Towards Omni-dimensional GUI Agent Navigation with Masked Trajectory Prediction

Anonymous ACL submission

Abstract

Graphical User Interface (GUI) Agents autonomously interact with software to fulfill user requests, where GUI navigation stands out as the most critical and challenging capability. Mastering this capability demands a complex synergy of step-wise decision-making, state-action alignment, and long-horizon planning. While directly mixing these corresponding navigation tasks seems intuitive to simultaneously acquire these skills, such a direct combination is severely bottlenecked by inconsistent optimization objectives and profound data heterogeneity. To overcome these barriers, we propose the MaP (stands for "**M**asked Tr**a**jectory **P**rediction"), a unified framework that seamlessly harmonizes divergent GUI navigation tasks. By modeling multi-turn GUI interactions as a trajectory and defining training objectives through component masking and prediction, MaP shifts the optimization from task-specific marginal distributions to a consistent objective. Furthermore, to handle the data heterogeneity across multiple navigation tasks, we design a role-aware adapter learning module that dynamically routes each token to a specialized representation space. Extensive experiments on five representative GUI navigation benchmarks demonstrate that MaP effectively mitigates gradient conflicts and significantly outperforms the direct mixture training, establishing a robust paradigm for multi-task GUI navigation.

1 Introduction

GUI Agents, designed to autonomously navigate graphical user interfaces to fulfill user requests, represent a promising frontier in building practical AI assistants (Wang et al., 2024b; Nguyen et al., 2025). Among the capabilities required for effective GUI agents, navigation stands out as the most critical and challenging, as it demands decomposing high-level user instructions (Xu et al., 2024) , aligning intermediate actions with screenshot changes (Shen et al., 2024; Qin et al., 2025), and analyzing the visual context for each interaction step (Wu et al., 2024b; Zhang et al., 2025a).

Building upon prior studies, we categorize existing navigation strategies into three core dimensions, as depicted in Figure 1. a). Step-wise decision (Wu et al., 2024b; Zhang et al., 2025a) provides the current screenshot and low-level instruction, requiring the GUI Agent to interpret UI semantics and generate structured reasoning traces. b). State-action alignment (Shen et al., 2024; Qin et al., 2025) tasks GUI Agents to predict the intermediate user action given two consecutive states, capturing the correspondence between actions and visual changes. c). Long-horizon planning (Xu et al., 2024) formulates GUI tasks as dialogue-like interactions, where GUI Agents leverage sequential visual contexts to decompose complex user instructions into step-wise plans.

While these paradigms have advanced GUI Agents along isolated dimensions, real-world GUI interaction inherently requires their joint competence. A straightforward solution is mixture training across diverse navigation tasks. However, as illustrated in Figure 2 Left, direct mixture yields only marginal improvements. We attribute this to two fundamental challenges: 1). **Inconsistent training objectives**. The three paradigms each capture distinct marginal distributions of GUI interactions, corresponding to different partial dependencies among states, reasoning traces, and interaction histories. Step-wise decision restricts its view to the current state; state-action alignment neglects the broader task context; and long-horizon planning often lacks fine-grained visual grounding for precise actions. 2). **Data heterogeneity.** The resulting corpora differ in reasoning styles, annotation protocols, and interface domains, hindering unified modeling.

To address these challenges, we propose MaP (stands for "**M**asked Tr**a**jectory **P**rediction"), a unified framework that enforces a consistent training

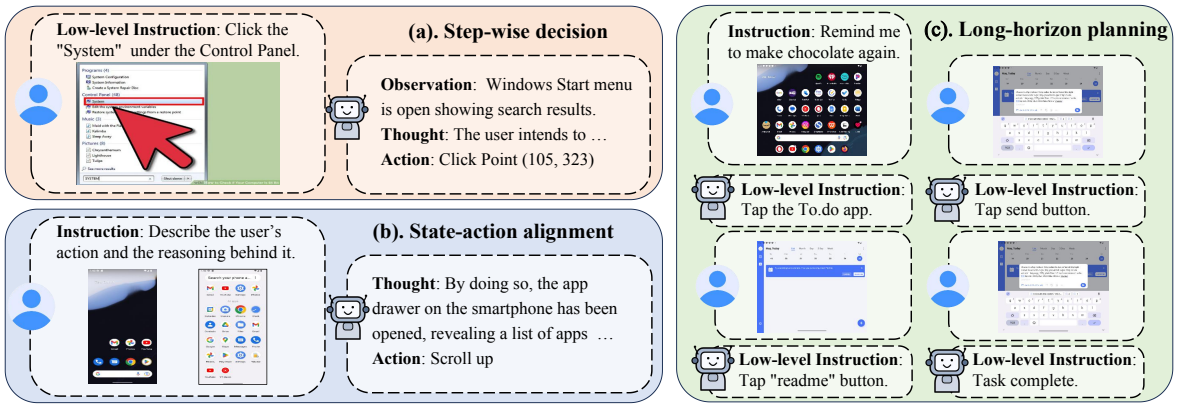


Figure 1: Overview of the three core dimensions of GUI navigation capabilities, which encompass (a) reasoning over current visual semantics, (b) understanding state-action alignment, and (c) maintaining long-horizon task consistency.

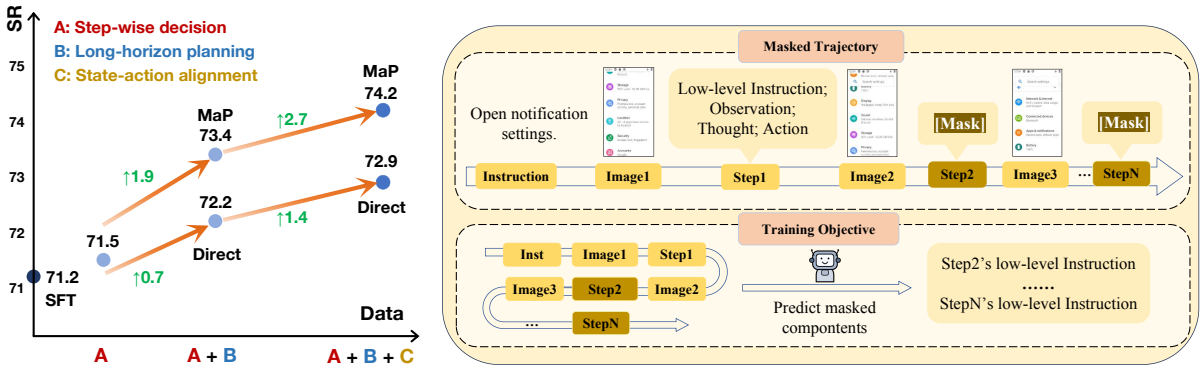


Figure 2: **Left:** Direct mixture training vs. MaP on the AndroidControl-High benchmark (Zhang et al., 2024). The ↑ denotes the performance improvements over the single step-wise decision paradigm. **Right:** An overview of MaP. By masking arbitrary trajectory contents and predicting the counterparts, MaP unifies heterogeneous GUI corpora under a consistent training objective.

objective and handles data heterogeneity across diverse GUI navigation tasks, as shown in Figure 2 Right. MaP formalizes multi-turn GUI interactions as trajectories and substitutes arbitrary components with [`Mask`] tokens, requiring the GUI Agent to predict the masked parts auto-regressively. By casting different paradigms into the same trajectory masking formulation, MaP shifts the optimization from task-specific marginal dependencies to a unified objective over complete trajectories, resolving the inconsistency among previous training objectives. As illustrated in Figure 2 Left, MaP unifies all three navigation paradigms and outperforms their pairwise combinations. Beyond the task formulation, we introduce a role-aware adapter learning module to tackle data heterogeneity, which dynamically routes each token to a specialized representation space based on its semantic role. Extensive experiments across multiple public benchmarks demonstrate the effectiveness of MaP, achieving significant improvements over existing methods.

The main contributions of our work are threefold:

- We identify that GUI Agents fundamentally benefit from multiple complementary navigation capabilities, and reveal that inconsistent optimization objectives and heterogeneous corpora severely hinder their effective composition through direct mixture training.
- We propose MaP, a unified mixture training framework that establishes a consistent training objective through masked trajectory prediction, complemented by a role-aware adapter learning module to address data heterogeneity.
- We conduct comprehensive experiments on five representative GUI navigation benchmarks, including AndroidWorld, AndroidControl, GUI-Odyssey, AITZ, and Mind2Web, verifying the effectiveness and generalization of MaP.

2 Related Work

2.1 GUI Agent

The advancements in LLMs and LVLMs have significantly accelerated the development of GUI Agents (Wang et al., 2024a; Bai et al., 2025; OpenAI, 2024). Early attempts parse GUIs into source code for LLM-based action inference (Shi et al., 2017; Kim et al, 2023; Chen et al., 2024), but their reliance on internal APIs limits applicability to commercial software, prompting a shift toward purely vision-based agents.

Recent GUI agents such as UI-TARS (Qin et al., 2025; Wang et al., 2025), Show-UI (Qin et al., 2025), and CogAgent (Hong et al., 2024) employ LVLMs to predict actions conditioned on GUI screenshots, focusing on two core abilities: interpreting GUI contexts and imitating human actions. For GUI interpretation, SeeClick (Cheng et al., 2024) pioneers pure-visual GUI grounding with an automated data construction pipeline, while GUI-R1 (Luo et al., 2025) and InfiGUI-R1 (Liu et al., 2025) further enhance visual understanding through reward design and action-centric deliberate reasoning, respectively. For action imitation, subsequent works (Zhang et al., 2025a; Xu et al., 2024) leverage human-annotated data and instructional tutorials to improve navigation capabilities.

2.2 GUI Corpus

GUI corpora encompass diverse user interactions across platforms such as mobile, web, and desktop. Android in The Wild (Rawles et al., 2023) introduces a large-scale 715k-episode GUI sequence from Android devices. Subsequent work extends the data with CoT annotations (Zhang et al., 2024) and cross-app navigation scenarios (Lu et al., 2024) to better emulate real user experiences.

However, the limited scale of manually annotated data remains insufficient for training LVLMs. To address this, existing efforts collect GUI data from instructional videos and generate synthetic trajectories for large-scale training. Mobile3M (Wu et al., 2024a) initiates this direction with 20 million synthetic interactions, MONDAY (Jang et al., 2025) introduces 313k annotated frames from instructional videos, and VideoAgentTrek (Lu et al., 2025) automatically mines training data from publicly available screen-recorded videos at web scale.

3 Method

3.1 Overview

Figure 2 Right illustrates the architecture of MaP, a unified framework proposed to resolve the inconsistent training objectives and data heterogeneity prevalent in direct mixture training. MaP is achieved through two core components: 1). Masked Trajectory Prediction task, which treats each GUI sequence as a trajectory and requires the GUI Agent to predict its masked components to unify diverse task-specific goals into the consistent objective. 2). The role-aware adapter learning module, which incorporates token-wise adapters to effectively handle divergent optimization directions introduced from heterogeneous GUI corpora.

3.2 Mask Trajectory Prediction

The core insight of our mixture training framework lies in establishing a consistent objective capable of accommodating heterogeneous GUI corpora. To achieve this, we regard the multi-turn interaction between users and GUI interfaces as a multi-modal trajectory, which may contain a user instruction, a sequence of screenshots, actions, and the associated reasoning processes. Notably, under this structure, applying a mask at any position within the trajectory, such as on actions or specific components of the reasoning steps, can effectively align diverse training tasks under the same objective.

At its core, masked trajectory prediction is to mask a fixed proportion of each component in the trajectory and guide the GUI Agent to predict the missing parts. Specifically, this strategy enables the GUI Agent to learn effectively even in the absence of partial contextual information, thereby enhancing its robustness and ability to generalize across diverse scenarios.

Formally, given a GUI Agent θ and a representative GUI trajectory $T = (\text{Inst}, \{(m_t, r_t, a_t)\}_{t=1}^{T})$,where m_t, r_t, and a_t denote the screenshot, the reasoning process, and the action at step t respectively, we randomly mask a fixed proportion of components within the trajectory. As exemplified in Figure 3(a), MaP replaces r_1 (CoT reasoning of the 1st step), r_3 (CoT reasoning of the 3rd step), a_3 (action of the 3rd step) in the trajectory T with the special [`Mask`] token, resulting in a masked version denoted as X'. Crucially, these [`Mask`] tokens act as explicit placeholders that prompt the agent to predict the missing components, computing the

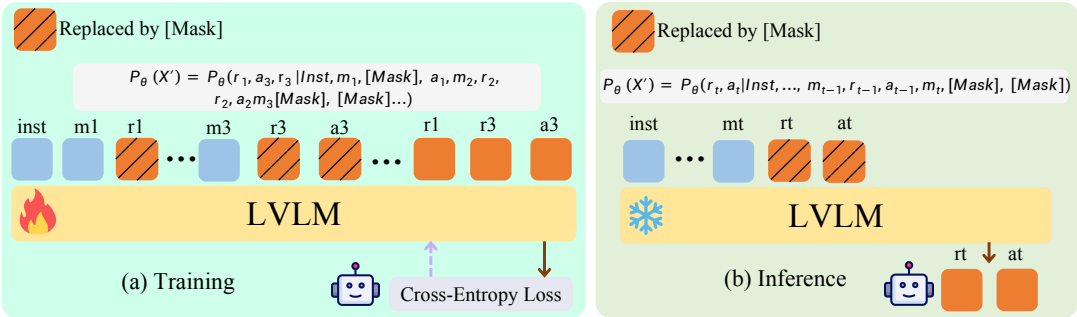


Figure 3: Overview of the MaP paradigm during training and inference.

loss exclusively on the masked positions. The training objective is to predict the specific masked components (i.e., r_1, a_3, and r_3) based on the masked trajectory X', which can be formulated as the following conditional probability:

$$P_\theta(X') = P_\theta\big(r_1, a_3, r_3 \mid \text{Inst}, m_1, [\texttt{Mask}], a_1, m_2, r_2, a_2, m_3, [\texttt{Mask}], [\texttt{Mask}], \ldots\big). \quad (1)$$

Fundamentally, this trajectory-level masking objective shifts the optimization paradigm from fitting isolated, task-specific marginal distributions to capturing the joint distribution of the entire GUI interaction process. By doing so, it inherently eliminates the objective inconsistencies among step-wise decision, state-action alignment, and long-horizon planning tasks to learn a globally coherent navigation logic within a unified sequence modeling framework.

Inspired by the success of masked autoencoders in the vision domain (He et al., 2022), we adopt a relatively high masked ratio to increase task difficulty and encourage the GUI Agent to reason over long-horizon dependencies within GUI trajectories. Empirically, we find that masking 80% of the components within a trajectory achieves the best trade-off between task difficulty and model performance. Considering the inherent heterogeneity of data sourced from multiple task paradigms and the presence of low-quality screenshots in existing open-source datasets, we avoid extreme masking ratios, such as 100%, which could adversely impact learning stability. An 80% masking ratio compels the GUI Agent to predict the masked components from the masked trajectory, while still ensuring sufficient data utilization and effective supervision. A detailed investigation into the effects of varying masking ratios is provided in the Section 4.4.

In the downstream inference stage, MaP simulates the same masking configuration as used during training by replacing the current-step CoT and action with the special [Mask] token. As illustrated in Figure 3(b), the GUI Agent predicts the corresponding reasoning and action based on the contextual trajectory, and the inference objective can be expressed as follows:

$$P_\theta(X') = P_\theta\big(a_t, r_t \mid \text{inst}, \ldots, m_{t-1}, r_{t-1}, a_{t-1}, m_t, [\texttt{Mask}], [\texttt{Mask}]\big). \quad (2)$$

where a_t and r_t denote the predicted action and the corresponding reasoning process at step t, respectively.

3.3 Role-aware Adapter Learning

While the previous section established how MaP provides a consistent training objective across diverse navigation tasks, the substantial data heterogeneity within existing GUI corpora remains a significant barrier to unified modeling. Specifically, data sourced from multiple navigation tasks inherently contains distinct reasoning processes and annotations. More importantly, this heterogeneity is inherent to the data itself, as diverse navigation tasks yield trajectories with fundamentally different reasoning distributions and semantic complexities. As shown in Figure 4, a representative GUI trajectory intertwines up to six distinct conceptual roles: high- and low-level instructions detailing abstract user goals and step-wise atomic commands, observations reflecting the current state of the interface, thoughts capturing the step-specific reasoning process, and executable GUI actions. Each of these components exhibits unique semantic densities and thus requires a specialized representation space. Furthermore, the quality of images varies significantly, with human-collected screenshots being generally high-quality, while those extracted from web tutorials often contain instructional visual elements such as red circles, arrows, or overlays (Zhang et al., 2025a).

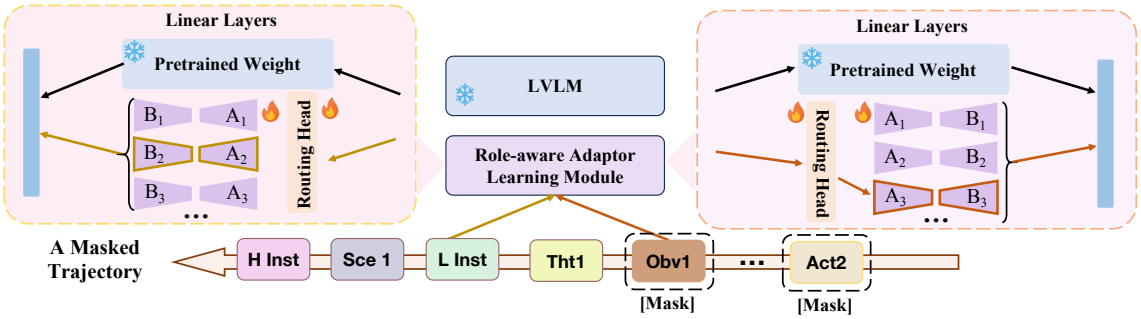


Figure 4: Illustration of the role-aware adapter learning module. Given a masked trajectory with heterogeneous components, the module dynamically assigns each token to a specific adapter for targeted optimization. Abbreviation: H Inst=High level instruction; L Inst=Low level instruction; Scne=Screenshot; Tht=Thought; Obv=Observation; Act=Action.

Accordingly, we introduce a role-aware adapter learning module to address the challenge of data heterogeneity. Since prior mid-training methods predominantly adopt Low-Rank Adaptation (LoRA) training (Hu et al., 2022), we extend LoRA by introducing multiple specialized adapters and a token-wise router that dynamically selects one for each token based on its role in the GUI trajectory.

To begin with, we briefly review the core concepts of LoRA. It assumes that parameter updates lie in a low-dimensional subspace, allowing training to be performed through a low-rank decomposition while keeping the pretrained weights frozen. Based on this formulation, the forward pass of a LoRA layer can be expressed as follows:

$$\Delta W_0 = BA, \quad h = W_0 x + \alpha \cdot \Delta W_0 x \tag{3}$$

where $x \in R^k$ is the input feature, $W_0 \in R^{d\times k}$ denotes the frozen pretrained weight, and ΔW_0 is the trainable update parameterized by a low-rank decomposition, with $B \in R^{d\times r}$ and $A \in R^{r\times k}$, such that $r \ll \min(d, k)$. The scalar α controls the contribution of the update during training.

As illustrated in Figure 4, this module extends standard LoRA by introducing multiple adapters for each component in the GUI trajectory, aiming to address the data heterogeneity inherent in existing training corpora. Specifically, to dynamically assign different tokens to appropriate adapters, a token-wise routing mechanism is employed. It selects the most suitable adapter for each token based on a linear scoring function:

$$\begin{cases} \hat{i} = \arg\max_i \left(w_i^\top x\right), \\ \Delta W_0 = B_{\hat{i}} A_{\hat{i}}, \\ h = W_0 x + \alpha \cdot \Delta W_0 x. \end{cases} \tag{4}$$

where x is the input feature, w_i is the learnable routing weight for the i-th adapter, and B_i, A_i are the low-rank matrices associated with adapter i. The selected adapter $\hat{i}$ is used to generate the low-rank update ΔW_0, which is then applied during forward propagation.

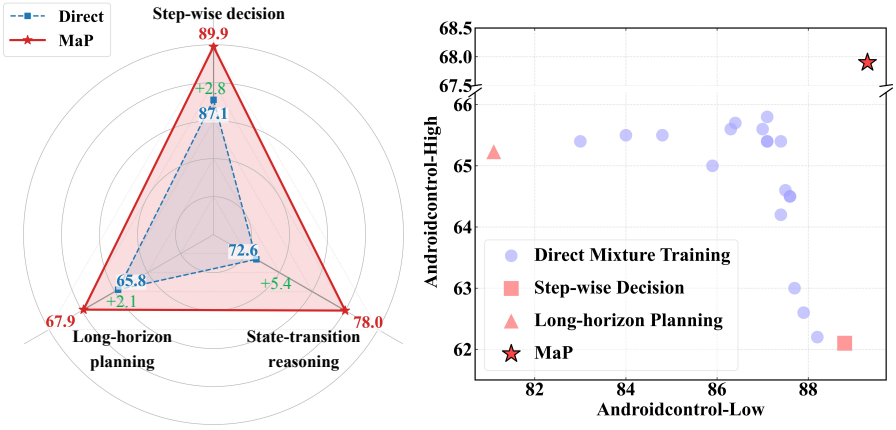
4 Experiments

4.1 Datasets and Evaluation Metrics

For benchmark evaluation, MaP is evaluated on five navigation datasets: AndroidWorld (Rawles et al., 2024), AndroidControl (Li et al., 2024), GUI-Odyssey (Lu et al., 2024), AITZ (Zhang et al., 2024), and Mind2Web (Deng et al., 2023). We categorize them into two groups: global evaluation benchmarks (AndroidWorld, AndroidControl-High, GUI-Odyssey, Mind2Web) that assess end-to-end task execution with high-level instructions, and local evaluation benchmarks (AndroidControl-Low) that focus on step-wise action execution with low-level instructions. Detailed dataset statistics are provided in the Appendix.

For evaluation metrics, we adopt three standard metrics (Wu et al., 2024b): Type (exact match of predicted action type), Ground (coordinate accuracy for click actions), and Step-wise Success Rate (SR, whether both action type and arguments are completely correct).

For training data, we integrate major public datasets from diverse mid-training paradigms (Zhang et al., 2025a; Xu et al., 2024; Wu et al., 2024a; Jang et al., 2025; Sun et al., 2024, 2025) to establish a unified action space, with trajectories averaging 5.6 steps in length. Detailed descriptions of the training data and unified action types are provided in Appendix. For the direct mixture training comparison, we utilize an equivalent volume of data spanning three navigation task configurations (Xu et al., 2024; Wu et al., 2024b; Shen et al.,



(a) Navigation Capabilites Evaluation (b) Trade-off Analysis

Figure 5: **Left:** Comparison of three core navigation capabilities between direct mixture training and MaP. **Right:** Trade-off analysis between direct mixture training and MaP.

2024; Qin et al., 2025): step-wise decision pairs the current screenshot with a low-level instruction (Xu et al., 2024; Wu et al., 2024b); state-action alignment provides two consecutive screenshots to predict the intermediate action (Shen et al., 2024); and long-horizon planning formulates GUI tasks as dialogue-like interactions for step-wise plan decomposition (Xu et al., 2024).

4.2 Superiority of MaP in Mixture Training

To investigate effective unification strategies to improve GUI navigation capabilities, we compare direct mixture training with MaP, assessing zero-shot performance across three navigation abilities and the impact on post-training downstream datasets. We further analyze the performance trade-offs inherent in direct mixture training to reveal the underlying conflicts that distinguish it from MaP.

4.2.1 Zero Shot Performance.

As illustrated in Figure 5(a), we evaluate direct mixture training and MaP under a zero-shot setting on AndroidControl (Li et al., 2024) across three navigation capabilities, serving to assess intrinsic generalization ability without dataset-specific adaptation. Our experiments show that MaP consistently outperforms direct mixture training across all three settings, yielding gains of 2.8%, 5.4%, and 2.1% on step-wise decision, state-action alignment, and long-horizon planning, respectively. This improvement is attributable to a consistent objective that aligns training signals across all navigation capabilities. By forcing the GUI Agent to reconstruct masked components in the GUI trajectory, MaP inherently learns robust, task-agnostic representations that generalize zero-shot to diverse navigation requirements.

4.2.2 Post Training Performance.

As illustrated in Section 4.2.1 and Section 4.2.1, we employ Qwen2.5-VL (Bai et al., 2025) with various scales to assess the impact of MaP on post-training efficacy across multiple benchmarks. Empirical results reveal a clear contrast: direct mixture training yields negligible improvements over the baseline, whereas MaP delivers substantial SR increases of 3.0% and 2.7% on AndroidControl-High and AndroidControl-Low, respectively. These improvements highlight that MaP serves as a superior initialization for downstream GUI tasks, irrespective of the underlying model scale (e.g., 3B or 7B).

4.2.3 Trade-off Analysis.

As illustrated in Figure 5(b), we analyze the performance trade-off inherent in direct mixture training, where the x-axis and y-axis represent SR on AndroidControl-Low and AndroidControl-High, respectively. Evaluating intermediate checkpoints from direct mixture training reveals a clear suboptimal trade-off, where it fails to surpass the specialized baselines on both benchmarks, indicating that optimizing for one capability often comes at the expense of the other. This highlights a typical negative transfer problem in multi-task learning, where mixing vastly different data formats leads to conflicting training signals. In contrast, MaP breaks this performance ceiling by unifying the optimization objectives across all three navigation tasks, achieving superior results of 89.9% and 67.9% on AndroidControl-Low and AndroidControl-High, respectively.

4.3 Main Result

As illustrated in Section 4.2.1 and Section 4.2.1, we present a comprehensive comparison with ex-

Methods	Param.	AndroidControl-High			AndroidControl-Low		
		Type	Ground	SR	Type	Ground	SR
Claude (Anthropic, 2024)	–	63.7	0.0	12.5	74.3	0.0	19.4
GPT-4o (OpenAI, 2024)	–	66.3	0.0	20.8	74.3	0.0	19.4
SeeClick (Cheng et al., 2024)	9.6B	82.9	62.9	59.1	93.0	73.4	75.0
CPM-GUI (Zhang et al., 2025b)	7B	77.7	–	69.2	94.4	–	90.2
OS-Genesis (Sun et al., 2024)	7B	66.2	–	44.5	74.2	–	90.7
OS-Atlas (Wu et al., 2024b)	7B	85.2	78.5	71.2	93.6	88.0	85.2
AGUVIS (Xu et al., 2024)	7B	–	–	61.5	–	–	80.5
UI-TARS (Qin et al., 2025)	7B	83.7	**80.5**	72.5	98.0	89.3	90.8
Falcon-UI (Shen et al., 2024)	7B	–	–	72.7	–	–	86.6
Qwen2.5-VL (Bai et al., 2025)	3B	84.9	75.4	68.9	96.5	87.8	87.0
+*Direct*	3B	$85.4_{+0.5}$	$75.9_{+0.5}$	$70.1_{+1.2}$	$96.8_{+0.3}$	$88.2_{+0.4}$	$87.9_{+0.9}$
+*MaP*	3B	$86.0_{+1.1}$	$76.8_{+1.4}$	$71.9_{+3.0}$	$97.1_{+0.6}$	$89.0_{+1.2}$	$89.7_{+2.7}$
Qwen2.5-VL (Bai et al., 2025)	7B	86.4	78.3	71.2	96.9	89.1	88.2
+*Direct*	7B	$86.7_{+0.1}$	$78.9_{+0.6}$	$72.9_{+1.7}$	$97.0_{+0.3}$	$89.2_{+0.1}$	$89.0_{+0.8}$
+*MaP*	7B	$\mathbf{87.2}_{+0.8}$	$79.7_{+1.4}$	$\mathbf{74.2}_{+3.0}$	$97.9_{+1.0}$	$\mathbf{90.1}_{+1.0}$	$\mathbf{90.9}_{+2.7}$

Table 1: Results on AndroidControl (Li et al., 2024). The best is **bold**, second is underlined. + numbers indicate improvement over Qwen2.5-VL.

Methods	Param.	GUI-Odyssey			AITZ	Mind2web
		Type	Ground	SR		
Claude (Anthropic, 2024)	–	60.9	0.0	3.1	–	–
GPT-4o (OpenAI, 2024)	–	34.3	0.0	3.3	–	56.6
SeeClick (Cheng et al., 2024)	9.6B	71.0	52.4	53.9	–	20.9
CPM-GUI (Zhang et al., 2025b)	7B	90.9	–	75.0	–	–
OS-Atlas (Wu et al., 2024b)	7B	84.5	67.8	62.0	–	–
UI-TARS (Qin et al., 2025)	7B	94.6	**90.1**	87.0	–	–
Falcon-UI (Shen et al., 2024)	7B	–	–	–	69.1	27.6
Qwen2.5-VL (Bai et al., 2025)	3B	95.1	85.4	83.1	71.3	54.5
+*Direct*	3B	$95.4_{+0.3}$	$86.3_{+0.9}$	$83.9_{+0.8}$	$71.9_{+0.6}$	$55.1_{+0.6}$
+*MaP*	3B	$96.0_{+0.9}$	$87.4_{+2.0}$	$85.0_{+1.9}$	$73.2_{+1.9}$	$56.9_{+2.4}$
Qwen2.5-VL (Bai et al., 2025)	7B	95.9	86.9	85.1	71.5	57.1
+*Direct*	7B	$97.1_{+1.2}$	$87.8_{+0.9}$	$86.2_{+1.1}$	$72.4_{+0.9}$	$57.4_{+0.3}$
+*MaP*	7B	$\mathbf{97.8}_{+1.9}$	$89.9_{+3.0}$	$\mathbf{88.5}_{+3.4}$	$\mathbf{73.9}_{+2.4}$	$\mathbf{59.3}_{+2.2}$

Table 2: Results on GUI-Odyssey (Lu et al., 2024), AITZ (Zhang et al., 2024), and Mind2Web (Deng et al., 2023). The best is **bold**, second is underlined. + numbers indicate improvement over Qwen2.5-VL.

isting GUI Agent mid-training methods, analyzing experimental results ranging from global to local action execution.

4.3.1 Online Evaluation.

As shown in Figure 6, with Qwen2.5-VL-7B as the backbone, we compare direct mixture training and MaP on the AndroidWorld (Rawles et al., 2024) in terms of Pass@1 and Pass@4. The online evaluation encompasses interactions with diverse and complex real-world applications, involving practical GUI tasks designed to enhance productivity and facilitate information retrieval. Specifically, MaP achieves consistent performance gains of 2.6% and 5.4% in Pass@1 and Pass@4 over direct mixture training on the AndroidWorld, where the larger margin in Pass@4 suggests that MaP unlocks greater GUI Agent's potential.

4.3.2 Offline Evaluation

Compared with existing GUI Agent mid-training methods (Qin et al., 2025; Shen et al., 2024; Wu et al., 2024b; Cheng et al., 2024; Xu et al., 2024), MaP demonstrates superior performance on the AndroidControl-High, GUI-Odyssey, AITZ, and Mind2Web datasets. Among these, AndroidControl-High and GUI-Odyssey dataset concentrate on the most prevalent mobile platforms for GUI agent evaluation. AndroidControl-High dataset (Li et al., 2024) covers various scenarios across 833 Android applications, while GUI-Odyssey (Lu et al., 2024) presents a long-horizon navigation challenge, with trajectories averaging 15.4 steps in length. As shown in Section 4.2.1 and Section 4.2.1, MaP on Qwen2.5-VL-7B achieves a higher SR than UI-TARS-7B (Qin et al., 2025),

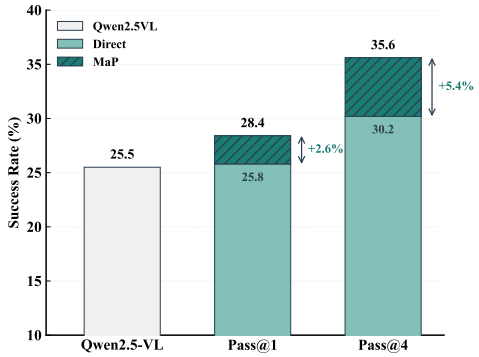


Figure 6: The performance of Qwen2.5VL-7B with direct mixture training and MaP on AndroidWorld.

Mask Ratio	Type	Ground	SR
0.2	86.3	78.4	72.5
0.5	86.9	79.3	73.7
0.8	**87.2**	**79.7**	**74.2**
1.0	86.7	79.1	73.8

Table 3: Ablation on mask ratio.

Num. Adapters	Type	Ground	SR
1	86.7	79.0	73.4
2	86.9	79.4	73.8
4	**87.2**	**79.7**	**74.2**
8	87.1	79.6	74.1

Table 4: Ablation on the numbers of adapter.

with improvements of 1.7% on AndroidControl-High and 1.5% on GUI-Odyssey, respectively. We attribute the performance gain to MaP's capacity to unify diverse navigation tasks into consistent training objectives while effectively addressing the heterogeneity of GUI corpora.

Regarding evaluations beyond mobile platforms, MaP on Qwen2.5-VL-7B surpasses Falcon-UI (Shen et al., 2024) by 4.8% on the AITZ dataset (Zhang et al., 2024), covering diverse scenarios in Section 4.2.1. On the Mind2Web dataset (Deng et al., 2023), MaP on Qwen2.5-VL-7B achieves state-of-the-art performance, surpassing UI-TARS even though the latter leverages extensive pretraining on 50B tokens of in-house data, as shown in Section 4.2.1. This superior performance stems from MaP's unified framework and role-aware adapter learning module, which jointly support consistent training across heterogeneous data from all prior navigation strategies.

4.3.3 Local Evaluation.

For local evaluation, the GUI Agent is provided with a current screenshot and a pair of high- and low-level instructions. As shown in Section 4.2.1, MaP (Qwen2.5-VL-7B) achieves higher SR than UI-TARS (Qin et al., 2025) under AndroidControl-Low, despite UI-TARS being pretrained on 50B tokens. We attribute this to training on diverse prior navigation strategies, particularly single-step instructions, which improve generalization to local planning tasks.

4.4 Ablation Studies

4.4.1 Mask Ratio.

Since the mask ratio is key to MaP, we investigate how different masking ratios (0.2, 0.5, 0.8, and 1.0) influence the model's effectiveness and conduct ablation studies on Qwen2.5-VL-7B (MaP). As shown in Section 4.3.2, performance consistently improves as the mask ratio increases from 0.2 to 0.8, since higher mask ratios allow more data to contribute to gradients, while a significant drop is observed at 1.0. This is because the model is forced to rely solely on visual context from multiple screenshots to reason, greatly increasing training difficulty. Moreover, the presence of low-quality samples such as web-based GUI trajectories containing noisy visual cues like red circles (Zhang et al., 2025a) further contributes to the degradation.

4.4.2 Number of Adapters.

To address the heterogeneity across diverse open-source navigation GUI datasets, we propose a role-aware adapter learning module that dynamically assigns different tokens of a trajectory to specialized adapters, allowing targeted optimization with heterogeneous data. We investigate the impact of using 1, 2, 4, or 8 adapters on Qwen2.5-VL-7B (MaP) with AndroidControl-High dataset. As shown in Section 4.3.2, performance improves as the number of adapters increases, peaking at four, which outperforms the standard single-adapter LoRA baseline ($Num.Adapters = 1$) (Hu et al., 2022).

5 Conclusion

In this paper, we address inconsistent objectives and data heterogeneity in multi-task GUI agent mixture training. We propose MaP, a unified framework that formulates diverse navigation strategies as masked trajectory prediction. With trajectory masking and a role-aware adapter, MaP consolidates conflicting objectives into a consistent learning goal on heterogeneous data. Extensive evaluations show MaP significantly outperforms direct mixture training, establishing a new standard for versatile GUI navigation.

Limitations

Although MaP achieves consistent improvements across multiple GUI navigation benchmarks, our current study is mainly conducted with moderate-scale backbone models and publicly available GUI corpora. Due to computational constraints, we have not fully explored the scaling potential of MaP with larger LVLMs or more diverse training data. Future work will investigate whether MaP can further benefit from larger model capacities and expanded GUI trajectory corpora.

Ethical Considerations

This work proposes a training framework for GUI agents and does not introduce additional ethical concerns beyond those of existing GUI agent systems. Potential risks mainly come from GUI interaction itself, such as sensitive information in screenshots or unintended operations in practical deployment. These issues should be handled with standard safeguards, including data anonymization, user authorization, and confirmation for sensitive actions.

References

Anthropic. 2024. Developing a computer use model. Anthropic blog.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Dongping Chen, Yue Huang, Siyuan Wu, Jingyu Tang, Liuyi Chen, Yilin Bai, Zhigang He, Chenlong Wang, Huichi Zhou, Yiqiang Li, and 1 others. 2024. Gui-world: A dataset for gui-oriented multimodal llm-based agents. *arXiv e-prints*, pages arXiv–2406.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*.

Laure Ciernik, Lorenz Linhardt, Marco Morik, Jonas Dippel, Simon Kornblith, and Lukas Muttenthaler. 2024. Objective drives the consistency of representational similarity across datasets.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and 1 others. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Yunseok Jang, Yeda Song, Sungryull Sohn, Lajanugen Logeswaran, Tiange Luo, Dong-Ki Kim, Kyunghoon Bae, and Honglak Lee. 2025. Scalable video-to-dataset generation for cross-platform mobile agents. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8604–8614.

Kim et al. 2023. Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36:39648–39677.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. 2024. On the effects of data scale on computer control agents. *arXiv e-prints*, pages arXiv–2406.

Yuhang Liu, Zeyu Liu, Shuanghe Zhu, Pengxiang Li, Congkai Xie, Jiasheng Wang, Xavier Hu, Xiaotian Han, Jianbo Yuan, Xinyao Wang, and 1 others. 2025. Infigui-g1: Advancing gui grounding with adaptive exploration policy optimization. *arXiv preprint arXiv:2508.05731*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Dunjie Lu, Yiheng Xu, Junli Wang, Haoyuan Wu, Xinyuan Wang, Zekun Wang, Junlin Yang, Hongjin Su, Jixuan Chen, Junda Chen, and 1 others. 2025. Videoagenttrek: Computer use pretraining from unlabeled videos. *arXiv preprint arXiv:2510.19488*.

Quanfeng Lu, Wenqi Shao, Zitao Liu, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. Gui odyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. *arXiv preprint arXiv:2406.08451*.

Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. 2025. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*.

Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, and 1 others. 2025. Gui agents:

A survey. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22522–22538.

OpenAI. 2024. Gpt-4o system card. OpenAI system card via arXiv.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, and 1 others. 2025. Ui-tars: Pioneering automated GUI interaction with native agents. *arXiv preprint arXiv:2501.12326*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, and 1 others. 2024. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2023. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36:59708–59728.

Huawen Shen, Chang Liu, Gengluo Li, Xinlong Wang, Yu Zhou, Can Ma, and Xiangyang Ji. 2024. Falcon-ui: Understanding gui before following user instructions. *arXiv preprint arXiv:2412.09362*.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*, pages 3135–3144. PMLR.

Qiushi Sun, Kanzhi Cheng, Zichen Ding, Chuanyang Jin, Yian Wang, Fangzhi Xu, Zhenyu Wu, Chengyou Jia, Liheng Chen, Zhoumianze Liu, and 1 others. 2024. Os-genesis: Automating gui agent trajectory construction via reverse task synthesis. *arXiv preprint arXiv:2412.19723*.

Yuchen Sun, Shanhui Zhao, Tao Yu, Hao Wen, Samith Va, Mengwei Xu, Yuanchun Li, and Chongyang Zhang. 2025. Gui-xplore: Empowering generalizable gui agents with one exploration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19477–19486.

Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Junting Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, and 1 others. 2025. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Shuai Wang, Weiwen Liu, Jingxuan Chen, Yuqi Zhou, Weinan Gan, Xingshan Zeng, Yuhan Che, Shuai Yu, Xinlong Hao, Kun Shao, and 1 others. 2024b. Gui agents with foundation models: A comprehensive survey. *arXiv preprint arXiv:2411.04890*.

Qinzhuo Wu, Weikai Xu, Wei Liu, Tao Tan, Jianfeng Liu, Ang Li, Jian Luan, Bin Wang, and Shuo Shang. 2024a. Mobilevlm: A vision-language model for better intra-and inter-ui understanding. *arXiv preprint arXiv:2409.14818*.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and 1 others. 2024b. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2024. Aguvis: Unified pure vision agents for autonomous gui interaction. *arXiv preprint arXiv:2412.04454*.

Bofei Zhang, Zirui Shang, Zhi Gao, Wang Zhang, Rui Xie, Xiaojian Ma, Tao Yuan, Xinxiao Wu, Song-Chun Zhu, and Qing Li. 2025a. Tongui: Building generalized gui agents by learning from multimodal web tutorials. *arXiv preprint arXiv:2504.12679*.

Jiwen Zhang, Jihao Wu, Yihua Teng, Minghui Liao, Nuo Xu, Xiao Xiao, Zhongyu Wei, and Duyu Tang. 2024. Android in the zoo: Chain-of-action-thought for gui agents. *arXiv preprint arXiv:2403.02713*.

Zhong Zhang, Yaxi Lu, Yikun Fu, Yupeng Huo, Shenzhi Yang, Yesai Wu, Han Si, Xin Cong, Haotian Chen, Yankai Lin, and 1 others. 2025b. Agentcpm-gui: Building mobile-use agents with reinforcement fine-tuning. *arXiv preprint arXiv:2506.01391*.

Appendix

The appendix includes the following aspects:

- A: Unified Action Space.
- B: Implementation Details.
- C: Evaluation Benchmarks.
- D: Theoretical Analysis.
- E: Gradient Optimization Analysis.
- F: Qualitative Analysis.

A Unified Action Space

In this section, we introduce the unified action space of our proposed MaP framework for GUI navigation. As shown in appendix A, we categorize actions into three distinct types: Ground, Content, and Func. Specifically, Ground actions require focusing on specific element targets within the screenshot, necessitating the prediction of both an action type and precise coordinates. Conversely, Content and Func actions demand a holistic understanding of the entire screenshot. Content actions involve generating an action type paired with specific textual content, whereas Func actions require only the action type. To formulate this efficiently, we adopt the native action set of the Qwen2.5-VL base model (Bai et al., 2025). This design guarantees the agent's universality across diverse operating environments while preserving the fine-grained flexibility necessary for complex interactions.

B Implementation Details

For MaP, which is built upon Qwen2.5-VL series (Bai et al., 2025), all input images are resized to 1280×720 to achieve a better trade-off between performance and efficiency. The maximum token sequence length is set to 16384 for each LVLM. During training, each LVLM in MaP is trained with a batch size of 32 and 8 gradient accumulation steps. We use the AdamW optimizer (Loshchilov and Hutter, 2017) for training, along with a cosine learning rate scheduler and a warm-up phase comprising 5% of the total training steps. To reduce GPU memory consumption, we adopt DeepSpeed optimization (Rasley et al., 2020), BF16 precision, and gradient checkpointing. All experiments are conducted on a cluster of H100-80G GPUs.

C Evaluation Benchmarks

In this section, we introduce more details of the evaluation benchmarks used in our work.

C.0.1 AndroidWorld.

AndroidWorld is a scalable environment for mobile GUI Agents evaluation. This benchmark encompasses 116 tasks across 20 authentic Android apps, ranging from system configurations to complex user workflows like online shopping and information retrieval. These tasks involve multi-step interactions that require the agent to locate specific elements and execute correct actions by following user instructions. For evaluation, we utilize the environment's state-based reward mechanism to deterministically verify task completion and report the Success Rate (SR).

C.0.2 AndroidControl.

AndroidControl comprises 15,226 tasks across 120 diverse applications, simulating a broad range of real-world mobile interactions. Following (Wu et al., 2024b), we adopt Grounding to quantify the accuracy of grounding actions and SR to measure the exact match of the predicted action step.

C.0.3 GUI-Odyssey.

GUI-Odyssey is a large-scale, long-horizon dataset specifically established for evaluating cross-app GUI navigation on mobile devices. It comprises 8,334 episodes spanning 212 diverse applications, capturing complex workflows that require the GUI Agent to manage long-term history and navigate interactions across multiple apps. Following (Wu et al., 2024b), we adopt Ground Acc to quantify the accuracy of grounding actions and SR to measure the exact match of the predicted action step.

C.0.4 AITZ.

AITZ (Android-In-The-Zoo) dataset is specifically constructed to evaluate and train autonomous GUI agents on smartphones. Comprising 18,643 screen-action pairs, AITZ is distinguished by its comprehensive Chain-of-Action-Thought (CoAT) annotations.

C.0.5 Mind2Web.

Mind2Web dataset is established to evaluate and train generalist GUI agents across diverse, real-world web environments. It comprises over 2,000 open-ended tasks spanning 137 dynamic websites across 31 distinct domains. A defining characteristic of Mind2Web is its rigorous focus on zero-shot

Category	Action Format	Action Description
Ground	`click(point)`	Tap on the specified position.
	`long_press(point)`	Long-press on the specified position.
	`scroll(point, point1)`	Swipe on the screen.
Content	`open(app_name)`	Open the specified application.
	`type(text)`	Enter the specified text.
	`answer(text)`	Answer the specified user's question.
Func	`wait()`	Temporarily pause the execution.
	`system_button('Home')`	Navigate to the home screen.
	`system_button('Back')`	Return to the previous screen.
	`system_button('Enter')`	Confirm an input to the next step.
	`terminate('success')`	No further actions required.
	`terminate('failure')`	Requires additional steps.

Table 5: Overview of the action space definition.

generalization, comprehensively testing an agent's capability to navigate entirely unseen tasks, websites, and domains.

D Theoretical Analysis

In this section, we analyze the consistency of training objectives in terms of optimization direction between direct unification mixture optimization and MaP from a gradient perspective. Specifically, we define the direct mixture optimization of the three navigation paradigms, where the gradient for parameter updates is represented by $\nabla_\theta L_{\text{mix}}$. Here, $\theta_{\text{SD}}, \theta_{\text{SA}}, \theta_{\text{LP}}$ denote the parameters for step-wise decision, state-action alignment, and long-horizon planning, respectively. The total gradient is formulated as:

$$\nabla_\theta L_{\text{mix}} = \nabla_{\theta_{\text{SD}}} L_{\text{SD}} + \nabla_{\theta_{\text{SA}}} L_{\text{SA}} + \nabla_{\theta_{\text{LP}}} L_{\text{LP}}, \quad (5)$$

where $L_{\text{SD}}, L_{\text{SA}}, L_{\text{LP}}$ are the loss functions of the three tasks. By summing the gradients from these tasks, the mixture optimization aims to balance the optimization directions of all paradigms.

The core idea of MaP is to treat any GUI trajectory T as a whole and predict the masked components by randomly masking parts of the trajectory (such as r_t, a_t, or m_t). This approach unifies the dependency modeling of all navigation tasks, as it allows each component of the trajectory to be predicted in the context of the entire sequence. From a gradient perspective, this means that the optimization process does not treat the tasks in isolation but instead optimizes the entire trajectory as a whole. By masking different components, MaP enforces a consistent training objective across tasks, ensuring that the gradients for different components of the trajectory align toward a common goal.

As mentioned in the main manuscript, we analyze the mean and variance of the gradient cosine similarity between MaP and the direct mixture of existing navigation paradigms, with the detailed formulas provided in the appendix. The formula for the cosine similarity between two gradient vectors ∇A and ∇B is given by:

$$\cos(\nabla A, \nabla B) = \frac{\nabla A \cdot \nabla B}{\|\nabla A\| \, \|\nabla B\|} \quad (6)$$

where ∇A and ∇B can be any samples from the existing navigation paradigms. The mean of the gradient cosine similarity is defined as:

$$\mu_{\cos} = E[\cos(\nabla A, \nabla B)] = \frac{1}{N} \sum_{i=1}^{N} \cos(\nabla A_i, \nabla B_i). \quad (7)$$

The variance of the gradient cosine similarity is given by:

$$\text{Varcos} = E\left[(\cos(\nabla A, \nabla B) - \mu\cos)^2\right]. \quad (8)$$

E Gradient Optimization Analysis.

To further investigate why MaP shows advantages in unification, we compare the gradient cosine similarity between MaP and direct mixture training. As shown in Figure 7(b), kernel density estimation plots at the LM head and the last attention k-projection layer reveal that MaP yields a mean

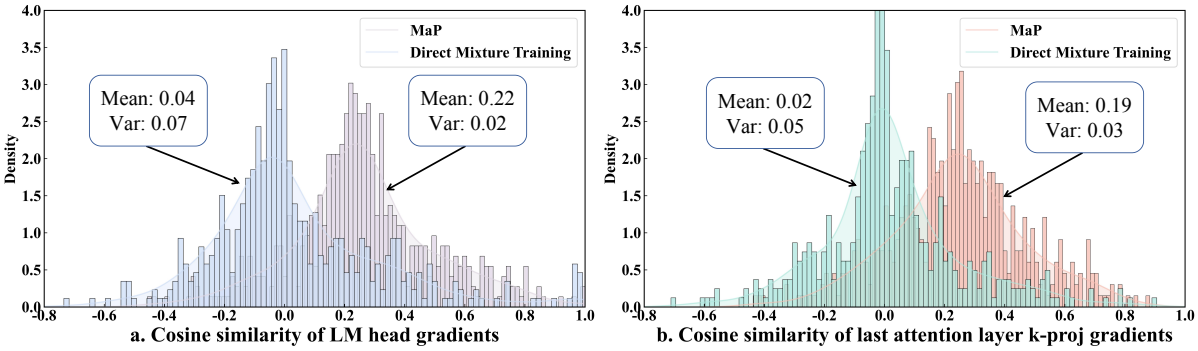


Figure 7: Kernel density estimation plots illustrating the gradient cosine similarity between individual training samples.

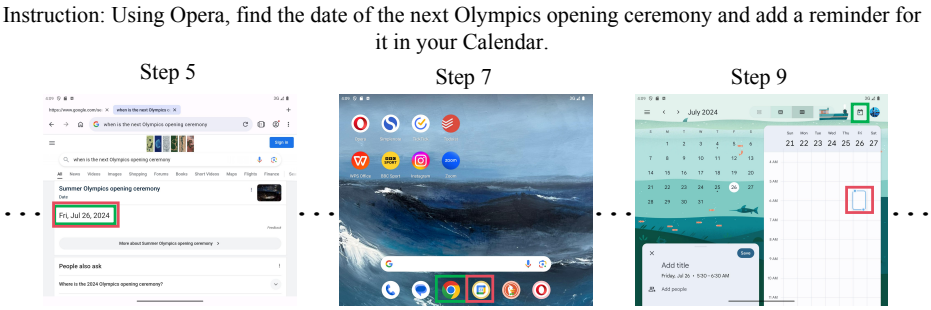


Figure 8: Qualitative results on the GUI-Odyssey dataset (Lu et al., 2024). High-level instructions are visualized at the top of each image. Predicted tap positions from our MaP (Qwen2.5-VL-3B) are shown in red, while those from the base model (Qwen2.5-VL-3B) are shown in green. Best viewed with zoom.

cosine similarity approximately 0.17 higher and a variance about 0.02 lower than direct mixture training. This enhanced cosine similarity reflects more harmonized training objectives, while the reduced variance suggests superior training stability (Ciernik et al., 2024), both of which highlight MaP's efficacy in joint optimization. Further theoretical analysis is provided in the Appendix.

F Qualitative Analysis

We present a qualitative comparison on the GUI-Odyssey dataset (Lu et al., 2024) between the base model (Qwen2.5-VL-3B) and MaP (Bai et al., 2025). As shown in Figure 8, the task involves adding the date of the Olympics opening ceremony to the calendar. The base model exhibits two major navigation failures, including opening Google Chrome instead of the calendar application in the step 7th and selecting an incorrect date in the 9th. In contrast, MaP successfully executes the intended sequence of actions, demonstrating that MaP significantly enhances long-horizon planning performance.

Moving from global to local evaluation, we present a qualitative comparison on the AndroidControl-Low dataset (Li et al., 2024) between the base model (Qwen2.5-VL-3B) and MaP (Qwen2.5-VL-3B), highlighting their differences in step-wise decison capabilities through four representative examples. For instance, as shown in Figure 9 (a), MaP demonstrates stronger generalization during training. While the base model tends to tap on "multiple options" when it fails to recognize the "send" icon, MaP correctly identifies and taps the intended target.

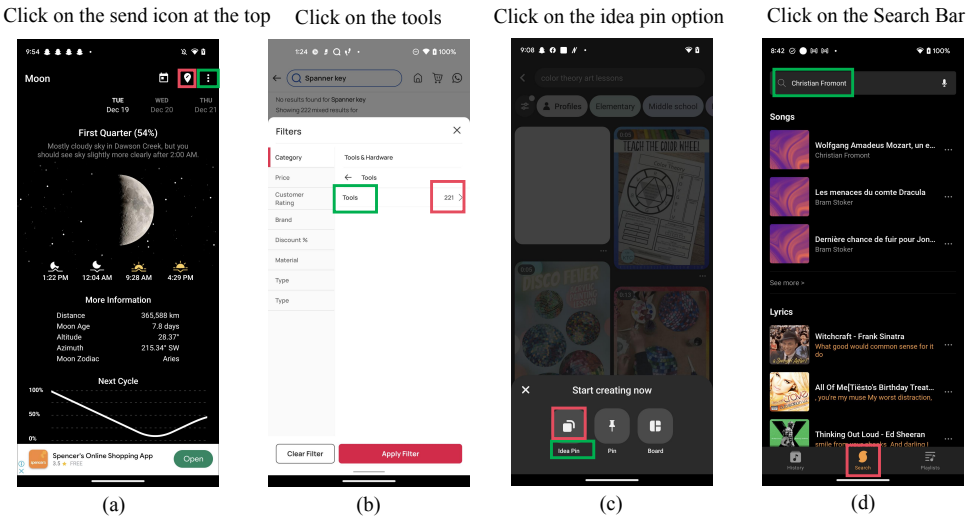


Figure 9: Qualitative results on the AndroidControl-Low dataset (Li et al., 2024). Low-level instructions are visualized at the top of each image. Predicted tap positions from our MaP (Qwen2.5-VL-3B) are shown in red, while those from the base model (Qwen2.5-VL-3B) are shown in green. Best viewed with zoom.