multimodal model. In *International Conference on Learning Representations (ICLR)*, 2026.

Fengqing Jiang, Yite Wang, Boyi Liu, Zhaoyang Wang, Canwen Xu, Zhewei Yao, Radha Poovendran, and Yuxiong He. MidTool: Mid-training data synthesis for agentic tool use. *arXiv preprint arXiv:2608.20314*, 2026.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024.

Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem AlShikh, and Ruslan Salakhutdinov. OmniACT: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In *European Conference on Computer Vision (ECCV)*, pp. 161–178, 2024. doi: 10.1007/978-3-031-73113-6_10.

Thibault Le Sellier de Chezelles, Maxime Gasse, Alexandre Drouin, Massimo Caccia, Léo Boisvert, Megh Thakkar, Tom Marty, Rim Assouel, Sahar Omidi Shayegan, Lawrence Keunho Jang, Xing Han Lù, Ori Yoran, Dehan Kong, Frank F. Xu, Siva Reddy, Quentin Cappart, Graham Neubig, Ruslan Salakhutdinov, Nicolas Chapados, and Alexandre Lacoste. The BrowserGym ecosystem for web agent research. *Transactions on Machine Learning Research*, 2025.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. ScreenSpot-Pro: GUI grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 8778–8786, 2025. doi: 10.1145/3746027.3755688.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on UI control agents. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement learning on web interfaces using workflow-guided exploration. In *International Conference on Learning Representations (ICLR)*, 2018.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, Lei Zhang, Jianfeng Gao, and Chunyuan Li. LLaVA-Plus: Learning to use tools for creating multimodal agents. In *European Conference on Computer Vision (ECCV)*, 2024. doi: 10.1007/978-3-031-72970-6_8.

Zhaoyang Liu, Jingjing Xie, Zichen Ding, Zehao Li, Bowen Yang, Zhenyu Wu, Xuehui Wang, Qiushi Sun, Shi Liu, Weiyun Wang, Shenglong Ye, Qingyun Li, Zeyue Tian, Gen Luo, Xiangyu Yue, Biqing Qi, Kai Chen, Bowen Zhou, Yu Qiao, Qifeng Chen, and Wenhai Wang. ScaleCUA: Scaling open-source computer use agents with cross-platform data. In *International Conference on Learning Representations (ICLR)*, 2026.

Fanbin Lu, Zhisheng Zhong, Shu Liu, Chi-Wing Fu, and Jiaya Jia. ARPO: End-to-end policy optimization for GUI agents with experience replay. *arXiv preprint arXiv:2505.16282*, 2025a.

Quanfeng Lu, Wenqi Shao, Zitao Liu, Lingxiao Du, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, and Ping Luo. GUIOdyssey: A comprehensive dataset for cross-app GUI navigation on mobile devices. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

Xing Han Lù, Zdeněk Kasner, and Siva Reddy. WebLINX: Real-world website navigation with multi-turn dialogue. In *International Conference on Machine Learning (ICML)*, 2024.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025. doi: 10.1109/taslpro.2025.3606231.

Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Codas, Yadong Lu, Wei-ge Chen, Olga Vrousgos, Corby Rosset, Fillipe Silva, Hamed Khanpour, Yash Lara, and Ahmed Awadallah. AgentInstruct: Toward generative teaching with agentic flows. *arXiv preprint arXiv:2407.03502*, 2024.

Kaixiang Mo, Yuxin Shi, Weiwei Weng, Zhiqiang Zhou, Shuman Liu, Haibo Zhang, and Anxiang Zeng. Mid-training of large language models: A survey. *arXiv preprint arXiv:2510.06826*, 2025.

Jian Mu, Chaoyun Zhang, Chiming Ni, Lu Wang, Bo Qiao, Kartik Mathur, Qianhui Wu, Yuhang Xie, Xiaojun Ma, Mengyu Zhou, Si Qin, Liqun Li, Yu Kang, Minghua Ma, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. GUI-360°: A comprehensive dataset and benchmark for computer-using agents. *arXiv preprint arXiv:2511.04307*, 2025.

Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. Training software engineering agents and verifiers with SWE-Gym. In *International Conference on Machine Learning (ICML)*, 2025.

Baolin Peng, Wenlin Yao, Qianhui Wu, Hao Cheng, Xiao Yu, Rui Yang, Tao Ge, Alessandro Sordoni, Xingdi Yuan, Yelong Shen, Pengcheng He, Tong Zhang, Zhou Yu, and Jianfeng Gao. Orchard: An open-source agentic modeling framework. *arXiv preprint arXiv:2605.15040*, 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 preprint arXiv:2501.12326*, 2025.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Android in the wild: A large-scale dataset for android device control. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2025.

Yifan Song, Weimin Xiong, Xiutian Zhao, Dawei Zhu, Wenhao Wu, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. AgentBank: Towards generalized LLM agents via fine-tuning on 50000+ interaction trajectories. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2124–2141, 2024. doi: 10.18653/v1/2024.findings-emnlp.116.

Yueqi Song, Ketan Ramaneti, Zaid Sheikh, Ziru Chen, Boyu Gou, Tianbao Xie, Yiheng Xu, Danyang Zhang, Apurva Gandhi, Fan Yang, Joseph Liu, Tianyue Ou, Zhihao Yuan, Frank Xu, Shuyan Zhou, Xingyao Wang, Xiang Yue, Tao Yu, Huan Sun, Yu Su, and Graham Neubig. Agent data protocol: Unifying datasets for diverse, effective fine-tuning of LLM agents. In *International Conference on Learning Representations (ICLR)*, 2026.

Liangcai Su, Zhen Zhang, Guangyu Li, Zhuo Chen, Chenxi Wang, Maojia Song, Xinyu Wang, Kuan Li, Jialong Wu, Xuanzhong Chen, Zile Qiao, Zhongwang Zhang, Huifeng Yin, Shihao Cai, Runnan Fang, Zhengwei Tao, Wenbiao Yin, Rui Ye, Yong Jiang, Ningyu Zhang, Pengjun Xie, Fei Huang, Kai Ye, Kewei Tu, Chenxiong Qian, and Jingren Zhou. Scaling agents via continual pre-training. In *International Conference on Learning Representations (ICLR)*, 2026.

Chengying Tu, Xuemiao Zhang, Rongxiang Weng, Rumei Li, Chen Zhang, Yang Bai, Hongfei Yan, Jingang Wang, and Xunliang Cai. A survey on LLM mid-training. *arXiv preprint arXiv:2510.23081*, 2025.

Venus Team, Zhuohan Cai, Haoxing Chen, Jiaxuan Chen, Weizhi Chen, Changlong Gao, Zhangxuan Gu, Yuan Guo, Yusong Hu, Jianrong Jiang, et al. UI-Venus-2 technical report. *arXiv preprint arXiv:2609.00028*, 2026.

Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Junting Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, Wanjun Zhong, Yining Ye, Yujia Qin, Yuwen Xiong, Yuxin Song, Zhiyong Wu, Aoyan Li, Bo Li, Chen Dun, Chong Liu, Daoguang Zan, Fuxing Leng, Hanbin Wang, Hao Yu, Haobin Chen, Hongyi Guo, Jing Su, Jingjia Huang, Kai Shen, Kaiyu Shi, Lin Yan, Peiyao Zhao, Pengfei Liu, Qinghao Ye, Renjie Zheng, Shulin Xin, Wayne Xin Zhao, Wen Heng, Wenhao Huang, Wenqian Wang, Xiaobo Qin, Yi Lin, Youbin Wu, Zehui Chen, Zihao Wang, Baoquan Zhong, Xinchun Zhang, Xujing Li, Yuanfan Li, Zhongkai Zhao, Chengquan Jiang, Faming Wu, Haotian Zhou, Jinlin Pang, Li Han, Qi Liu, Qianli Ma, Siyao Liu, Songhua Cai, Wenqi Fu, Xin Liu, Yaohui Wang, Zhi Zhang, Bo Zhou, Guoliang Li, Jiajun Shi, Jiale Yang, Jie Tang, Li Li, Qihua Han, Taoran Lu, Woyu Lin, Xiaokang Tong, Xinyao Li, Yichi Zhang, Yu Miao, Zhengxuan Jiang, Zili Li, Ziyuan Zhao, Chenxin Li, Dehua Ma, Feng Lin, Ge Zhang, Haihua Yang, Hangyu Guo, Hongda Zhu, Jiaheng Liu, Junda Du, Kai Cai, Kuanye Li, Lichen Yuan, Meilan Han, Minchao Wang, Shuyue Guo, Tianhao Cheng, Xiaobo Ma, Xiaojun Xiao, Xiaolong Huang, Xinjie Chen, Yidi Du, Yilin Chen, Yiwen Wang, Zhaojian Li, Zhenzhu Yang, Zhiyuan Zeng, Chaolin Jin, Chen Li, Hao Chen, Haoli Chen, Jian Chen, Qinghao Zhao, and Guang Shi. UI-TARS-2 technical report: Advancing GUI agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544*, 2025a.

Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025b.

Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024a.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents. In *International Conference on Machine Learning (ICML)*, 2024b.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. OpenHands: An open platform for AI software developers as generalist agents. In *International Conference on Learning Representations (ICLR)*, 2025c.

Xingyao Wang, Simon Rosenberg, Juan Michelini, Calvin Smith, Hoang Tran, Engel Nyst, Rohit Malhotra, Xuhui Zhou, Valerie Chen, Robert Brennan, and Graham Neubig. The OpenHands software agent SDK: A composable and extensible foundation for production agents. In *Conference on Machine Learning and Systems (MLSys)*, 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025d.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. WebAgent-R1: Training web agents via end-to-end multi-turn reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

Zheng Wu, Chengcheng Han, Zhengxi Lu, Tianjie Ju, Yanyu Chen, Qi Gu, Xunliang Cai, and Zhuosheng Zhang. GUI-CIDER: Mid-training GUI agents via causal internalization and density-aware exemplar reselection. *arXiv preprint arXiv:2605.28534*, 2026.

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 *International Conference on Learning Representations (ICLR)*, 2025.

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.

Haiyang Xu, Xi Zhang, Haowei Liu, Junyang Wang, Zhaozai Zhu, Shengjie Zhou, Xuhao Hu, Feiyu Gao, Junjie Cao, Zihua Wang, Zhiyuan Chen, Jitong Liao, Qi Zheng, Jiahui Zeng, Ze Xu, Shuai Bai, Junyang Lin, Jingren Zhou, and Ming Yan. Mobile-Agent-v3.5: Multi-platform fundamental GUI agents. *arXiv preprint arXiv:2602.16855*, 2026.

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 *International Conference on Machine Learning (ICML)*, pp. 69772–69805, 2025.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V. *arXiv preprint arXiv:2310.11441*, 2023.

Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, Yuquan Deng, and Jianfeng Gao. Magma: A foundation model for multimodal AI agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14203–14214, 2025a.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 50528–50652, 2024. doi: 10.52202/079017-1601.

John Yang, Kilian Lieret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. SWE-smith: Scaling data for software engineering agents. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025b. doi: 10.52202/085713-3239.

Rui Yang, Qianhui Wu, Zhaoyang Wang, Hanyang Chen, Ke Yang, Hao Cheng, Huaxiu Yao, Baolin Peng, Huan Zhang, Jianfeng Gao, and Tong Zhang. GUI-Libra: Training native GUI agents to reason and act with action-aware supervision and partially verifiable RL. *arXiv preprint arXiv:2602.22190*, 2026.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. AgentTuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, 2024. doi: 10.18653/v1/2024.findings-acl.181.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Chaoyun Zhang, Shilin He, Jiaxu Qian, Bowen Li, Liqun Li, Si Qin, Yu Kang, Minghua Ma, Guyue Liu, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Large language model-brained GUI agents: A survey. *Transactions on Machine Learning Research*, 2025a.

Jianguo Zhang, Tian Lan, Rithesh Murthy, Zhiwei Liu, Weiran Yao, Ming Zhu, Juntao Tan, Thai Hoang, Zuxin Liu, Liangwei Yang, Yihao Feng, Shirley Kokane, Tulika Awalgaonkar, Juan Carlos Niebles, Silvio Savarese, Shelby Heinecke, Huan Wang, and Caiming Xiong. AgentOhana: Design unified data and training pipeline for effective agent learning. *arXiv preprint arXiv:2402.15506*, 2024a.

Jiwen Zhang, Jihao Wu, Yihua Teng, Minghui Liao, Nuo Xu, Xiao Xiao, Zhongyu Wei, and Duyu Tang. Android in the zoo: Chain-of-action-thought for GUI agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024b.

Junlei Zhang, Zichen Ding, Chang Ma, Zijie Chen, Qiushi Sun, Zhenzhong Lan, and Junxian He. Breaking the data barrier – building GUI agents through task generalization. In *Conference on Language Modeling (COLM)*, 2025b.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. GPT-4V(ision) is a generalist web agent, if grounded. In *International Conference on Machine Learning (ICML)*, 2024a.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. OpenCodeInterpreter: Integrating code generation with execution and refinement. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12834–12859, 2024b. doi: 10.18653/v1/2024.findings-acl.762.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 400–410, 2024c. doi: 10.18653/v1/2024.acl-demos.38.

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 *International Conference on Learning Representations (ICLR)*, 2024.

A Corpus and Training Details

Table 4 compares the MMADP corpus with other open agent-trajectory corpora by size, domain, and schema. Table 5 gives the statistics of the training data and the mixture drawn from it for the mixed-domain and single-domain models, and Table 6 the evaluation protocol of every benchmark. Table 7 lists the numbers plotted in Figure 3 and the results of the AgentNet-only model, and Table 8 reports two open models that we did not train, run under the same protocol.

Table 4: Comparable agent-trajectory corpora not currently in MMADP. Most of these corpora could be added with one converter each. Domains are coding (C), web (W), computer use (U), and general (G). Images: the release holds screenshots or images. Schema: the source has a single consistent action-observation format (● = yes, ○ = partly). Traj. is the total trajectory count. [s]Counted as single-step samples (grounding or single-turn). [c]Coding trajectories only.

Corpus	Domain(s)	Images	Schema	Size (GB)	Traj.	Turns/traj.	License
OS-Atlas-data (Wu et al., 2025)	W, U	●	●	816	13.6M[s]	1	Apache 2.0
GUI-360 (Mu et al., 2025)	U	●	●	574	79.4K	15.4	MIT
GUI-Net-1M (Zhang et al., 2026)	W, U	●	●	570	895K[s]	1–9	Apache 2.0
UGround-V1-Data (Gou et al., 2025)	W	●	●	425	9.4M[s]	1	CC BY-NC-SA 4.0
Aguvis stage 1 (Xu et al., 2025)	W, U	●	●	180	1.04M[s]	1	Apache 2.0
Aguvis stage 2 (Xu et al., 2025)	W, U	●	●	168	35K	3.6–15.3	Apache 2.0
GUI-Odyssey (Lu et al., 2025b)	U	●	●	93	8.3K	15.3	CC BY 4.0
GUIMid (Zhang et al., 2025b)	W, U	●	●	44	356K[s]	1	MIT
Orchard (Peng et al., 2026)	C, W	●	○	11	107.6K	47.5[c]	MIT
MMADP	C, W, U, G	●	●	13,407	14.7M	9.0	per source

Table 5: Statistics of the training data and the 250M-token mixture drawn from it for the mixed-domain model and the three single-domain models. Turns is the mean number of assistant turns per record. Tokens per screenshot and the image share use the tokenizer and image tiling of the trained model. Trunc. is the share of records that the selection stage truncated to fit the token cap. Dom. is the domain as in Table 1.

		Training data					Mixture (M tokens)			
Dataset	Dom.	Records	Turns	Tokens /shot	Image (%)	Trunc. (%)	All	Coding	Web	Comp. use
SWE-smith	C	12,000	21.3	–	0	34	40	129	–	–
SWE-Gym	C	491	18.7	–	0	17	11	22	–	–
OpenHands	C	118	13.5	2,304	3	33	5	5	–	–
Code-Feedback	C	12,000	4.5	–	0	0	12	39	–	–
CodeActInstruct	C	7,109	4.0	–	0	0	12	39	–	–
Go-Browse	W	6,594	4.1	2,559	27	34	40	–	182	–
WebLINX	W	564	11.7	2,311	45	77	20	–	41	–
Multimodal-Mind2Web	W	1,009	5.8	2,304	68	13	13	–	27	–
AgentNet	U	12,000	17.7	2,097	75	84	35	–	–	95
OmniACT	U	6,789	1.0	1,964	32	0	10	–	–	27
AndroidControl	U	12,000	5.9	2,794	52	11	27	–	–	73
AITW	U	12,000	6.5	867	35	31	10	–	–	27
LLaVA-Plus	G	12,000	2.3	2,097	24	0	10	–	–	27
Orca AgentInstruct	G	12,000	1.1	–	0	0	5	16	–	–
Total		106,674	7.5	1,992	36	21	250	250	250	250

Table 6: Evaluation protocol. Every model in Tables 2 and 8 runs through an XML tool-calling harness built on the OpenHands SDK, served at 32k context with seed 42; all models sample at their base's default settings (InternVL2.5 and Qwen3-VL defaults). Stuck, empty, and errored episodes count as failures. MiniWoB++ excludes 26 tasks that need drag, text selection, a slider, a select option, key events, or a free coordinate, and AndroidWorld excludes two tasks with broken checkers.

Benchmark	Tasks	Tools advertised	Step limit
SWE-bench Verified	500	terminal, file editor, think, finish	50 iterations; 4,096 output tokens; observations cut at 10k chars
MiniWoB++	99	browser navigate, click, type, scroll, go back; think, finish	30 steps; 360 s per step
AndroidWorld	116	AndroidControl surface (click by coordinates, long press, type, scroll, home, back, open app, wait); think, finish	suite default per task

Table 7: Percent of tasks solved at 8B when the same 250M-token budget is filled from one source, one domain, or all domains (Figure 3). The best entry per column is in **bold**, and the second best is underlined.

Training data	SWE-bench Verified	MiniWoB++	AndroidWorld
None (base)	0.0%	14.1%	0.0%
AgentNet only	0.0%	0.0%	0.0%
Coding only	2.0%	0.0%	0.0%
Web only	0.0%	25.3%	0.9%
Computer use only	0.0%	0.0%	0.9%
All	**3.0%**	**32.3%**	**6.0%**

Table 8: Percent of tasks solved by two open models we did not train, run under the protocol of Table 6. Qwen3-VL-8B-Instruct had agentic post-training before release, and GUI-Owl-1.5-8B is the same model after further GUI pretraining, SFT, and RL on unreleased data. Both were trained for their own agent frameworks. Our harness measures them outside the setting of their published scores. They share neither our base nor our training data. The rows give context for Table 2 and do not test the corpus.

Model	SWE-bench Verified	MiniWoB++	AndroidWorld
Qwen3-VL-8B-Instruct (Bai et al., 2025)	14.6%	46.5%	18.1%
GUI-Owl-1.5-8B (Xu et al., 2026)	15.0%	35.4%	11.2%

B Converter Details

Raw to standardized. One converter per source reads the source release and emits trajectories in the MMADP schema, which the schema's Pydantic models validate. It populates image observations and, where the source provides them, their element annotations. Each converter is specific to its dataset but follows three rules. First, it propagates environment feedback, such as browser error messages from a previous action. Second, it populates missing fields where reasonable: when a web source lacks accessibility trees, the converter renders its pages in headless Chromium to produce the tree, a set-of-marks screenshot, and resolved element IDs. Third, it preserves the tool call names and arguments that the source defines in its `api.py`.

Standardized to training records. This stage renders each trajectory as a trainable object. In our experiments, this took the form of ShareGPT records: a system prompt, user turns that hold the observations, and assistant turns that hold the agent's tool calls in the OpenHands SDK format. The converter passes text observations through as messages and renders each image observation as an `<image>` token followed by its annotations as a pseudo-accessibility tree. The markup, the `think` and `finish` tools, and the observation wrapper are the same in every environment; only the tool set differs. We provide an abridged example of a `think` call, its observation, and the following action:

```
<function=think>
<parameter=thought>The About link in the navigation
should list the hosting partners.</parameter>
</function>

EXECUTION RESULT of [think]:
Your thought has been logged.

<function=browser_click>
<parameter=bid>117</parameter>
</function>
```

The converter selects one of three system prompts by the task type of the source: coding, web, or desktop. The coding prompt is the SDK default, with its in-context example; the web and desktop prompts remove coding guidance (version control, pull requests, environment setup) and add one line that describes the environment. Every prompt advertises the SDK's terminal, file editor, think, and finish tools; the web prompt adds the SDK browser tools, and the desktop prompt adds a PyAutoGUI tool. A dataset's helpers are advertised from its `api.py` signatures and passed through as-is. A source needs no mapping to convert. Where we notice that a common helper duplicates an SDK tool, we may map it onto that tool for consistency, as with Go-Browse's `fill` to `browser_type` and SWE-smith's `submit` to `finish`.

Before rendering, one script transforms the data at the schema level. For sources whose actions carry an explanation but no separate reasoning turn, we add a step in which an LLM splits each explanation into a thought and a short narrative, producing a synthetic `think` call before the action. The first explained action of a trajectory always receives a `think` call, and later turns are converted at random. This script is written once for the standard schema and processes four datasets' action explanations together. This was done to improve the model's use of the `think` tool, and the change had a measurable impact on the frequency of `think` calls.

Selection. The selection stage (Section 3.6) draws records with a fixed seed up to a per-source record cap, then truncates each record to the longest prefix that fits under a token cap and has an even number of turns, as the LlamaFactory ShareGPT format requires. The cap is the smaller of a fixed token limit and a limit predicted from the record's peak GPU memory in our training setup; the released pipeline takes both as settings. We fit peak memory to the image and text token counts of measured training steps and keep each record under that prediction.

Quality checks. Checks run at both conversion boundaries. After the first stage, the schema validates every trajectory. At rendering, every tool call is checked against the tool's signature. We also verify that every element ID a call references appears in the observation the model sees. Set-of-marks filtering drops most entries of long lists; we change it to keep the element that each gold action references, and a final check drops every action whose element is still absent (Section 3.5). We drop trajectories with placeholder element IDs and, in WebLINX, the actions whose target element lacks the `data-webtasks-id` attribute that identifies elements in its BrowserGym release (Section 3.2). Where a coordinate click maps unambiguously to one element, we randomly rewrite it as an element-index click so that both grounding forms appear in the corpus.

C Dataset Provenance and Licenses

Table 9 gives the platform, collection process, converted size, source release, and license of every dataset in the MMADP corpus of Table 1.

Table 9: Provenance and citation of each dataset in MMADP. Collection states who or what produced the trajectories and the success filter we apply. Size is disk space. Sources are Hugging Face dataset repositories unless marked GCS (Google Cloud Storage).

Dataset	Platform	Collection	Size (GB)	Source	License
SWE-smith (Yang et al., 2025b)	repository sandbox	Claude 3.7, Claude 3.5, GPT-4o agents; resolved	2.1	SWE-bench/SWE-smith-trajectories	MIT
SWE-Gym (Pan et al., 2025)	repository sandbox	GPT-4o, Claude 3.5 agents; resolved	<0.1	SWE-Gym/OpenHands-Sampled-Trajectories	Apache-2.0
OpenHands (All Hands AI, 2024)	OpenHands sandbox	user sessions; rated positive	<0.1	OpenHands/openhands-feedback	MIT
Code-Feedback (Zheng et al., 2024b)	Python interpreter	GPT-3.5/4 simulated	1.2	m-a-p/Code-Feedback	Apache-2.0
CodeActInstruct (Wang et al., 2024b)	Python interpreter	GPT, Claude agents; successful	0.3	xingyaoww/code-act	Apache-2.0
Go-Browse (Gandhi & Neubig, 2026)	WebArena sites	Claude 3.7, GPT-4o mini, Qwen2.5 agents; reward 1	10.6	apurvaga/go-browse-wa-raw	MIT
WebLINX (Lù et al., 2024)	155 real websites	expert annotators	5.3	McGill-NLP/weblinx-browsergym	CC BY-NC-SA 4.0
Multimodal-Mind2Web (Deng et al., 2023; Zheng et al., 2024a)	137 real websites	crowdworkers	0.5	osunlp/Multimodal-Mind2Web	OpenRAIL
AgentNet (Wang et al., 2025d)	Windows, macOS, Ubuntu	annotators; completed	138	xlangai/AgentNet	MIT
OmniACT (Kapoor et al., 2024)	desktop apps, websites	student annotators	0.3	Writer/omniact	MIT
AndroidControl (Li et al., 2024)	Android phones	paid annotators	44.6	google-research android_control (GCS)	Apache-2.0
AITW (Rawles et al., 2023)	Android emulator	human raters	527	google-research android_in_the_wild (GCS)	Apache-2.0
WildGUI (Xiong et al., 2026)	Windows, macOS, Linux, Android, iOS	tutorial videos; Gemini-annotated	12,555	xwm/WildGUI + joker-112/WildGUI_Screenshots	CC BY-NC 4.0
ScaleCUA (Liu et al., 2026)	Windows, macOS, Ubuntu, Android, web	annotators and agents; VLM-annotated	66.2	OpenGVLab/ScaleCUA-Data	Apache-2.0
LLaVA-Plus (Liu et al., 2024)	static images	GPT-4 synthetic	16.2	LLaVA-VL/llava-plus-data; images from six external sources	CC BY-NC 4.0
Orca AgentInstruct (Mitra et al., 2024)	none (text)	GPT-4 synthetic	39.8	microsoft/orca-agentinstruct-1M-v1	CDLA-Permissive-2.0
Total			13,407		

We have ensured compliance with the licenses of all datasets used in this paper. All licenses permit research use.

Licenses may have changed since we collected the datasets. Corpus users are responsible for verifying current terms; we do not assume responsibility for license violations.

D Limitations

Our training runs are supervised fine-tuning and differ from a midtraining run in three respects. They start from InternVL2.5-MPO, a post-trained model, rather than a pretrained checkpoint. They mask the loss to the assistant turns of chat-formatted records, whereas midtraining keeps the next-token objective of pretraining (Tu et al., 2025), computed over every token (Su et al., 2026). They train on 250M tokens drawn from the training data; the corpus holds 366B. A midtraining run on the corpus is untested.