The Fortieth AAAI Conference on Artificial Intelligence (AAAI-26)

Co-EPG: A Framework for Co-Evolution of Planning and Grounding in Autonomous GUI Agents

Yuan Zhao[1*], **Hualei Zhu**[1*], **Tingyu Jiang**[1†], **Shen Li**[1], **Xiaohang Xu**[2†], **Hao Henry Wang**[1]
[1]Alibaba Cloud Computing
[2]Graduate School of Information Science and Technology, The University of Tokyo
{zhaoyuan.yz, zhuhualei.zhl, jiangtingyu.jty, lishen.ls, qiao.wh}@alibaba-inc.com
xhxu@eidos.ic.i.u-tokyo.ac.jp

Abstract

Graphical User Interface (GUI) task automation constitutes a critical frontier in artificial intelligence research. While effective GUI agents synergistically integrate planning and grounding capabilities, current methodologies exhibit two fundamental limitations: (1) insufficient exploitation of cross-model synergies, and (2) over-reliance on synthetic data generation without sufficient utilization. To address these challenges, we propose **Co-EPG**, a self-iterative training framework for **Co**-**E**volution of **P**lanning and **G**rounding. Co-EPG establishes an iterative positive feedback loop: through this loop, the planning model explores superior strategies under grounding-based reward guidance via Group Relative Policy Optimization (GRPO), generating diverse data to optimize the grounding model. Concurrently, the optimized Grounding model provides more effective rewards for subsequent GRPO training of the planning model, fostering continuous improvement. Co-EPG thus enables iterative enhancement of agent capabilities through self-play optimization and training data distillation. On the Multimodal-Mind2Web and AndroidControl benchmarks, our framework outperforms existing state-of-the-art methods after just three iterations without requiring external data. The agent consistently improves with each iteration, demonstrating robust self-enhancement capabilities. This work establishes a novel training paradigm for GUI agents, shifting from isolated optimization to an integrated, self-driven co-evolution approach.

1 Introduction

In recent years, with the rapid development of large-scale vision language models (LVLMs), building autonomous agents capable of understanding and interacting with graphical user interfaces (GUIs) has emerged as a highly promising application area, attracting extensive research attention (OpenAI 2025; Li et al. 2025).

The academic community has not yet established a unified paradigm for designing GUI agents. Fundamentally, a GUI agent requires two core capabilities: planning and grounding. Planning determines the action and its values based on the current screen state, while grounding identifies the target element's location. Common approaches involve training a monolithic model (He et al. 2024; Cheng et al. 2024), such as directly training on large-scale trajectory data (Xu et al. 2024a; Pahuja et al. 2025), pretraining on grounding tasks followed by fine-tuning on planning tasks (Xu et al. 2024b; Wu et al. 2024), or introducing online Reinforcement Learning (RL) and environment interaction to explore more generalized strategies (Wei et al. 2025). Some studies even employ mechanisms like world model construction (Fang et al. 2025) or self-evolving curriculum learning (Qi et al. 2024) to enable continuous self-improvement of model capabilities. However, these monolithic models increasingly expose limitations in perception and interaction when generalized to diverse GUI environments (Gou et al. 2024). Consequently, the research focus has shifted towards more flexible modular designs, primarily characterized by task decoupling and multi-model collaboration. For instance, some studies employ a collaborative framework between high-level planning and low-level grounding to enhance the accuracy and flexibility of agents (Zhang et al. 2025a), while others construct cooperative multi-agent systems to better handle complex tasks (Zhao et al. 2025; Zhang et al. 2024; Wang et al. 2024b; Liu et al. 2025b). However, current collaborative architectures for GUI agents face two critical challenges: (1) They predominantly rely on independent model optimization, which neglects the potential for synergistic co-evolution between interdependent components like planning and grounding. (2) This reliance fosters a dependency on vast synthetic datasets, underutilizing existing data and introducing synthetic noise. Therefore, it is imperative to develop a novel collaborative paradigm that enables the synergistic evolution of planning and grounding models while maximizing the utility of available data.

In this study, we propose Co-EPG, a self-iterative training framework for Co-Evolution of Planning and Grounding. The core of Co-EPG is a positive feedback loop that drives the co-evolution of both models. The planning model explores new strategies using GRPO to generate diverse and specific plans, which progressively enhance the grounding model's execution capabilities. In turn, the improved grounding model delivers higher-quality rewards, guiding the planning model toward more effective strategies. This iterative loop, where the plan bridges the two models and provides specific information to the grounding model, enables continuous self-improvement. During the collabora-

---

[*] The first two authors contributed equally.
[†] Corresponding Author.

tive training of both models, we propose a confidence-based dynamic reward ensemble mechanism (C-DREM), which effectively reduces reward noise by aggregating rewards from multiple grounding models with confidence-based weighting. Our contributions can be summarized as follows:

1. We propose Co-EPG, a self-iterative training framework for Co-Evolution of Planning and Grounding. The framework establishes a positive feedback loop in which the grounding model guides the planning model's strategy exploration through reward, while the optimized planning model generates high-quality data to further enhance the grounding model. This closed-loop self-improvement mechanism drives the continuous co-evolution of both models.
2. We present the C-DREM that harnesses multiple grounding models to assess plan executability. By dynamically weighting reward based on each model's confidence score, C-DREM constructs a robust composite reward signal, which significantly enhances the stability and accelerates the convergence of the GRPO training process.
3. The experimental results indicate that the Co-EPG exhibits excellent generalization by solely relying on the benchmark dataset for self-iterative optimization. It outperforms existing state-of-the-art methods on both Multimodal-Mind2web (58.4%) and AndroidControl (83.1%) benchmarks.

2 Related Work

Recent advances in Large Language Models (LLMs) and Vision Language Models (VLMs) have laid a solid foundation for research on GUI agents (Liu et al. 2024; Achiam et al. 2023; Wang et al. 2024c; Bai et al. 2023). Current researches primarily revolve around two core views: architecture design and capability enhancement.

2.1 GUI Agent Architecture Design

GUI agent architectures have evolved from LLM-centric frameworks to end-to-end VLM-driven agents, and most recently, flexible modular systems. Early agents are built around LLMs (Gur et al. 2023; Zhao et al. 2024; Fu et al. 2024; Wang et al. 2024a), but they struggle with complex visual information. This requires auxiliary models for tasks like element selection (Deng et al. 2023) or information parsing (Lee et al. 2025). However, this reliance on text overlooks crucial visual and semantic cues (Zheng et al. 2024). Consequently, the field pivots towards VLMs to natively integrate visual data. These fall into two main categories: hybrid approaches (Yang et al. 2023; Lu et al. 2024), which use external tools to parse screen information for VLMs; and end-to-end VLM-driven agents (Niu et al. 2024; Qin et al. 2025a; He et al. 2024; Cheng et al. 2024). However, these models face new generalization challenges in diverse GUI environments (Gou et al. 2024). To overcome generalization issues, recent works focus on flexible, modular designs, which involve either decoupling planning and grounding for accuracy and flexibility (Zheng et al. 2024; Zhang et al. 2025a) or employing multi-agent collaboration for complex tasks (Liu et al. 2025b; Agashe et al. 2025).

2.2 GUI Agent Capability Enhancement

To enhance GUI agent capabilities, researches focus on two main fronts: data synthesis and training strategies.

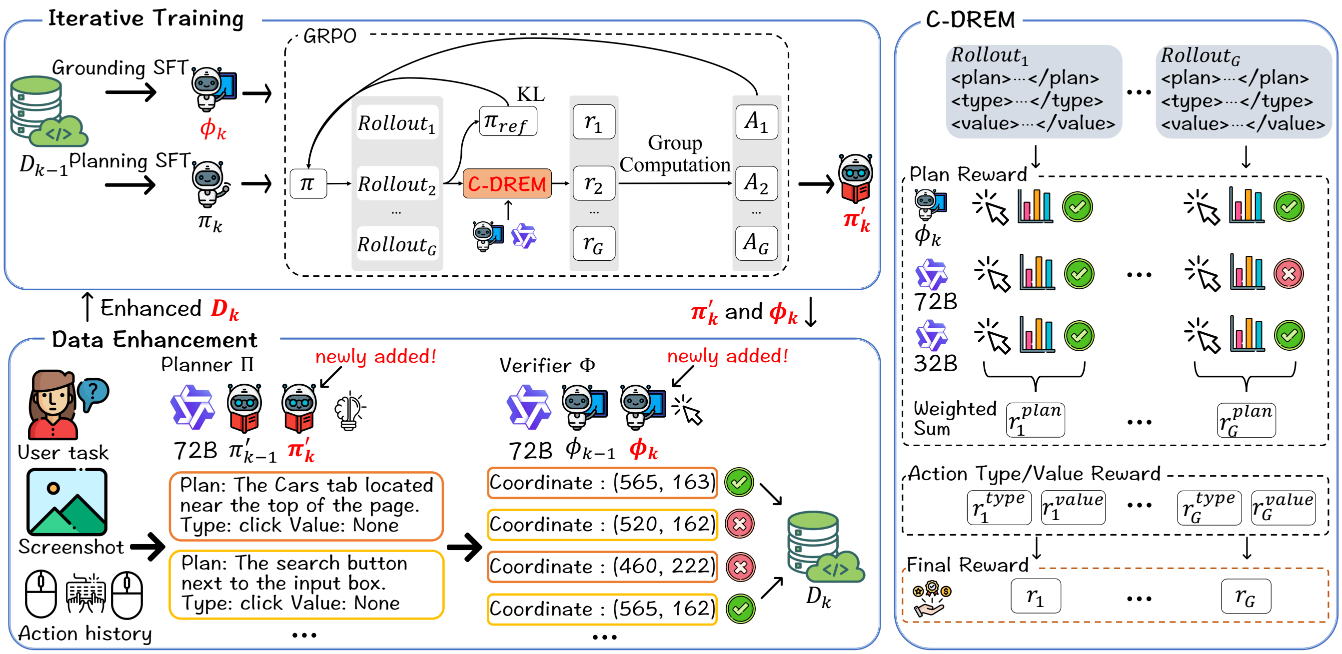
Automated Data Synthesis. Data synthesis often depends on significant human labor, leading to high costs. To replace costly manual annotation, various automatic methods have been proposed. Explorer (Pahuja et al. 2025) generates over 94K successful trajectories via dynamic exploration of web environments. AgentTrek (Xu et al. 2024a) simulates execution traces using web tutorials as step-by-step guides. Winclick (Hui et al. 2025) builds a 60k-sample dataset by identifying interactive elements from raw screenshots and generating corresponding natural language instructions. To improve data quality, Aguvis (Xu et al. 2024b) generates "inner monologues" to enhance logical reasoning.

Advanced Training Strategies. Researchers have explored two primary paths for training. The first involves optimizing Supervised Fine-Tuning (SFT), such as two-stage training (Xu et al. 2024b; Wu et al. 2024) and curriculum learning (Chen et al. 2025). However, the generalization capability of SFT-based agents is heavily dependent on the scale of the training data (Jiang et al. 2025). Additionally, AgentSymbiotic (Zhang et al. 2025b) proposes collaborative learning between large and small LLMs, though its advantages have not yet been applied to decoupled planning-and-grounding architectures. The second path leverages RL, such as rule-based rewards (Luo et al. 2025; Wei et al. 2025) or hybrid distillation-RL approaches (Liu et al. 2025c). More advanced agents aim to self-evolve by generating tasks from failures (Qi et al. 2024), using co-evolving world models (Fang et al. 2025), and applying attention-guided self-improvement (Yuan et al. 2025). These ideas are largely confined to end-to-end models and have not been integrated with modular architectures. *To the best of our knowledge, our work is the first to propose a co-evolution framework for decoupled architectures that iteratively refines planning and grounding capabilities to maximize data value.*

3 Preliminaries

We formulate the GUI task as a Partially Observable Markov Decision Process (POMDP) following (Xu et al. 2024b).

Definition 1 (POMDP). Formally, a POMDP is defined by a tuple ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{O}$, $\mathcal{F}$, $\mathcal{R}$), where $\mathcal{S}$ represents the complete state space, $\mathcal{O}$ is the set of partial observations the agent can perceive (e.g., visual screenshots and HTML content), and $\mathcal{A}$ denotes the space of feasible actions. $\mathcal{F}$ represents the state transition function, $\mathcal{F}(s, a, s') = P(s'|s, a)$, where is the probability of transitioning to state s' after taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. $\mathcal{R}$ is the reward function, $\mathcal{S} \times \mathcal{A} \to [0, 1]$, which specifies the reward for taking action a in state s. At each timestep t, the agent determines an action based on the current observation $o_t = (o_t^{vision}, o_t^{html}) \in \mathcal{O}$, the task description Q, and its interaction history $h_t = \{a_0, a_1, ..., a_{t-1}\}$. The action a_t is a composite tuple (a_t^{coor}, a_t^{type}, a_t^{value}), where a_t^{coor} represents



Plan: The Cars tab located near the top of the page. Type: click Value: None

Plan: The search button next to the input box. Type: click Value: None

Coordinate : (565, 163) ✓

Coordinate : (520, 162) ✗

Coordinate : (460, 222) ✗

Coordinate : (565, 162) ✓

Figure 1: Overview of our proposed Co-EPG framework. The framework drives the co-evolution of the planning model (π) and the grounding model (ϕ) through an optimization loop. This loop alternates between Iterative Training, which employs C-DREM to drive the co-evolution of both models, and Data Enhancement, which refines the dataset for the next iteration.

the target coordinates of action, a_t^{type} is the action type, and a_t^{value} is the action value. This process generates a complete interaction trajectory $T = \{(o_1, a_1), \ldots, (o_{|T|}, a_{|T|})\}$, where $|T|$ is the total number of steps.

4 Methodology

In this section, we propose a self-iterative training framework for **Co**-**E**volution of **P**lanning and **G**rounding (**Co-EPG**), built upon the planning model π and the grounding model ϕ, as illustrated in Figure 1.

4.1 P-G Dual-Model

Inspired by (Gou et al. 2024), we adopt a P-G dual-model architecture whose decoupled design allows each model to specialize in its respective function, thereby enabling the GUI agent to efficiently manage complex multi-step tasks. At each timestep t, the complex decision-making process is divided into two cooperative subtasks. Specifically, the planning model π acts as a high-level strategist. Given the current observation o_t, task description Q, and interaction history h_t, it generates a multi-part action decision, which comprises a textual plan p_t, an action type a_t^{type}, and a corresponding action value a_t^{value}, as follows:

$$p_t, a_t^{type}, a_t^{value} = \pi(Q, o_t, h_t). \tag{1}$$

Subsequently, the grounding model ϕ utilizes the specific plan p_t with the visual input from the vision observation (e.g., a screenshot) o_t^{vision} to predict the exact coordinates a_t^{coor} of the target element, as follows:

$$a_t^{coor} = \phi(o_t^{vision}, p_t). \tag{2}$$

The outputs of these two models are then combined to form the current action $a_t = (a_t^{coor}, a_t^{type}, a_t^{value})$ to be executed. This sequence of operations is repeated iteratively, generating a complete task-execution trajectory T.

4.2 Co-Evolving Optimization Loop

The self-iterative collaborative training loop of Co-EPG primarily includes the following two core steps: Iterative Training and Data Enhancement.

Iterative Training. Training is driven by the iterative dataset D_k. In each iteration k (where $k \geq 1$), we first fine-tune the model on the dataset D_{k-1} to obtain π_k and ϕ_k. Then, we refine the planning model π_k into π'_k through collaborative GRPO training, which is guided by our proposed C-DREM. The mechanism adaptively aggregates the rewards generated from an ensemble of the grounding model ϕ_k and VLMs (e.g., Qwen2.5-VL-72B-Instruct and Qwen2.5-VL-32B-Instruct), which is detailed further in Section 4.3. Guided by these grounding models, the planning model explores more successful strategies, which makes its plans more comprehensible. The grounding model ϕ_k, in turn, is further strengthened by fine-tuning on high-quality data distilled from the previous stage, which enhances its perception capabilities. This process iteratively aligns the capabilities of the planning and grounding models, ultimately yielding the planning model π'_k and the grounding model ϕ_k.

Data Enhancement. We develop a self-enhancing data evolution mechanism: Initially ($k = 0$), we form two specialized pools, each consisting of open-source VLMs: the Planner Π and the Verifier Φ. The Planner Π generates the specific plan p_t based on the current observation o_t, task description Q, and historical actions h_t. The Verifier Φ then validates each plan and constructs the initial dataset $(o_t, h_t, p_t, a_t) \in D_0$ by retaining only successfully verified plans. In subsequent iterations ($k \geq 1$), the updated planning model π'_k and grounding model ϕ_k participate in the data production process: the Planner Π incorporates π'_k to enhance planning diversity, and the Verifier Φ integrates ϕ_k to improve discrimination reliability. To balance effectiveness and efficiency, only the latest $\{\pi'_k, \pi'_{k-1}\}$ and $\{\phi_k, \phi_{k-1}\}$ are reserved for data production, maintaining the size of both pools. Through a self-evolution loop, the planning and grounding models achieve synergistic improvement. Ultimately, we obtain a more powerful GUI agent M_k= $\{\pi'_k, \phi_k\}$.

4.3 C-DREM

As described in Section 4.2, we facilitate synergy between the planning and grounding models through collaborative GRPO training. To build a more comprehensive reward signal (Liu et al. 2025a), we evaluate the planning model's output on its plan, action type, and action value. For clarity, we omit the time step subscript t in the following contents, as all formulas are defined under a unified time step.

Plan Reward. A key challenge for the planning model during GRPO training is that the quality of a generated plan cannot be directly evaluated, because its effectiveness is ultimately determined by whether the grounding model can use this plan to accurately locate the target element. As a result, we use the grounding model's prediction accuracy as a reward to guide the optimization of the planning model. The accuracy, denoted as Acc^{plan}, is calculated based on whether the coordinates a^{coor} predicted by the grounding model fall within the target's bounding box $bbox$:

$$Acc^{plan} = \begin{cases} 1, & \text{if } a^{coor} \in bbox, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

To address the inherent bias and poor performance of single reward model on out-of-distribution data, we propose C-DREM, a confidence-based dynamic reward ensemble mechanism. The core idea of C-DREM is to aggregate the collective intelligence of diverse grounding models, which include both open-source models and the grounding model ϕ_k generated during our iterative training process. The reward r^{plan} is defined as a weighted sum:

$$r^{plan} = \sum_{j=1}^{N} w_j \cdot Acc_j^{plan}, \tag{4}$$

where Acc_j^{plan} represents the reward from the j-th grounding model (for $j = 1, \ldots, N$), N is the number of grounding models, and w_j is the corresponding weight. The weight w_j is determined by two components: a static prior σ_j and a dynamic confidence score c_j:

$$w_j = \frac{\exp(\sigma_j \cdot c_j)}{\sum_{n=1}^{N} \exp(\sigma_n \cdot c_n)}. \tag{5}$$

Here, the static prior σ_j is set higher for our trained grounding model, reflecting its critical importance in production deployment. The dynamic confidence score c_j is calculated as the sum of the log-likelihoods of the predicted coordinate token τ of a_j^{coor}, normalized by its length L:

$$c_j = \frac{1}{L} \sum_{l=1}^{L} \log P(\tau_l \mid o, h, p). \tag{6}$$

Next, we calculate rewards based on action type and action value according to (Luo et al. 2025).

Action Type Reward. The action type reward r^{type} depends on whether the predicted action type a^{type} exactly matches the ground truth action type gt^{type}:

$$r^{type} = \begin{cases} 1, & \text{if } a^{type} = gt^{type}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Action Value Reward. The action value reward r^{value} is calculated based on the F_1 score between the predicted value a^{value} and the ground truth value gt^{value}, which is called as:

$$r^{value} = \begin{cases} 1, & \text{if } F_1(a^{value}, gt^{value}) > 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Final Reward. The final reward r_i of the i-th generation for the GUI tasks is calculated by r_i^{plan}, r_i^{type} and r_i^{value}:

$$r_i = \begin{cases} 0, & \text{if } r_i^{type} = 0 \text{ or } r_i^{value} = 0, \\ r_i^{plan}, & \text{otherwise.} \end{cases} \tag{9}$$

Group Computation. Subsequently, we normalize these rewards across the G generated $Rollouts$ to compute the advantage A_i, which subsequently serves as the objective for GRPO policy optimization (Shao et al. 2024):

$$A_i = \frac{r_i - mean(\{r_1, r_2, \cdots, r_G\})}{std(\{r_1, r_2, \cdots, r_G\})}. \tag{10}$$

5 Experiments

5.1 Datasets and Metrics

We conduct systematic evaluations on two main benchmarks: Multimodal-Mind2Web (Deng et al. 2023) for web interactions and AndroidControl (Li et al. 2024) for mobile applications. For Multimodal-Mind2Web, we report element accuracy (Ele.Acc), operation F1 (Op.F1), and step success rate (Step SR), while for AndroidControl, we use the standard metric of step accuracy (Step Acc). Detailed benchmark information is provided in Appendix. A.

5.2 Implementation Details

We implement the experiments using PyTorch 2.6.0 on a Linux server equipped with 984GB RAM, Intel Xeon Platinum 8369B CPU @ 2.90GHz, and Nvidia A100 Tensor Core 80GB GPUs. We select Qwen2.5-VL as our backbone in experiments, with the training pipeline implemented based on the MS-SWIFT framework. All results represent the average of three independent runs to ensure credibility. Training details are provided in Appendix. B.

Planner	Grounder	Cross-Task			Cross-Website			Cross-Domain			Avg SR
		Ele.Acc	Op.F1	Step SR	Ele.Acc	Op.F1	Step SR	Ele.Acc	Op.F1	Step SR	
GPT-4	Choice	46.4	73.4	40.2	38.0	67.8	32.4	42.4	69.3	36.8	36.5
	SoM	29.6	-	20.3	20.1	-	13.9	27.0	-	23.7	19.3
GPT-4o	SeeClick	32.1	-	-	33.1	-	-	33.5	-	-	-
	UGround-V1-2B	48.6	-	-	47.6	-	-	47.7	-	-	-
	UGround-V1-7B	50.7	-	-	48.1	-	-	48.5	-	-	-
GPT-4V	OmniParser	42.4	87.6	39.4	41.0	84.8	36.5	45.5	85.7	42.0	39.3
Explorer-4B		53.4	88.1	50.7	55.6	89.5	51.4	49.8	88.8	47.2	49.8
Explorer-7B		56.5	90.3	53.2	60.5	90.7	56.7	55.7	90.4	53.0	54.3
AgentTrek-7B		60.8	88.9	55.7	57.6	88.1	51.4	56.0	87.5	52.6	53.2
SeeClick-9.6B		28.3	87.0	25.5	21.4	80.6	16.4	23.2	84.8	20.8	20.9
AGUVIS-7B		64.2	89.8	60.4	60.7	88.1	54.6	**60.4**	89.2	**56.6**	57.2
Co-EPG-Web-3B		57.7	88.8	53.1	56.9	87.7	51.1	53.6	89.6	50.0	51.4
Co-EPG-Web-7B		**66.3**	**92.4**	**61.9**	**62.3**	**91.7**	**58.1**	59.3	**92.2**	55.3	**58.4**

Table 1: Performance on Multimodal-Mind2Web. We bold the best results and underline the second-best performance.

5.3 Main Results

Multimodal-Mind2Web Results. The experimental results in Table 1 highlight the self-improvement capability of our proposed Co-EPG framework, which achieves state-of-the-art performance on web tasks. Specifically, the Co-EPG-Web-7B model achieves 58.4% on average Step SR of three subtasks, outperforming both Explorer-7B (Pahuja et al. 2025) (54.3%) and the previous leading model, AGUVIS-7B (Xu et al. 2024b) (57.2%). Notably, unlike AGUVIS-7B, which relies on extensive auxiliary data construction, Co-EPG achieves superior performance using only the original benchmark data. Co-EPG also shows strong generalization at smaller scales. Co-EPG-Web-3B outperforms the previous state-of-the-art model with similar parameters, Explorer-4B (Pahuja et al. 2025), by nearly 1.6% on average Step SR. Based on the experimental results across varying model scales, this consistent performance gain demonstrates that the co-evolution of planning and grounding models enables agents to learn from more diverse and higher-quality data, without requiring external data.

AndroidControl Results. Table 2 presents the experimental results of Co-EPG on the AndroidControl benchmark for the mobile task. Co-EPG-Mob-7B achieves the best performance with 83.1% on average Step Acc of high-level and low-level tasks, demonstrating a significant 1.4% advantage over the previous state-of-the-art UI-TARS-7B (Qin et al. 2025b). This strong performance extends to smaller models as well: the Co-EPG-Mob-3B variant also performs excellently with a score of 81.8%, maintaining a competitive advantage compared to InfiGUI-R1-3B (Liu et al. 2025c) (81.6%). These experiments validate the effectiveness and generalization of Co-EPG across diverse GUI environments.

Planner	Grounder	Step Acc		Avg Acc
		High	Low	
GPT-4o	SeeClick	41.8	52.8	47.3
GPT-4o	UGround-v1-2B	50.0	65.0	57.5
GPT-4o	UGround-v1-7B	49.8	66.2	58.0
AGUVIS-7B		61.5	80.5	71.0
AGUVIS-72B		66.4	84.4	75.4
OS-Atlas-4B		67.5	80.6	74.1
OS-Atlas-7B		71.2	85.2	78.2
GUI-R1-3B		46.6	64.4	55.5
GUI-R1-7B		51.7	66.5	59.1
UI-R1-3B		45.4	66.4	55.9
UI-TARS-2B		68.9	89.3	79.1
UI-TARS-7B		72.5	90.8	81.7
InfiGUI-R1-3B		71.1	**92.1**	81.6
Co-EPG-Mob-3B		73.4	90.2	81.8
Co-EPG-Mob-7B		**74.2**	92.0	**83.1**

Table 2: Performance on AndroidControl. We bold the best results and underline the second-best performance.

5.4 Ablation Study

The effectiveness of the Co-EPG framework arises from three synergistic design principles: the P-G dual-model architecture, the self-iterative loop for continuous evolution, and the confidence-based dynamic reward ensemble mechanism (C-DREM) for precise and adaptive guidance. In this section, we conduct three iterations and use M_k (detailed in Section 4.2) to represent the agent after each iteration.

Impact of P-G Dual-Model. We validate the contributions of the P-G dual-model by analyzing model variants with and without the decoupling structure mediated by planning instructions. Specifically, the performance of the P-G dual-model is compared with that of an end-to-end model under fine-tuning training. The experimental results, as shown in Table 3, demonstrate that the decoupled architecture improves performance by 3.4% over the end-to-end approach, confirming its effectiveness.

Impact of Iterative Evolution. To evaluate the synergistic co-evolution of Co-EPG, we track the performance of Co-EPG-Web-3B/7B across multiple iterations. As illustrated in Figure 2, both models show clear and steady improve-

Method	Avg SR
End2End	50.1
Co-EPG-Web-7B-M_1	**53.5**

Table 3: Performance Comparison: Decoupled P-G Dual-Model Architecture vs. End-to-End.

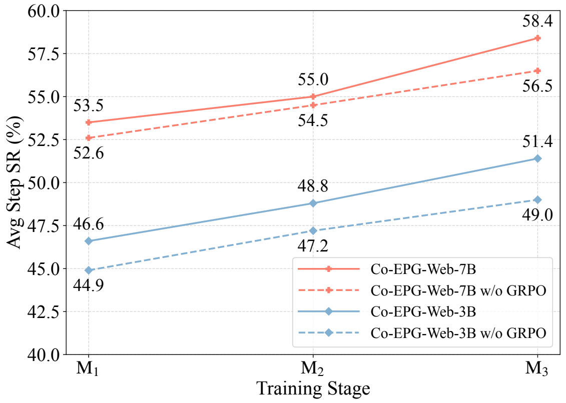


Figure 2: Effectiveness of the self-iterative evolution mechanism in Co-EPG on Multimodal-Mind2Web.

ments, confirming the performance gains from our self-evolutionary framework. To determine whether this gain is from data iteration alone or whether GRPO collaborative training also plays a crucial role, we remove GRPO and rely solely on SFT. Although the models without GRPO (dashed lines) also improve, they significantly underperform the full Co-EPG architecture. This result demonstrates that data iteration and GRPO collaborative training are dual drivers of this evolution. Specifically, GRPO collaborative training acts as an accelerator, providing the precise exploration rewards required to transcend the performance limitations inherent to purely data-centric refinement.

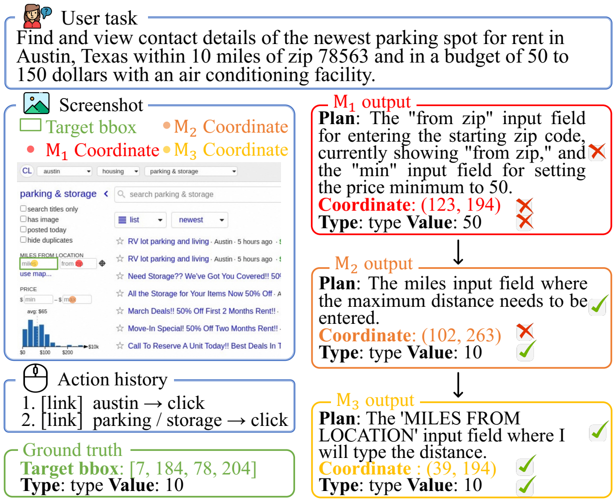
Impact of C-DREM. To validate the effectiveness of key components in C-DREM, we conduct ablation studies with the following variants:

- **w/o C-DREM**. Uses the trained grounding model as the single reward model.
- **w/o Confidence & Prior Weights**. Uses average weighting but removes confidence and prior weights.
- **w/o Confidence Weights**. Uses prior weighting but removes the confidence weights.

We establish a performance baseline by using a single grounding model as the reward model (w/o C-DREM). Table 4 demonstrates three key findings: Firstly, average weighting (w/o Confidence & Prior Weights) already shows 0.51% improvement over the baseline. Secondly, prior weighting (w/o Confidence Weights) achieves a further 0.66% gain but with static weights. These verify the ensemble's inherent advantage in generating robust rewards. Finally, our complete C-DREM achieves a 1.91% improvement, highlighting the critical role of confidence-based dynamic weighting. These findings confirm that C-DREM can generate more adaptive and precise rewards, addressing the limitations of static weights, and collectively validate the effectiveness of C-DREM in improving model performance.

Method	Avg SR
w/o C-DREM	56.50
w/o Confidence & Prior Weights	57.01
w/o Confidence Weights	57.67
Co-EPG-Web-7B-M_3	**58.41**

Table 4: Comparative Analysis of C-DREM in GRPO Collaborative Training.



Find and view contact details of the newest parking spot for rent in Austin, Texas within 10 miles of zip 78563 and in a budget of 50 to 150 dollars with an air conditioning facility.

1. [link] austin → click
2. [link] parking / storage → click

Target bbox: [7, 184, 78, 204]
Type: type **Value**: 10

Plan: The "from zip" input field for entering the starting zip code, currently showing "from zip," and the "min" input field for setting the price minimum to 50. ✗
Coordinate: (123, 194) ✗
Type: type **Value**: 50 ✗

Plan: The miles input field where the maximum distance needs to be entered. ✓
Coordinate: (102, 263) ✗
Type: type **Value**: 10 ✓

Plan: The 'MILES FROM LOCATION' input field where I will type the distance. ✓
Coordinate : (39, 194) ✓
Type: type **Value**: 10 ✓

Figure 3: An Evolution Example of Co-EPG.

5.5 Case Study

As shown in Figure 3, we visualize evolution in the capabilities of the planning and grounding models across three stages. In M_1 stage, the planning model not only produces an ambiguous plan by merging multiple UI targets but also predicts an incorrect action value. In M_2 stage, the planning model is able to identify the correct UI element and action value for the task, however, due to the misalignment between the capabilities of the planning and grounding model, the predicted coordinates are still incorrect. Moreover, in M_3 stage, the planning model achieves high semantic accuracy by generating plans with specific UI text (”MILES FROM LOCATION”) rather than generic descriptions (”miles input field”), leading to comprehensive success. This progression from ambiguous to precise planning, along with the improved capabilities of the grounding model, greatly boosts the model's overall performance on GUI tasks.

5.6 Efficiency Study

Data Efficiency. Figure 4 powerfully demonstrates the superior data efficiency of our Co-EPG framework. Remarkably, our Co-EPG-Web-7B surpasses the previous state-of-the-art model AGUVIS-7B on average Step SR while utiliz-

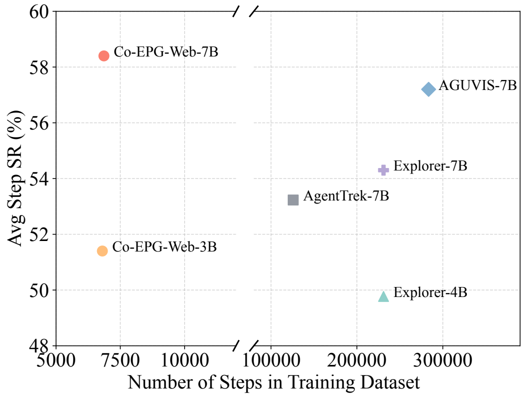


Figure 4: Performance and data efficiency on the Multimodal-Mind2Web benchmark.

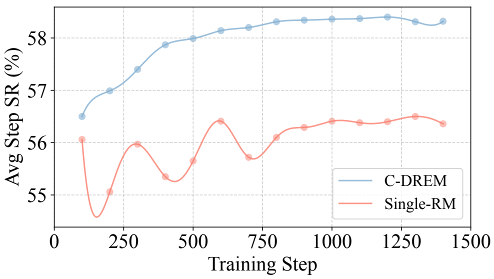


Figure 5: The performance curves of different reward mechanisms evaluated on Multimodal-Mind2Web.

ing only 2.42% of the labeled step data (6862 vs. 283500). This achievement validates the effectiveness of Co-EPG's innovative data value mining mechanism, particularly highlighting its advantages in low-resource scenarios.

C-DREM Efficiency. On the Multimodal-Mind2Web dataset, we track Step SR to compare the efficiency of the C-DREM with a single grounding model. In Figure 5, our C-DREM plays a crucial role in enhancing learning stability and accelerating convergence efficiency. Relying on a single grounding model inevitably leads to cognitive blind spots, resulting in high-variance rewards during the agent's exploration process. This variability causes significant policy fluctuations and inefficient learning. In contrast, our proposed mechanism mitigates noise, enhances exploration efficiency, and speeds up the planning model's convergence by integrating multiple grounding models.

5.7 Analysis

Evolution of Data Quality. We evaluate the data quality evolution of the Co-EPG across two dimensions: purity and diversity. The purity metric is the proportion of plans successfully executed by grounding models, and the diversity metric is the average number of generated plans per task. As shown in Figure 6, both core metrics exhibit significant upward trends as the iteration progresses, with purity improving by 8.84% and the diversity metric increasing by nearly 4. This result confirms the framework's data self-enhancement and highlights its significant potential for self-evolution.

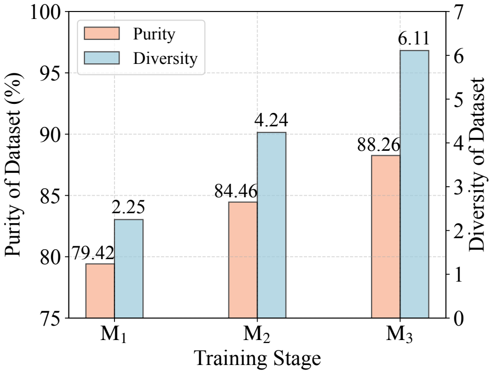


Figure 6: Iterative enhancement of data quality in Co-EPG.

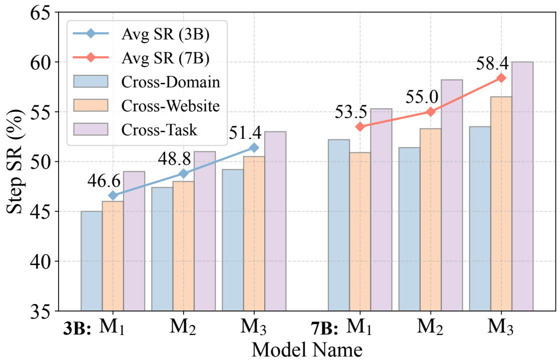


Figure 7: Iterative improvement of Co-EPG-Web models across all tasks on the Multimodal-Mind2Web.

Improvement across Different Tasks. Figure 7 demonstrates that on the Multimodal-Mind2web benchmark, both our 3B and 7B models exhibit significant and stable Step SR growth across all tasks. This steady improvement validates the robustness of our co-evolutionary learning process. Notably, this trend holds even for challenging cross-website and cross-domain tasks, demonstrating the framework's ability to foster strong generalization rather than mere memorization. Additionally, the consistent performance across various model scales demonstrates the effectiveness of Co-EPG.

6 Conclusion

In this paper, we propose Co-EPG, a self-iterative training framework for Co-Evolution of Planning and Grounding. Co-EPG utilizes planning instructions as interaction media, successfully achieving collaborative GRPO training between planning and grounding models. Extensive experiments demonstrate that Co-EPG exhibits state-of-the-art performance across both web and mobile tasks, showing stable performance improvement through iterative rounds. We believe that applying Co-EPG with other data synthesis techniques could unlock even greater data potential, and we hope our work can inspire future research in GUI agents.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR21M2, including the AIP Challenge Program.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Agashe, S.; Wong, K.; Tu, V.; Yang, J.; Li, A.; and Wang, X. E. 2025. Agent s2: A compositional generalist-specialist framework for computer use agents. *arXiv preprint arXiv:2504.00906*.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Chen, W.; Cui, J.; Hu, J.; Qin, Y.; Fang, J.; Zhao, Y.; Wang, C.; Liu, J.; Chen, G.; Huo, Y.; et al. 2025. GUICourse: From General Vision Language Model to Versatile GUI Agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 21936–21959.

Cheng, K.; Sun, Q.; Chu, Y.; Xu, F.; Li, Y.; Zhang, J.; and Wu, Z. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*.

Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; and Su, Y. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36: 28091–28114.

Fang, T.; Zhang, H.; Zhang, Z.; Ma, K.; Yu, W.; Mi, H.; and Yu, D. 2025. WebEvolver: Enhancing Web Agent Self-Improvement with Coevolving World Model. *arXiv preprint arXiv:2504.21024*.

Fu, Y.; Kim, D.-K.; Kim, J.; Sohn, S.; Logeswaran, L.; Bae, K.; and Lee, H. 2024. Autoguide: Automated generation and selection of context-aware guidelines for large language model agents. *Advances in Neural Information Processing Systems*, 37: 119919–119948.

Gou, B.; Wang, R.; Zheng, B.; Xie, Y.; Chang, C.; Shu, Y.; Sun, H.; and Su, Y. 2024. Navigating the digital world as humans do: Universal visual grounding for gui agents. *arXiv preprint arXiv:2410.05243*.

Gur, I.; Furuta, H.; Huang, A.; Safdari, M.; Matsuo, Y.; Eck, D.; and Faust, A. 2023. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*.

He, H.; Yao, W.; Ma, K.; Yu, W.; Dai, Y.; Zhang, H.; Lan, Z.; and Yu, D. 2024. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*.

Hui, Z.; Li, Y.; Chen, T.; Banbury, C.; Koishida, K.; et al. 2025. Winclick: Gui grounding with multimodal large language models. *arXiv preprint arXiv:2503.04730*.

Jiang, T.; Li, S.; Song, Y.; Zhang, L.; Zhu, H.; Zhao, Y.; Xu, X.; Taura, K.; and Wang, H. H. 2025. Importance-Aware Data Selection for Efficient LLM Instruction Tuning. arXiv:2511.07074.

Lee, D.; Lee, J.; Kim, K.; Tack, J.; Shin, J.; Teh, Y. W.; and Lee, K. 2025. Learning to contextualize web pages for enhanced decision making by LLM agents. *arXiv preprint arXiv:2503.10689*.

Li, K.; Zhang, Z.; Yin, H.; Zhang, L.; Ou, L.; Wu, J.; Yin, W.; Li, B.; Tao, Z.; Wang, X.; et al. 2025. WebSailor: Navigating Super-human Reasoning for Web Agent. *arXiv preprint arXiv:2507.02592*.

Li, W.; Bishop, W. E.; Li, A.; Rawles, C.; Campbell-Ajala, F.; Tyamagundlu, D.; and Riva, O. 2024. On the effects of data scale on ui control agents. *Advances in Neural Information Processing Systems*, 37: 92130–92154.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Liu, X.; Liang, D.; Shan, H.; Liu, P.; Liu, Y.; Wu, M.; Li, Y.; Wu, X.; Miao, L.; Shen, J.; et al. 2025a. Structural Reward Model: Enhancing Interpretability, Efficiency, and Scalability in Reward Modeling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 672–685.

Liu, Y.; Li, P.; Wei, Z.; Xie, C.; Hu, X.; Xu, X.; Zhang, S.; Han, X.; Yang, H.; and Wu, F. 2025b. Infiguiagent: A multimodal generalist gui agent with native reasoning and reflection. *arXiv preprint arXiv:2501.04575*.

Liu, Y.; Li, P.; Xie, C.; Hu, X.; Han, X.; Zhang, S.; Yang, H.; and Wu, F. 2025c. Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative reasoners. *arXiv preprint arXiv:2504.14239*.

Lu, Y.; Yang, J.; Shen, Y.; and Awadallah, A. 2024. Omniparser for pure vision based gui agent. *arXiv preprint arXiv:2408.00203*.

Luo, R.; Wang, L.; He, W.; and Xia, X. 2025. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*.

Niu, R.; Li, J.; Wang, S.; Fu, Y.; Hu, X.; Leng, X.; Kong, H.; Chang, Y.; and Wang, Q. 2024. Screenagent: A vision language model-driven computer control agent. *arXiv preprint arXiv:2402.07945*.

OpenAI. 2025. Introducing deep research. *https://openai.com/index/ introducing-deep-research/*.

Pahuja, V.; Lu, Y.; Rosset, C.; Gou, B.; Mitra, A.; Whitehead, S.; Su, Y.; and Awadallah, A. 2025. Explorer: Scaling exploration-driven web trajectory synthesis for multimodal web agents. *arXiv preprint arXiv:2502.11357*.

Qi, Z.; Liu, X.; Iong, I. L.; Lai, H.; Sun, X.; Zhao, W.; Yang, Y.; Yang, X.; Sun, J.; Yao, S.; et al. 2024. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. *arXiv preprint arXiv:2411.02337*.

Qin, Y.; Ye, Y.; Fang, J.; Wang, H.; Liang, S.; Tian, S.; Zhang, J.; Li, J.; Li, Y.; Huang, S.; et al. 2025a. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*.

Qin, Y.; Ye, Y.; Fang, J.; Wang, H.; Liang, S.; Tian, S.; Zhang, J.; Li, J.; Li, Y.; Huang, S.; et al. 2025b. Ui-tars: Pioneering automated gui interaction with native agents, 2025. *URL https://arxiv. org/abs/2501.12326*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Wang, J.; Jiang, R.; Yang, C.; Wu, Z.; Onizuka, M.; Shibasaki, R.; Koshizuka, N.; and Xiao, C. 2024a. Large language models as urban residents: An llm agent framework for personal mobility generation. *Advances in Neural Information Processing Systems*, 37: 124547–124574.

Wang, J.; Xu, H.; Jia, H.; Zhang, X.; Yan, M.; Shen, W.; Zhang, J.; Huang, F.; and Sang, J. 2024b. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *Advances in Neural Information Processing Systems*, 37: 2686–2710.

Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. 2024c. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Wei, Z.; Yao, W.; Liu, Y.; Zhang, W.; Lu, Q.; Qiu, L.; Yu, C.; Xu, P.; Zhang, C.; Yin, B.; et al. 2025. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. *arXiv preprint arXiv:2505.16421*.

Wu, Z.; Wu, Z.; Xu, F.; Wang, Y.; Sun, Q.; Jia, C.; Cheng, K.; Ding, Z.; Chen, L.; Liang, P. P.; et al. 2024. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*.

Xu, Y.; Lu, D.; Shen, Z.; Wang, J.; Wang, Z.; Mao, Y.; Xiong, C.; and Yu, T. 2024a. Agenttrek: Agent trajectory synthesis via guiding replay with web tutorials. *arXiv preprint arXiv:2412.09605*.

Xu, Y.; Wang, Z.; Wang, J.; Lu, D.; Xie, T.; Saha, A.; Sahoo, D.; Yu, T.; and Xiong, C. 2024b. Aguvis: Unified pure vision agents for autonomous gui interaction. *arXiv preprint arXiv:2412.04454*.

Yang, J.; Zhang, H.; Li, F.; Zou, X.; Li, C.; and Gao, J. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Yuan, X.; Zhang, J.; Li, K.; Cai, Z.; Yao, L.; Chen, J.; Wang, E.; Hou, Q.; Chen, J.; Jiang, P.-T.; et al. 2025. Enhancing Visual Grounding for GUI Agents via Self-Evolutionary Reinforcement Learning. *arXiv preprint arXiv:2505.12370*.

Zhang, C.; Li, L.; He, S.; Zhang, X.; Qiao, B.; Qin, S.; Ma, M.; Kang, Y.; Lin, Q.; Rajmohan, S.; et al. 2024. Ufo: A ui-focused agent for windows os interaction. *arXiv preprint arXiv:2402.07939*.

Zhang, D.; Rama, B.; Ni, J.; He, S.; Zhao, F.; Chen, K.; Chen, A.; and Cao, J. 2025a. Litewebagent: The open-source suite for vlm-based web-agent applications. *arXiv preprint arXiv:2503.02950*.

Zhang, R.; Qiu, M.; Tan, Z.; Zhang, M.; Lu, V.; Peng, J.; Xu, K.; Agudelo, L. Z.; Qian, P.; and Chen, T. 2025b. Symbiotic cooperation for web agents: Harnessing complementary strengths of large and small llms. *arXiv preprint arXiv:2502.07942*.

Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.-J.; and Huang, G. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19632–19642.

Zhao, D.; Ma, L.; Wang, S.; Wang, M.; and Lv, Z. 2025. COLA: A Scalable Multi-Agent Framework For Windows UI Task Automation. *arXiv preprint arXiv:2503.09263*.

Zheng, B.; Gou, B.; Kil, J.; Sun, H.; and Su, Y. 2024. Gpt-4v(ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.