Retrieval-augmented GUI Agents with Generative Guidelines

Ran Xu[1], **Kaixin Ma**[2], **Wenhao Yu**[2], **Hongming Zhang**[2],
Joyce C. Ho[1], **Carl Yang**[1], **Dong Yu**[2]
[1] Emory University [2] Tencent AI Lab
ran.xu@emory.edu

Abstract

GUI agents powered by vision-language models (VLMs) show promise in automating complex digital tasks. However, their effectiveness in real-world applications is often limited by scarce training data and the inherent complexity of these tasks, which frequently require long-tailed knowledge covering rare, unseen scenarios. We propose RAG-GUI , a lightweight VLM that leverages web tutorials at inference time. RAG-GUI is first warm-started via supervised finetuning (SFT) and further refined through self-guided rejection sampling finetuning (RSF). Designed to be model-agnostic, RAG-GUI functions as a generic plug-in that enhances any VLM-based agent. Evaluated across three distinct tasks, it consistently outperforms baseline agents and surpasses other inference baselines by 2.6% to 13.3% across two model sizes, demonstrating strong generalization and practical plug-and-play capabilities in real-world scenarios.

1 Introduction

Graphical User Interface (GUI) agents have emerged as powerful tools capable of automating complex interactions across diverse digital platforms, including web browsers (Zhou et al., 2024; He et al., 2024), computer use (Xie et al., 2024), and mobile applications (Rawles et al., 2023, 2024). Recent advances in vision-language models (VLMs) (Liu et al., 2023; Bai et al., 2025) have greatly enhanced agents' abilities in grounding, visual context understanding, and reasoning, leading to notable progress in GUI-based interaction. However, these agents still struggle with real-world tasks due to their inherently complex, multi-step nature and the limited availability of high-quality training data (Aksitov et al., 2023; Sun et al., 2024).

To address these challenges, several studies leverage web tutorials, which provide step-by-step instructions and rich contextual information (Xu et al., 2025b; Ou et al., 2024; Zhang et al., 2025), to synthesize agent trajectories for model finetuning. However, the quality of such synthetic data remains variable, limiting flexibility and generalization to novel tasks.

We propose leveraging web tutorials as a non-parametric knowledge base at inference time to enhance agents' adaptability across diverse tasks. While this setup resembles the retrieval-augmented generation (RAG) paradigm, it differs from standard RAG pipelines that rely on cleaned, chunked Wikipedia passages for QA (Lewis et al., 2020; Liu et al., 2025; Shi et al., 2024) and introduces two intrinsic challenges: (1) tutorials often encode procedural knowledge that is lost with fixed-length chunking, while leaving them unprocessed leads to long, noisy inputs that may degrade LLM performance (Yu et al., 2024a); and (2) tutorial relevance is not guaranteed—unlike QA corpora, tutorials may not align with the agent's task. Addressing these challenges is critical to fully leveraging tutorials for guiding agents towards desired behaviors.

Motivated by the above challenges, we introduce RAG-GUI, a lightweight VLM as an *adapter* between the agent backbone and the tutorial during inference. RAG-GUI is designed to: (1) assess the relevance between the current task (query and prior actions) and a given tutorial; and (2) generate a useful guidance from the relevant tutorial to assist task completion. Training RAG-GUI begins with supervised finetuning (SFT) with synthetic relevance labels generated by a teacher VLM to bootstrap learning. We then use rejection sampling that allows RAG-GUI to refine both relevance prediction and summarization, based on the assumption that *better guidance improves agent performance*. RAG-GUI is model-agnostic and can serve as a plug-and-play module for any VLM-based agent, making it broadly applicable across different tasks.

We evaluate RAG-GUI on three tasks with two backbones, consistently improving VLM-based

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 17878–17887

agents at inference and often outperforming agents trained on synthetic tutorial trajectories. On the online AndroidWorld benchmark, RAG-GUI achieves 13.3% and 10.7% absolute gains for 7B and 72B backbones, respectively. Moreover, it narrows or surpasses the gap with training-based methods in real-world settings, highlighting its strong generalization across diverse, real-world scenarios.

2 Related Works

Enabling large language models (LLMs) and vision-language models (VLMs) to function as capable GUI agents has been a vibrant area of research. Early approaches (Deng et al., 2023; Gur et al., 2023) improved web agents by leveraging raw HTML elements but did not incorporate visual information. More recent works have demonstrated the promise of VLMs for GUI tasks, capitalizing on their strong visual understanding (Zheng et al., 2024; Yan et al., 2023; Hong et al., 2024). Concurrently, efforts such as (Gou et al., 2025; Wu et al., 2025; Xu et al., 2024) focus on collecting large-scale training data to enhance model capabilities.

Closest to our approach, AgentTrek (Xu et al., 2025b) and TongUI (Zhang et al., 2025) leverage web tutorials by synthesizing training trajectories to improve agent performance. Our work takes a complementary perspective: rather than relying solely on synthetic data generation, we directly integrate tutorial-based guidance at inference time through a lightweight, adaptive retrieval-augmented framework. This design enables flexible, plug-and-play enhancement of VLM-based GUI agents without the need for extensive retraining.

Retrieval-augmented generation (RAG) serves as a powerful technique for knowledge intensive tasks under both text-only (Lewis et al., 2020; Shi et al., 2024; Yu et al., 2024a,b; Xu et al., 2025a) and multimodal scenarios (Yu et al., 2025). However, in our scenario, leveraging tutorials poses unique challenges due to their length and the presence of potentially irrelevant or noisy context, which motivates our design of task-aware guidance generation.

3 Methodology

3.1 Preliminaries

We formulate GUI tasks as a sequential decision-making problem and adopt the SeeAct framework (Zheng et al., 2024). Given a website state s, a task description $g \in \mathcal{G}$, and an action space $\mathcal{A}$, the agent generates a sequence of actions $A = (a_1, a_2, \ldots, a_n) \in \mathcal{A}^n$ to complete the task. At t-th step, the VLM-based agent π selects the next action a_t based on the current environment observation s_t, the task description g, and the history of previous actions $A_t = (a_1, \ldots, a_{t-1})$ as $a_t = \pi(g, s_t, A_t)$. In our tutorial-augmented setting, we first retrieve a set of potentially relevant tutorials $\tau = \{\tau_1, \ldots, \tau_k\}$ using the task description g. A lightweight VLM f_θ then processes each tutorial to produce task-aware guidance as:

$$\hat{\tau}_{i,t} = f_\theta(g, s_t, A_t, \tau_i).$$

Here, the guidance $\hat{\tau} = (\ell, \sigma)$ contains a binary relevance label ℓ with the task-aware guidance σ. The agent then generates the next action based on the current state, task description, action history, and the set of summaries identified as relevant:

$$\hat{a}_t = \pi(g, s_t, A_t, \hat{\sigma}_t),$$

where $\hat{\sigma}_t = \{\hat{\sigma}_{i,t} \mid 1 \le i \le k, \ell_i = 1\}$ is the filtered set of guidance summarized from relevant tutorials. We emphasize that the agent policy π is fixed without parameter update and and only the VLM f_θ is updated.

3.2 Tutorial Collection

Our initial phase involves constructing a comprehensive dataset of GUI tutorials from diverse open-source online repositories. We define a GUI tutorial as a resource that provides sequential, step-by-step instructions for executing a task within a GUI, incorporating both textual descriptions and visual screenshots. Following Qin et al. (2025), we select two large-scale, image-text interleaved datasets, MINT (Awadalla et al., 2024) and OmniCorpus (Li et al., 2024a), as primary sources for potential tutorial data. Additionally, we incorporate articles crawled from the WikiHow website[1] as a supplementary corpus, given its focus on how-to content. Recognizing that MINT and OmniCorpus include a wide spectrum of topics beyond GUI instruction, we conduct a multi-stage filtering process to extract relevant high-quality tutorials. This process comprises three key steps: (1) FastText Filtering, (2) Deduplication, and (3) LLM Labeling. The details for these steps are deferred to Appendix B. After all steps, we obtain a high-quality set of approximately 2.6M GUI tutorials from MINT, OmniCorpus and GUI tutorials, which serves as the task-adaptive guidance resource for the subsequent experiments.

[1] www.wikihow.com

3.3 Optimization for Task-aware Guidance Generation f_θ

Using a frozen VLM to generate task-specific guidance often leads to suboptimal results, as these models are typically fine-tuned for general-purpose tasks. This can create a mismatch between what the adapter f_θ perceives as 'good' guidance and what the agent π effectively utilizes. To this end, we finetune f_θ to align the generated guidance towards target tasks by leveraging the training data containing $\mathcal{D} = \{(g_i, s_i, A_i, a_i)\}_{i=1}^{|\mathcal{D}|}$ where a_i is the ground-truth action step.

SFT Warmup. To warm up f_θ for the target application, we first learn an initial policy via supervised finetuning by imitating the behavior of an expert VLM u. Specifically, we use a frontier model, `GPT-4.1-mini`, to generate high-quality guidance on open-sourced datasets of (state, tutorial, action) pairs as $\mathcal{D}_{\text{SFT}} = \{(x, h)\}$, where $x = (g, s, A, \tau_i)$ is the input containing both environment states, actions and retrieve tutorial, $h \sim u(x)$ is sampled from the expert VLM. This dataset is used to perform supervised fine-tuning on f_θ as $\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(x,h)\sim\mathcal{D}_{\text{SFT}}} \sum_{l=1}^{|h|} \log f_\theta\left(h_l \mid h_{<l}, x\right)$, providing a warm-up for downstream optimization.

Self-guided Rejection Sampling Finetuning. Generating large-scale high-quality distillation data is often infeasible in real-world settings. To improve the model's ability to produce effective guidance from tutorials, we adopt the hypothesis that *good guidance should help the agent take the correct action*. To that end, we repurpose the existing training tuples (g, s_t, a_t) to first retrieve relevant tutorials τ, then train the guidance generation model such that its output increases the likelihood of the agent selecting the correct action as (note that $\hat{\tau}_i = (\hat{\ell}_i, \hat{\sigma}_i)$):

$$p(\hat{a}_t = a_t|g, s_t, \tau_i) = \log \sum_{\hat{\tau}_i} p_\theta(\hat{\tau}_i|\tau_i) p(\hat{a}_t = a_t|g, s_t, \hat{\sigma}_i)$$
$$\geq \underbrace{\mathbb{E}_q[\log p(\hat{a}_t = a_t|g, s_t, \hat{\sigma}_i)] + \mathbb{D}_{\text{KL}}(q \,||\, p_\theta(\hat{\tau}_i|\tau_i))}_{\text{ELBO } \mathcal{L}(p_\theta, q)}$$

Rather than directly optimizing the marginal log-likelihood, we maximize its evidence lower bound (ELBO) $\mathcal{L}(p_\theta, q)$, where $q(\hat{\tau})$ is the posterior over the sampled guidance $\hat{\tau}$. The optimal q is given by:

$$q\left(\hat{\tau}_i \mid g, s_t, a_t, \tau_i\right) = \frac{p_\theta\left(\hat{\tau}_i \mid \tau_i\right) p\left(\hat{a}_t = a_t \mid g, s_t, \hat{\sigma}_i\right)}{\sum_{\hat{\tau}'} p_\theta\left(\hat{\tau}'_i \mid \tau_i\right) p\left(\hat{a}_t = a_t \mid g, s_t, \hat{\sigma}'_i\right)}$$
$$\propto p_\theta(\hat{\tau}_i \mid \tau_i) \cdot p(\hat{a}_t = a_t \mid g, s_t, \hat{\sigma}_i).$$

Assume $p(a_t \mid g, s_t, \hat{\tau}) \propto \mathbb{I}[\hat{a}_t = a_t]$, i.e., the probability is nonzero only when the generated action matches the ground-truth. Then, the overall rejection sampling procedure is as follows:

1. **Sample guidance:** For each tutorial τ_i, draw m candidate guidances $\{\hat{\tau}_i^{(j)}\}_{j=1}^m \sim p_\theta(\hat{\tau} \mid \tau_i)$.
2. **Guideline Filtering:** For each Guideline $\hat{\tau}_i^{(j)}$, let $\hat{a}_t^{(j)} = f(g, s_t, \hat{\sigma}_i^{(j)})$, then only those $(\tau_i, \hat{\tau}_i^{(j)})$ for which $\hat{a}_t^{(j)} = a_t$ are kept, yielding a filtered set $\mathcal{D}_{\text{RSF}}$.
3. **Fine-tune:** Update f_θ by minimizing the RSF Loss of the retained guideline: $\mathcal{L}_{\text{RSF}} = \min\left(-\sum_{(\tau_i,\hat{\tau})\in\mathcal{D}_{\text{RSF}}} \log p_\theta(\hat{\tau} \mid \tau_i)\right)$.

In practice, if the model assigns conflicting relevance labels to the same tutorial across different generations—yet both instances result in the correct action—we discard these examples to prevent introducing ambiguity into the training of the guideline generator f_θ.

3.4 Model Inference

At the inference stage, for example (g, s_t, A_t, a_t) and retrieved tutorials $\tau = \{\tau_1, ..., \tau_k\}$, we first use the guideline generation model f_θ to generate relevant guidelines $(\ell_i, \sigma_i) = f_\theta(g, s_t, A_t, \tau_i)$ for each tutorial τ_i. Next, we filter the generated guidelines to retain only those marked relevant and augment the original prompt with this filtered set as $\hat{\sigma}_t = \{\sigma_i \mid 1 \leq i \leq k, \ell_i = 1\}$. The agent π uses this augmented guideline to make its final action prediction as $\hat{a}_t = \pi\left(g, s_t, A_t, \hat{\sigma}_t\right)$, which updates the environment state s_{t+1} and action history A_{t+1} for the subsequent step.

4 Experiments

4.1 Experiment Setups

Datasets and Evaluation Metrics We evaluate RAG-GUI on one challenging out-of-distribution online dataset Android World (Rawles et al., 2024) and two in-distribution offline datasets Android Control (Li et al., 2024b) and Multimodal-Mind2Web (Deng et al., 2023). The details for these datasets are in Appendix A.

Baselines We compare against the following baselines: (1) **Inference-based methods without tutorials**, including Seeclick (Cheng et al., 2024b), UGround (Gou et al., 2025), OmniParse (Lu

Table 1: The performance of RAG-GUI and baselines on three benchmarks. M2W, AC and AW stands for mm-Mind2Web, AndroidControl and AndroidWorld, respectively.

Models		AW	M2W - Cross Task			M2W - Cross Website			M2W - Cross Domain			AC - High	AC - Low
Planner	**Grounder**	**SR**	**Ele. Acc**	**Op. F1**	**Step SR**	**Ele. Acc**	**Op. F1**	**Step SR**	**Ele. Acc**	**Op. F1**	**Step SR**	**Step Acc.**	**Step Acc.**
GPT-4o (2024)	SeeClick (2024a)	–	32.1	–	–	33.1	–	–	33.5	–	–	41.8	52.8
GPT-4o (2024)	UGround (2025)	32.8	47.7	–	–	46.0	–	–	46.6	–	–	48.4	62.4
GPT-4V (2023)	OmniParse (2024)	–	42.4	**87.6**	39.4	41.0	**84.8**	36.5	45.5	85.7	42.0	–	–
GPT-4o (2024)		34.5 (SOM)	5.7	77.2	4.3	5.7	79	3.9	5.5	**86.4**	4.5	20.8	19.4
Claude Computer Use (2025)		27.9	62.7	84.7	53.5	59.5	79.6	47.7	64.5	85.4	56.4	12.5	19.4
AgentTrek 7B (2025b)		–	45.5	84.9	40.9	40.8	82.8	35.1	48.6	84.1	42.1	–	–
Qwen2.5-VL-7B (2025)		22.0	57.9	82.5	45.3	58.6	80.0	41.6	57.5	83.4	45.2	52.9	72.5
w/ vanilla RAG		22.4	59.0	83.1	46.0	57.4	80.4	41.3	57.9	83.5	45.3	55.8	74.3
RAG-GUI-7B		35.3	63.9	84.1	51.5	60.7	83.6	46.1	60.3	84.5	47.8	59.6	81.6
w/o RSF		32.8	59.9	83.5	46.8	58.6	82.3	44.2	59.5	84.5	47.3	54.9	79.4
Qwen2.5-VL-72B (2025)		35.0	63.4	81.7	51.8	64.0	81.8	49.6	56.5	84.1	46.2	57.0	78.6
w/ vanilla RAG		37.5	58.6	82.9	45.8	63.8	81.8	49.0	60.0	81.6	49.8	56.2	76.7
RAG-GUI-72B		45.7	**69.5**	85.0	**56.8**	**68.1**	83.1	**53.0**	**66.3**	85.9	**55.0**	**60.7**	**84.6**
w/o RSF		44.4	66.3	84.2	53.8	66.9	82.9	52.1	63.6	84.7	52.2	58.7	80.9
—			*Training-based LLMs using in-distribution training data. For Reference Only.*										
AgentTrek 7B (w/ M2W) (2025b)		–	60.8	88.9	55.7	57.6	88.1	51.4	56.0	87.5	52.6	–	–
Aguvis 7B (2024)		–	64.2	89.8	60.4	60.7	88.1	54.6	60.4	89.2	56.6	61.5	80.5
Aguvis 72B (2024)		26.1	69.5	90.8	64.0	62.6	88.6	56.5	63.5	88.5	58.2	66.4	84.4
UI-TARS 7B (2025)		33.0	73.1	92.2	67.1	68.2	90.9	61.7	66.6	90.9	60.5	72.5	90.8
UI-TARS 72B (2025)		**46.6**	74.7	92.5	68.6	72.4	91.2	63.5	68.9	91.8	62.1	74.7	91.3

et al., 2024), GPT-4o (Hurst et al., 2024), and Claude (Anthropic, 2025); (2) **Tutorial-based inference method**: vanilla RAG (Lewis et al., 2020), which augments tasks, states, and previous actions with top retrieved tutorials; (3) **Tutorial-based synthetic data generation**: AgentTrek (Xu et al., 2025b), which finetunes the VLM using synthetic trajectories generated from web tutorials. We also report results from finetuning-based methods that leverage large amounts of grounding data and agent trajectories (Xu et al., 2024; Qin et al., 2025), but mainly for reference. Designing approaches to incorporate our tutorial guideline generation into finetuning is an interesting direction for future work.

Implementation Details Our guideline generation model f_θ is built upon the `Qwen-2.5-VL-7B` backbone model. To create task-specific guideline for training, we utilize AitW (Rawles et al., 2023), AMEX (Chai et al., 2024), and GUIAct (web-multi and web-single) (Chen et al., 2024) as seed datasets. We use E5 (Wang et al., 2022) as the default embedding model for tutorial retrieval. For the SFT warmup, we train the backbone model for 1 epoch using a learning rate of $1e-5$ and a cosine scheduler. During rejection sampling, we set the temperature to 1.0. For the RSF stage, we continue training the model for one additional epoch, building on the SFT checkpoint, with a learning rate of $5e-6$. For evaluation, we use `Qwen-2.5-VL-7B/72B` (Bai et al., 2025) as the agent backbone model.

4.2 Main Experiments

Table 1 exhibits the experiment results on three tasks. We have the following findings:

Experiment on Offline Tasks Evaluations on AndroidControl and mm-Mind2Web reveal that naively incorporating tutorials via a standard RAG pipeline yields limited improvements, particularly for smaller backbones (7B), due to LLMs' constrained ability to process tutorials directly. In contrast, RAG-GUI achieves notable gains (4.4% on Mind2Web and 6.3% on AndroidControl) and narrow the gap between training-based methods and training-free models, verifying the benefit of carefully integrating tutorials. RAG-GUI also outperforms AgentTrek trained with synthetic trajectories converted from tutorials. We hypothesize that AgentTrek is hampered by dataset quality issues and still relies on high-quality trajectories to achieve strong performance.

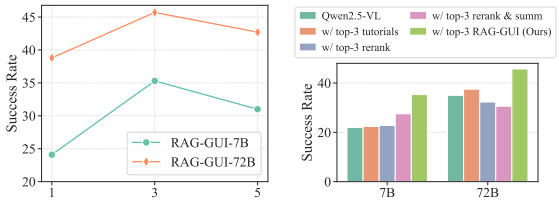
Experiment on Online Tasks We evaluate RAG-GUI on AndroidWorld, an online environment requiring multi-step reasoning that mirrors real-world scenarios. Remarkably, incorporating guided summarization leads to more than 10% success rates compared to direct inference, which further validates the effectiveness of our proposed approach.

Effect of Two-Stage Training The results also demonstrate that incorporating self-guided rejection sampling finetuning (RSF) consistently improves performance, emphasizing the benefit of enabling the model to self-evolve and avoid costly manual annotation.

Effect of Textual Guidance Generator In order to evaluate the capability of RAG-GUI to generate high-quality textual guidance, we conduct additional experiments that leverage frozen

Table 2: The performance of RAG-GUI and frozen Qwen as textual guidance generators on the AndroidWorld benchmark.

Models	AW SR
Qwen2.5-VL-7B (2025)	22.0
w/ vanilla RAG	22.4
w/ textual guidance from frozen Qwen2.5-VL-7B	27.5
RAG-GUI-7B	35.3
Qwen2.5-VL-72B (2025)	35.0
w/ vanilla RAG	37.5
w/ textual guidance from frozen Qwen2.5-VL-72B	37.9
RAG-GUI-72B	45.7



(a) Different k (b) Alternative Designs

Figure 1: Additional studies.

Qwen2.5-VL-7B/72B as textual guidance generator for the Android World dataset. As shown in Table 2, RAG-GUI consistently outperforms the frozen models generating textual guidance for both 7B and 72B backbones. This demonstrates that the task-adaptive fine-tuning and guidance optimization in RAG-GUI lead to substantial improvements over simply using an online LLM to generate guidance, validating the effectiveness of our approach.

Additional Studies We present additional analyses on guideline generation in Figure 1. As illustrated in Figure 1(a), retrieving $k = 3$ tutorials yields optimal performance—too many tutorials result in lengthy contexts that may confuse the model, while too few ($k = 1$) risk missing relevant information. Furthermore, Figure 1(b) compares our guideline generation framework against alternative designs, including prompting Qwen-2.5-VL-72B for reranking or using it solely as a summarization model without task-specific adaptation. The inferior performance of these variants verifies the effectiveness of our proposed guideline generation approach. More cases studies are presented in Appendix E for better illustration.

5 Conclusion

In this work, we introduce RAG-GUI, a VLM–based guideline generation framework that enables agents to effectively harness the vast information contained in web tutorials. We curate a large dataset of 2.6 million tutorials and adopt a two-stage training approach to produce high-quality guideline. Experiments across three benchmarks with two VLM backbones show consistent performance gains of RAG-GUI, ranging from 2.6% to 13.3% across model sizes. Notably, the improvements are more pronounced in online environments that closely simulate real-world scenarios, demonstrating the strong generalization capability of RAG-GUI in assisting VLM-based Agents.

Limitations

Our study is limited to inference-time evaluation without any model finetuning, which may constrain the achievable performance gains through adaptation. Furthermore, all experiments are performed exclusively on the Qwen-VL-series models, limiting the generalizability of our results to other architectures such as LLaVA (Liu et al., 2023). While incorporating guideline generation introduces additional inference latency, the consistent performance improvements justify this overhead. Nonetheless, future work could focus on optimizing the efficiency of both retrieval and guideline generation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Renat Aksitov, Sobhan Miryoosefi, Zonglin Li, Daliang Li, Sheila Babayan, Kavya Kopparapu, Zachary Fisher, Ruiqi Guo, Sushant Prakash, Pranesh Srinivasan, and 1 others. 2023. Rest meets react: Self-improvement for multi-step reasoning llm agent. *arXiv preprint arXiv:2312.10003*.

Anthropic. 2025. Introducing computer use, a new claude 3.5 sonnet, and claude 3.5 haiku.

Anas Awadalla, Le Xue, Oscar Lo, Manli Shu, Hannah Lee, Etash Guha, Sheng Shen, Mohamed Awadalla, Silvio Savarese, Caiming Xiong, and 1 others. 2024. Mint-1t: Scaling open-source multimodal data by 10x: A multimodal dataset with one trillion tokens. *Advances in Neural Information Processing Systems*, 37:36805–36828.

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*.

Yuxiang Chai, Siyuan Huang, Yazhe Niu, Han Xiao, Liang Liu, Dingyu Zhang, Peng Gao, Shuai Ren,

and Hongsheng Li. 2024. Amex: Android multi-annotation expo dataset for mobile gui agents. *arXiv preprint arXiv:2407.17490*.

Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, and 1 others. 2024. Guicourse: From general vision language models to versatile gui agents. *arXiv preprint arXiv:2406.11317*.

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

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024b. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9313–9332, Bangkok, Thailand. Association for Computational Linguistics.

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.

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

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2023. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. WebVoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890, Bangkok, Thailand. Association for Computational Linguistics.

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.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Qingyun Li, Zhe Chen, Weiyun Wang, Wenhai Wang, Shenglong Ye, Zhenjiang Jin, Guanzhou Chen, Yinan He, Zhangwei Gao, Erfei Cui, and 1 others. 2024a. Omnicorpus: A unified multimodal corpus of 10 billion-level images interleaved with text. *arXiv preprint arXiv:2406.08418*.

Wei Li, William E Bishop, Alice Li, Christopher Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. 2024b. On the effects of data scale on ui control agents. *Advances in Neural Information Processing Systems*, 37:92130–92154.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Tianci Liu, Haoxiang Jiang, Tianze Wang, Ran Xu, Yue Yu, Linjun Zhang, Tuo Zhao, and Haoyu Wang. 2025. Roserag: Robust retrieval-augmented generation with small-scale llms via margin-aware preference optimization. *arXiv preprint arXiv:2502.10993*.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. 2024. Omniparser for pure vision based gui agent. *arXiv preprint arXiv:2408.00203*.

Tianyue Ou, Frank F Xu, Aman Madaan, Jiarui Liu, Robert Lo, Abishek Sridhar, Sudipta Sengupta, Dan Roth, Graham Neubig, and Shuyan Zhou. 2024. Synatra: Turning indirect knowledge into direct demonstrations for digital agents at scale. *arXiv preprint arXiv:2409.15637*.

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*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.

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*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. 2025. OS-ATLAS: Foundation action model for generalist GUI agents. In *The Thirteenth International Conference on Learning Representations*.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, and 1 others. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.

Ran Xu, Hui Liu, Sreyashi Nag, Zhenwei Dai, Yaochen Xie, Xianfeng Tang, Chen Luo, Yang Li, Joyce C. Ho, Carl Yang, and Qi He. 2025a. SimRAG: Self-improving retrieval-augmented generation for adapting large language models to specialized domains. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11534–11550, Albuquerque, New Mexico. Association for Computational Linguistics.

Yiheng Xu, Dunjie Lu, Zhennan Shen, Junli Wang, Zekun Wang, Yuchen Mao, Caiming Xiong, and Tao Yu. 2025b. Agenttrek: Agent trajectory synthesis via guiding replay with web tutorials. In *The Thirteenth International Conference on Learning Representations*.

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*.

An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, and 1 others. 2023. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. *arXiv preprint arXiv:2311.07562*.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025. VisRAG: Vision-based retrieval-augmented generation on multi-modality documents. In *The Thirteenth International Conference on Learning Representations*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024a. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14672–14685, Miami, Florida, USA. Association for Computational Linguistics.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. RankRAG: Unifying context ranking with retrieval-augmented generation in LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

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

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

A Details for Evaluation Tasks

Android World assesses performance within a virtual Android emulator, with 116 distinct tasks across 20 mobile applications. For this dataset, we report Step accuracy, which measures the correctness of the final stage of task execution. Multimodal-Mind2Web focuses on interactions

within web environments, with 1,013 tasks spanning 100 different websites. We utilize three metrics for evaluation: Element accuracy (Ele. Acc.), Operation F1 (Op. F1), and Step success rate (Step SR). Android Control is designed for the mobile environment. Following the methodology of Li et al. (2024b), we randomly sample 500 tasks to form our test set. Performance on this dataset is measured by Step accuracy. Notably, Android Control includes both high-level instructions, requiring the model to simultaneously plan and execute actions, and low-level instructions, where the model only needs to execute a predefined action.

B Details for Tutorial Collection

FastText Filtering Given the substantial scale of the MINT (1054M documents) and OmniCorpus (988M documents) datasets, we employ the computationally efficient FastText classifier (Joulin et al., 2016) for an initial filtering of potential tutorials. Leveraging the inherent feature of the WikiHow corpus, which categorizes its articles by topic, we curated a positive training set by extracting approximately 24K samples classified under the "Computers and Electronics" category. To create a balanced training set, we randomly selected 12K negative samples from WikiHow articles belonging to other categories and an additional 12K random samples from the MINT dataset. This resulted in a training corpus of 48K labeled examples. The trained FastText classifier is then applied to filter the MINT and OmniCorpus datasets, after which 26.5M documents from MINT and 52M documents from OmniCorpus are retained, forming a preliminary candidate set of tutorials.

Deduplication To mitigate redundancy within the candidate tutorial set derived from MINT and OmniCorpus (e.g., content duplication across different URLs and overlap between the two datasets), we performed content-based deduplication on the filtered documents from the previous stage. This process yields a refined set of approximately 7.5M unique documents from MINT and 24M unique documents from OmniCorpus.

LLM Labeling To achieve a more precise selection of GUI-related tutorials, we employ Qwen2.5-7b-it (Qwen et al., 2025) for a subsequent classification step. We prompt the LLM to analyze the content of each document and identify those specifically providing instructions for GUI tasks. This more granular filtering process further reduces the number of false positives, resulting in a high-quality set of approximately 0.74M GUI tutorials from MINT and 1.8M from OmniCorpus. The prompt format is detailed in Appendix C. Aggregating these with the initial 24K positive samples sourced from WikiHow yields a final pool of 2.6M GUI tutorials, which serves as the task-adaptive guidance resource for the subsequent experiments.

C Prompt Format for LLM Labeling

Figure 2 shows the prompts used for estimating whether the content is a GUI-related tutorial.

System Prompt: You are an assistant that classifies content based on specific criteria. Your task is to evaluate whether a given piece of content serves as a tutorial specifically related to graphical user interfaces (GUI), such as for web applications, desktop applications, or operating systems.
Classification Criteria The content qualifies as a GUI-related tutorial if it meets the following conditions: 1. It includes a task description outlining what needs to be achieved. 2. It provides clear step-by-step instructions for interacting with a GUI, such as: - Step 1: Open the application - Step 2: Navigate to the settings menu.

User Prompt: Given the below content, predict if the content is a GUI-related tutorial or not. Output 'Yes' if the above content is a GUI-related tutorial and 'No' if it is not. Provide only 'Yes' or 'No' as the output.
{content}

Assistant Prompt:
Output: {Generated relevant label}

Figure 2: Prompt for using Qwen-2.5 for generating quality labels for web tutorials.

D Prompt Format for Inference

The prompt template for guidance generation is listed in Figure 3.

E Case Study

We present a case study in Table 3 from Android Control to demonstrate the task guidance generated by RAG-GUI based on goal, previous actions

Table 3: A case study from Android Control demonstrating the task guidance generated by RAG-GUI. Green text stands for the alignment of guidance and action goal while blue text justify the ability of RAG-GUI in generating customized guidance by taking the previous actions into the consideration.

Goal	I want to search for a flight from Rotterdam to Puerto Natales to visit my favorite travel destination Patagonia on the MakeMyTrip app for 11th January.
Previous Actions	Step 1: Open MakeMyTrip app. Step 2: Click on the Flights option.
Retrieved Tutorial	Search In the Main Page select the "Flights" tab, and a form is shown with the information you submit for search flights. Complete the Form In the form, enter the information for your reservation: Depart and return cities, depart and return dates, amount of tickets, etc. After filling out the form, click on: Select Flights You will see a list with the flights that match your search criteria in previous step. Verify pricing information, dates, etc; and select the flight you want by clicking on this button: Verify The reservation summary information is shown,with Flight data, prices, purchase conditions, etc. To continue, click on the following button: Enter your Data The form is shown to collect the information for passengers, contact, etc. Complete the form and click on "Reserve". Payment Information Now the form to enter payment information, that could be Credit Card or Deposit. IMPORTANT: Your credit card is not charged on this step. It is used only for reservation validation. A Sales executive will contact you to complete the payment process. Click on "Accept Terms and Conditions" and then on "Confirm Trip". Reservation Complete Congratulations! you have reserved your flight ticket. A Sales Executive will contact you to complete the payment and issue the Electronic Ticket. Packages Hotels Air Tickets Search for Tourism Packages Ingrese el nombre del hotel o la ciudad: Partners Hotels recommended
Task Guidance Generated by RAG-GUI	In the main page of the MakeMyTrip app, after selecting the "Flights" tab, proceed to fill out the flight search form by entering the departure city as Rotterdam, the destination city as Puerto Natales, and set the travel date to 11th January. After completing the form, initiate the search to view available flights matching these criteria. Select the preferred flight from the search results to verify details such as pricing and dates. Continue by entering passenger and contact information, then proceed to payment information where you accept terms and confirm the trip to complete the booking.

and the retrieved tutorial. We observe that the retrieved tutorial contains noise, much of which is irrelevant to the specific task at hand.. However, the task guidance generated by RAG-GUI is much more concise, while still retaining the important information. Furthermore, the generated guidance seamlessly integrates contextual data derived from both the stated goal and the prior actions. Crucially, it provides specific, actionable cues on how to execute the subsequent step, directly addressing the requirements of the current task in light of the performed interactions.

System Prompt: You are a helpful assistant that aim to use a tutorial for completing a specific GUI-based task. Given a query, previous actions and a related tutorial, your task is to first identify the relevance between the tutorial and the task and previous actions. Then, if the tutorial is relevant, please generate a concise summary for the tutorial with the following requirements: - 1. You should only retain the most relevant information from the tutorial that help to solve the task conditioned on previous actions. - 2.Please make sure to include detailed guidance from the tutorial if it is helpful to solve the problem based on the current state. - 3. If the tutorial is not helpful or relevant to the task, then please only generate the score without summary. Use the following format in your output: <score> [the relevance score (0 or 1)] </score> <summary> [Your summary of the tutorial conditioned on the task and previous actions] </summary>

User Prompt: The user query: {instruction}
Task progress (You have done the following operation on the current device): {previous_actions};
Tutorial: {tutorial}

Assistant Prompt:
Output: {Model generated guidance}

Figure 3: Prompt for guidance generation.