WinSpot: GUI Grounding Benchmark with Multimodal Large Language Models

Zheng Hui, **Yinheng Li**, **Dan Zhao**, **Colby Banbury**, **Tianyi Chen**, **Kazuhito Koishida**
Columbia University, Microsoft
{zackhui, yinhengli, tianyi.chen, colbybanbury }@microsoft.com,
{zh2483}@columbia.edu, {dz1158}@nyu.edu

Abstract

Graphical User Interface (GUI) automation relies on accurate GUI grounding. However, obtaining large-scale, high-quality labeled data remains a key challenge, particularly in desktop environments like Windows Operating System (OS). Existing datasets primarily focus on structured web-based elements, leaving a gap in real-world GUI interaction data for non-web applications. To address this, we introduce a new framework that leverages LLMs to generate large-scale GUI grounding data, enabling automated and scalable labeling across diverse interfaces. To ensure high accuracy and reliability, we manually validated and refined 5,000 GUI coordinate-instruction pairs, creating WinSpot—the first benchmark specifically designed for GUI grounding tasks in Windows environments. WinSpot provides a high-quality dataset for training and evaluating visual GUI agents, establishing a foundation for future research in GUI automation across diverse and unstructured desktop environments[1].

1 Introduction

Multimodal Large Language Models (MLLMs) exhibit impressive visual understanding and reasoning (Gandhi et al., 2023; Liu et al., 2024; Li et al., 2024; Zhang et al., 2025; Li et al., 2025b), enabling automation in complex real-world scenarios (Ai et al., 2024; Hui et al., 2025b). Among these, Graphical User Interface (GUI) automation has emerged as a critical application, where agents interpret on-screen elements and execute context-relevant actions for tasks such as software testing and application management (Yang et al., 2023a; Li et al., 2020; Wang et al., 2024b).

Despite significant advances in web and mobile GUI automation (Bavishi et al., 2023; Yang et al., 2023a; Cheng et al., 2024; Wang et al., 2024a; Hui et al., 2025a), GUI in Windows desktop environments remain largely unexplored, despite Windows system widespread use in professional and enterprise applications. This gap is particularly challenging because Windows applications lack a standardized UI representation such as HTML or DOM structures, requiring GUI grounding to rely purely on visual perception. Furthermore, Windows interfaces exhibit highly diverse layouts, where applications designed using different frameworks (e.g., Win32, UWP, Electron) follow inconsistent UI structures. Additionally, overlapping windows introduce ambiguity in detecting actionable elements, as interactable regions may be partially or fully obscured. Unlike web applications, where ARIA (Accessible Rich Internet Applications) attributes provide accessibility metadata, many Windows applications lack structured accessibility trees (a11y trees), making it difficult to extract UI component descriptions programmatically. Existing screenshot-based methods (Bavishi et al., 2023; Cheng et al., 2024) show promise but lack a large-scale, standardized benchmark specifically designed for Windows GUI automation. Without such a benchmark, researchers face challenges in systematically measuring progress, comparing approaches, and addressing the distinct complexities of desktop interfaces.

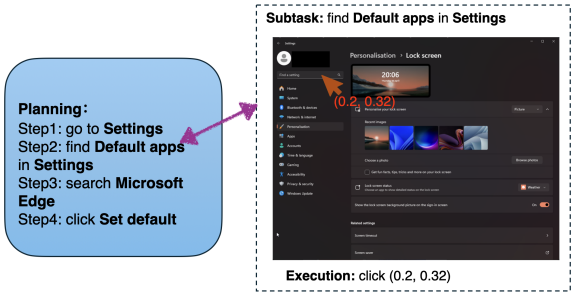

Figure 1: GUI grounding: locating actionable UI elements based on instructions.

To fill this void, we introduce **WinSpot**, a *large-*

[1]https://github.com/zackhuiiiii/WinSpot.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1086–1096

scale benchmark specifically designed for GUI grounding in Windows environments. Our main contributions are summarized below:

- **Two-Stage Labeling Framework.** We propose a scalable approach that utilizes MLLMs to generate coordinate-instruction pairs from diverse Windows screenshots, significantly reducing the initial labeling burden. Importantly, our method relies exclusively on raw screenshots, ensuring seamless adaptation across different Windows applications.
- **WinSpot: A First-of-its-Kind Windows GUI Benchmark.** Expanding on our two-stage framework, we introduce **WinSpot**—a comprehensive dataset with **over 5,000** human-validated coordinate-instruction pairs, covering diverse Windows environments, 21 times larger than previous benchmarks.

2 Related Work

2.1 UI Screen Understanding Dataset

A variety of datasets (Moran et al., 2018; He et al., 2021; Wu et al., 2023) have been developed to support UI modeling, primarily in the mobile domain. For instance, the AMP dataset (Zhang et al., 2021), containing 77k screens from 4,068 iOS apps. Another significant resource is Rico (Deka et al., 2017), the largest publicly available dataset for Android apps UI understanding. In the broader web and OS domain, datasets such as Mind2Web (Deng et al., 2024) , Visual-WebArena (Koh et al., 2024), and Windows Arena (Bonatti et al., 2024) offer simulated environments for various tasks. Existing GUI Grounding datasets overwhelmingly focus on mobile and web platforms, leaving desktop environments underexplored. The closest dataset related to desktop UI understanding is SeeClick (Cheng et al., 2024) and Os-atlas (Wu et al., 2024b), though it predominantly targets cross-platform settings and lacks a specific focus on Windows. Our work fills this gap by introducing a dataset tailored to desktop environments, particularly for Windows OS, which marks a novel contribution to the field.

3 Method

3.1 Data Construction

Unlike previous work (Cheng et al., 2024) that focuses on cross-domain tasks and structured data for training dataset construction, our approach targets the Windows OS (Figure 2). We propose

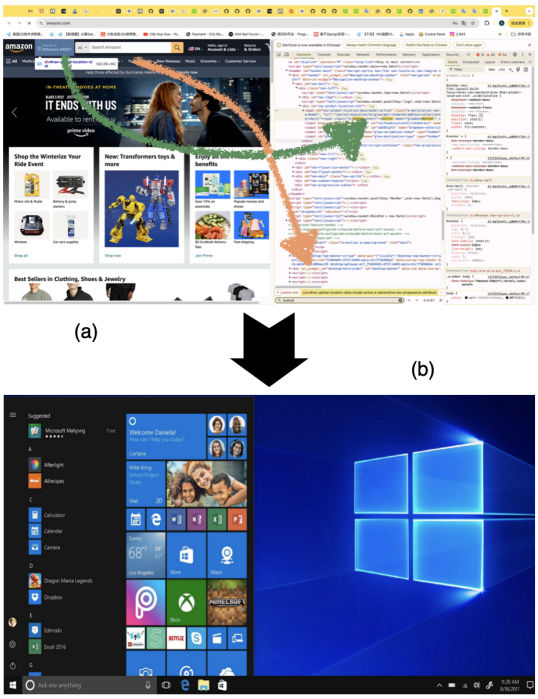

Figure 2: (a) Traditional methods rely on HTML or DOM files to locate icons during data construction. (b) Our proposed data alignment framework requires only raw screenshot images.

Instruction-Interactable Region Alignment (Figure 3), leveraging MLLMs to generate training data without HTML elements or accessibility trees. Since Windows applications lack a standardized UI representation and display diverse layouts across frameworks (e.g., Win32, UWP, Electron), our method relies entirely on visual cues for effective GUI grounding.

We first retrieve and filter images via the Bing API, then verify quality with Phi3-vision (Abdin et al., 2024b). Our goal is to collect diverse, representative Windows screenshots. We query screenshots of 700+ top apps from the Microsoft Store[2]. This model verifies resolution and screenshot validity. Quality-approved images are randomly added to our data bank. We expand our dataset via Bing API's similar image feature. Images failing quality checks are discarded.

Once filtered, we apply a proprietary Bert model with ViT encoder to perform icon grounding on the selected images. The ViT-Bert model generates bounding boxes around interactable icons in the images, which we use to create structured data. We then use GPT-4o for aligning the filtered images with corresponding icon descriptions. This align-

[2]https://www.microsoft.com/en-us/store/most-popular/apps/pc

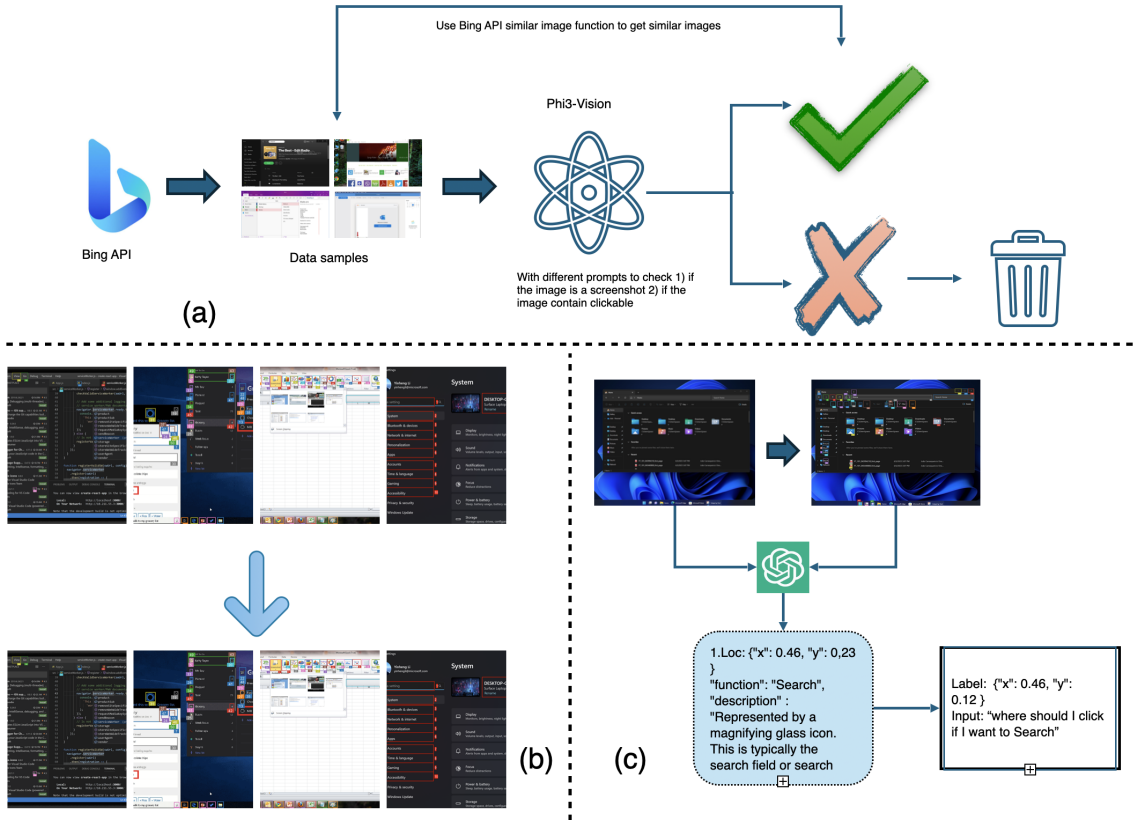

Figure 3: Overview of the Data Alignment Process: (a) illustrates the data filtering strategy using the Phi3 vision model, (b) shows the input and output of icon grounding with the in-house ViT-BERT model, and (c) the use of LLMs for GUI and description alignment.

ment process serves multiple purposes: 1) By using expensive models like GPT-4o only in the data alignment stage, we reduce computational costs while maintaining accuracy during the reasoning and inference processes. 2) Previous work (Zheng et al., 2024) has shown that providing GPT-4V with screenshots that only include bounding boxes and IDs can be misleading. The limited ability of GPT-4V to extract semantic information while predicting actions poses a challenge. To address this, our automated labeling pipeline incorporates semantic cues directly into the images during data construction, primarily by enhancing the prompts generated by GPT-4V. 3) By enriching the dataset with diverse semantic information, we ensure that the subsequent click agents can handle distributed tasks more effectively, improving overall performance. In addition to the data collected via the Bing API, we incorporated 500 images from CoVA dataset(Kumar et al., 2022), 500 images from WebSight dataset(Laurençon et al., 2024) to further enhance our training set. Result in a dataset around 60K to train our model. For more examples about the data construction, please refer to Appendix B.

3.2 WinSpot Benchmark

The *WinSpot* dataset consists of over 5,000 annotated [3] screenshots from 14 core Windows applications, each representing unique interaction types and layout structures. Examples from *WinSpot* are shown in Figure 4. The applications and their respective contributions to the dataset are shown in Figure 5. Each screenshot in *WinSpot* contains multiple interactable regions, such as buttons, menus, and icons, each annotated with its corresponding function. These annotations include bounding boxes around the interactable elements and their associated semantic descriptions, which are aligned with natural language instructions for both grounding and task prediction tasks. This variety ensures that *WinSpot* provides a challenging and comprehensive evaluation framework for GUI agents, enabling robust testing of both interaction precision and generalization across different applications.

WinSpot presents a diverse array of tasks, including file navigation, system settings adjustment, and text input, as well as more specialized tasks such

[3] More annotation detail in Appendix C

Figure 4: WinSpot examples

as process management in Task Manager and command execution in Command Prompt. These tasks encompass a wide range of complexity, from simple button clicks to more involved interactions that require an understanding of application-specific layouts. In addition to supporting GUI grounding tasks $P(y|S, x)$, *WinSpot* also be used in reverse tasks $P(x|S, y)$, where the model must predict the description of a given GUI element based on its location in the screenshot. This two-way task formulation enhances the evaluation by testing both the agent's understanding of visual cues and its ability to map interactions to the correct interface components.

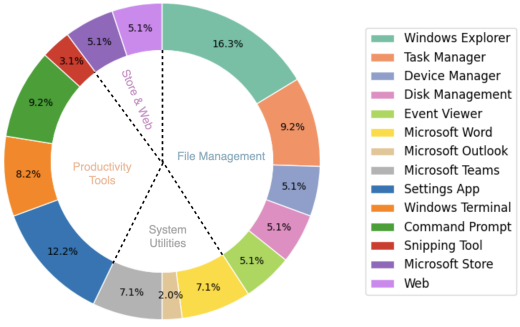

Figure 5: WinSpot Category

4 Experiments and Results

In this section, we evaluate both general-purpose models and GUI-specific models on our newly introduced **WinSpot** benchmark.

4.1 Baselines and Evaluation Metric

We compare multiple baselines, including general-purpose MLLMs (e.g., GPT-4o (OpenAI et al., 2024), GPT-4V (OpenAI et al., 2024), Phi3-Vision (Abdin et al., 2024a), MinGPT-v2 (Chen et al., 2023)) and GUI-focused models (e.g., Fuyu (Bavishi et al., 2023), CogAgent (Wang et al., 2024c), SeeClick (Cheng et al., 2024), Uground (Gou et al., 2024)). Consistent with prior studies on GUI grounding (Li et al., 2022; Yang et al., 2023b; Cheng et al., 2024), we adopt **click accuracy** as our primary metric. A prediction is considered correct if the model's predicted click coordinates fall within the bounding box of the ground-truth.

4.2 Results

Table 1 presents the click accuracy of various models across the four major subcategories of the WinSpot benchmark: File Management, System Settings, Productivity Tools, and MS Store & Web applications. These categories span a wide range of GUI interaction patterns allowing us to assess both generalization and domain sensitivity of each model. The best-performing model is **Uground**, which achieves a remarkable **44.2%** overall accuracy, significantly outperforming all other baselines. Its dominance is particularly evident in **System Settings** and **MS Store & Web**, where it reaches **51.4%** and **82.4%** respectively.

Among the general-purpose MLLMs, **GPT-4V** and **GPT-4o** show relatively higher click accuracy (**18.3%** and **16.5%**, respectively), with notable strengths in the **MS Store & Web** category—where visual layout conventions tend to be more standardized and semantically interpretable. This aligns with prior observations that LLMs pretrained on web data tend to generalize better in semi-structured interfaces but struggle with unstructured system UIs. However, their low scores in **System Settings** and **File Management** (e.g., GPT-4V: 6.3%, GPT-4o: 7.5%) reveal key limitations when navigating system-level layouts, likely due to the absence of such interfaces in their training data and the lack of spatial attention mechanisms specialized for desktop contexts. **Phi-3.5 Vision** and **MiniGPTv-2**, both smaller open-source models,

Method	Size	WinSpot				Total Task
		File Management	System	Productivity Tools	MS Store & Web	
MiniGPTv-2	7B	0%	0.6%	2.2%	5.8%	1.7%
GPT-4V	Unkown	8.0%	6.3%	15.6%	58.1%	18.3%
GPT-4o	Unkown	8.8%	7.5%	14.1%	47.7%	16.5%
Phi-3.5 Vision	4.2B	4.7%	5.8%	3.7%	25.5%	7.9%
Fuyu	8B	5.0%	9.2%	7.1%	34.3%	9.4%
CogAgent	18B	6.4%	8.1%	10.5%	60.8%	13.8%
SeeClick	9.6B	8.6%	16.6%	18.9%	70.6%	20.1%
GUIAct-Qwen	7B	10.8%	6.1%	13.6%	78.4%	18.0%
Uground	7B	27.2%	51.4%	45.4%	82.4%	44.2%

Table 1: Evaluation of Various Methods on WinSpot Subcategories

perform poorly across all subcategories, with overall accuracy below 10%. These results reinforce the importance of scale, training modality, and data coverage in grounding tasks. Interestingly, **Phi-3.5 Vision** performs slightly better in the **MS Store & Web** category, suggesting even smaller models can benefit from interface regularity if provided with sufficient multimodal alignment. Specialized GUI models such as **CogAgent**, **Fuyu**, **SeeClick**, and **GUIAct-Qwen** fall in an intermediate performance range (between **9.4%** and **20.1%** overall), with **SeeClick** standing out as the strongest among them. Notably, **SeeClick** attains **70.6%** accuracy on **MS Store & Web**, highlighting its suitability for commercial UI tasks, but only **16.6%** in **System Settings**, pointing to challenges in less standardized environments. Similarly, **GUIAct-Qwen** achieves competitive results in web-based domains but lacks consistency across system-heavy tasks, suggesting an over-reliance on pretraining priors that fail to capture the visual intricacies of Windows system utilities.

5 Discussion and Future Work

Our findings highlight a clear performance divide between general-purpose MLLMs and domain-specific GUI models. Generalist models such as GPT-4o and GPT-4V demonstrate only modest proficiency in GUI grounding, reflecting their limited pretraining exposure to Windows UI paradigms. Meanwhile, specialized models like Uground and SeeClick perform significantly better, particularly in structured tasks like web and app store interactions. However, even these tailored models struggle with system-level operations—such as task management, file navigation, or control panel interactions—where contextual reasoning and fine-grained spatial precision are required. This underscores a broader insight: current models, despite their size and multimodal capacity, lack robust spatial reasoning and memory mechanisms necessary for GUI automation in real-world settings. WinSpot helps uncover these limitations by evaluating not just interaction precision, but also the ability of models to align natural language instructions with semantically meaningful UI regions. Furthermore, this work is situated within a broader movement in NLP and multimodal learning: applying LLMs to real-world utility tasks beyond traditional text benchmarks. With growing industrial interest in automating workflows, testing software, and enabling human-in-the-loop systems, GUI agents will increasingly become critical enablers. Our benchmark and methodology lay the groundwork for these systems, while also exposing their current gaps. Going forward, we advocate for more research at the intersection of vision-language grounding, procedural planning (Li et al., 2025a), and user intent modeling. In particular, incorporating temporal dynamics (Jiang et al., 2025), multi-turn interactions (Liu et al., 2025), and UI state tracking may bridge the gap between static grounding and true GUI manipulation. Additionally, as LLM-driven agents are deployed in productivity tools, safety (Hui et al., 2024a; Zhang et al., 2024; Wu et al., 2024a) and interpretability will become pressing concerns—especially in high-stakes domains like healthcare, finance, and enterprise automation. In summary, WinSpot offers a much-needed benchmark for evaluating GUI grounding in Windows environments and serves as a testbed for the next generation of GUI agent. It pushes the research community to build models that are not only linguistically fluent but also visually and operationally grounded in environments where real users work.

6 Limitations

While WinSpot establishes a valuable benchmark for GUI grounding in Windows environments, several limitations remain. First, the dataset is derived mainly from a curated selection of popular Windows applications and supplemental sources, which may not capture the full diversity of desktop interfaces—especially those used in specialized or enterprise settings. Second, our automated labeling pipeline, although designed to reduce manual effort, relies on MLLMs for generating semantic cues; any inaccuracies or biases inherent in these models can propagate into the final annotations. Third, our evaluation metric, click accuracy, offers a simplified perspective on interaction performance, potentially overlooking nuanced aspects of user engagement such as multi-step workflows or gesture dynamics. Finally, the framework is optimized for static screenshots and may not generalize well to dynamic or adaptive interfaces that evolve in real time.

7 Ethical Considerations

In constructing WinSpot, we took deliberate steps to safeguard user privacy and ensure ethical data practices. All screenshots were rigorously reviewed and post-processed to remove personal or sensitive information before inclusion in the dataset. However, the selection process for source applications may introduce biases that affect the representativeness of the dataset. Additionally, the automated labeling pipeline's reliance on MLLMs could inadvertently propagate existing biases present in these models. We advocate for continuous audits and transparent documentation of both dataset composition and model performance, especially when these systems are applied in real-world scenarios such as automated testing or user assistance. As the deployment of GUI automation technology expands, it is imperative to consider the impact on user autonomy and employment, ensuring that such tools are used in a manner that respects consent, fairness, and accountability.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024a. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024b. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Lin Ai, Zheng Hui, Zizhou Liu, and Julia Hirschberg. 2024. Enhancing pre-trained generative language models with question attended span extraction on machine reading comprehension. In *In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 10046–10063, Miami, Florida, USA. Association for Computational Linguistics.*

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. 2023. Fuyu-8b: A multimodal architecture for ai agents.

Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Bucker, Lawrence Jang, and Zack Hui. 2024. Windows agent arena: Evaluating multi-modal os agents at scale. *Preprint*, arXiv:2409.08264.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoor-

thi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *Preprint*, arXiv:2310.09478.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. 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.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pages 845–854.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2024. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Kanishk Gandhi, Jan-Philipp Fraenken, Tobias Gerstenberg, and Noah Goodman. 2023. Understanding social reasoning in language models with language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 13518–13529. Curran Associates, Inc.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2024. Navigating the digital world as humans do: Universal visual grounding for gui agents. *arXiv preprint arXiv:2410.05243*.

Zecheng He, Srinivas Sunkara, Xiaoxue Zang, Ying Xu, Lijuan Liu, Nevan Wichers, Gabriel Schubiner, Ruby Lee, and Jindong Chen. 2021. Actionbert: Leveraging user actions for semantic understanding of user interfaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5931–5938.

Zheng Hui, Zhaoxiao Guo, Hang Zhao, Juanyong Duan, Lin Ai, Yinheng Li, Julia Hirschberg, and Congrui Huang. 2024a. Can open-source llms enhance data augmentation for toxic detection?: An experimental study. *arXiv preprint arXiv:2411.15175*.

Zheng Hui, Zhaoxiao Guo, Hang Zhao, Juanyong Duan, and Congrui Huang. 2024b. ToxiCraft: A novel framework for synthetic generation of harmful information. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16632–16647, Miami, Florida, USA. Association for Computational Linguistics.

Zheng Hui, Yinheng Li, Tianyi Chen, Colby Banbury, Kazuhito Koishida, et al. 2025a. Winclick: Gui grounding with multimodal large language models. *arXiv preprint arXiv:2503.04730*.

Zheng Hui, Xiaokai Wei, Yexi Jiang, Kevin Gao, Chen Wang, Frank Ong, Se eun Yoon, Rachit Pareek, and Michelle Gong. 2025b. Matcha: Can multi-agent collaboration build a trustworthy conversational recommender? *Preprint*, arXiv:2504.20094.

Yue Jiang, Jichu Li, Yang Liu, Dingkang Yang, Feng Zhou, and Quyu Kong. 2025. Danmakutppbench: A multi-modal benchmark for temporal point process modeling and understanding. *arXiv preprint arXiv:2505.18411*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*.

Anurendra Kumar, Keval Morabia, William Wang, Kevin Chang, and Alex Schwing. 2022. CoVA: Context-aware visual attention for webpage information extraction. In *Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 80–90, Dublin, Ireland. Association for Computational Linguistics.

Hugo Laurençon, Léo Tronchon, and Victor Sanh. 2024. Unlocking the conversion of web screenshots into html code with the websight dataset. *Preprint*, arXiv:2403.09029.

Ao Li, Yuexiang Xie, Songze Li, Fugee Tsung, Bolin Ding, and Yaliang Li. 2025a. Agent-oriented planning in multi-agent systems. In *The Thirteenth International Conference on Learning Representations*.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025b. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*.

Chengzu Li, Caiqi Zhang, Han Zhou, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2024. TopViewRS: Vision-language models as top-view spatial reasoners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1786–1807, Miami, Florida, USA. Association for Computational Linguistics.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. 2022. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975.

Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. 2020. Mapping natural language instructions to mobile UI action sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8198–8210, Online. Association for Computational Linguistics.

Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. 2024. MMC: Advancing multimodal chart understanding with large-scale instruction tuning. 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 1287–1310, Mexico City, Mexico. Association for Computational Linguistics.

Yuhang Liu, Pengxiang Li, Zishu Wei, Congkai Xie, Xueyu Hu, Xinchen Xu, Shengyu Zhang, Xiaotian Han, Hongxia Yang, and Fei Wu. 2025. Infiguiagent: A multimodal generalist gui agent with native reasoning and reflection. *arXiv preprint arXiv:2501.04575*.

Kevin Moran, Carlos Bernal-Cárdenas, Michael Curcio, Richard Bonett, and Denys Poshyvanyk. 2018. Machine learning-based prototyping of graphical user interfaces for mobile apps. *IEEE Transactions on Software Engineering*, 46(2):196–221.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024a. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *arXiv preprint arXiv:2406.01014*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024b. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi

Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2024c. Cogvlm: Visual expert for pretrained language models. *Preprint*, arXiv:2311.03079.

Chen Henry Wu, Jing Yu Koh, Ruslan Salakhutdinov, Daniel Fried, and Aditi Raghunathan. 2024a. Adversarial attacks on multimodal agents. *arXiv e-prints*, pages arXiv–2406.

Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. 2023. Webui: A dataset for enhancing visual ui understanding with web semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

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

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

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023b. The dawn of lmms: Preliminary explorations with gpt-4v(ision). *Preprint*, arXiv:2309.17421.

Xiaofeng Zhang, Fanshuo Zeng, Yihao Quan, Zheng Hui, and Jiawei Yao. 2025. Enhancing multimodal large language models complex reason via similarity computation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 10203–10211.

Xiaoyi Zhang, Lilian De Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, et al. 2021. Screen recognition: Creating accessibility metadata for mobile applications from pixels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Yanzhe Zhang, Tao Yu, and Diyi Yang. 2024. Attacking vision-language computer agents via pop-ups. *arXiv preprint arXiv:2411.02391*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. Gpt-4v(ision) is a generalist web agent, if grounded. *Preprint*, arXiv:2401.01614.

A Training Details

To improve **WinClick**'s (Hui et al., 2025a) capacity for GUI understanding in Windows environments, we conducted extensive training using both **full fine-tuning** and **parameter-efficient tuning** via LoRA (Low-Rank Adaptation). This dual approach allowed us to explore trade-offs between performance and resource efficiency while maintaining compatibility with large-scale multimodal architectures.

For full fine-tuning, we updated the entire model, including the vision encoder, language decoder, and cross-modal attention layers. The visual encoder was initialized from a pre-trained ViT model and fine-tuned with an initial learning rate of 2×10^{-6}. The language backbone, based on Phi-3, was initialized with a learning rate of 5×10^{-6} and trained using a batch size of 32. A linear learning rate scheduler was applied with a warmup ratio of 0.03 to stabilize early training steps and mitigate gradient instability. The optimizer used was AdamW with weight decay set to 0.01.

For LoRA-based tuning, we injected low-rank matrices into the cross-modal attention layers and trained only these additional parameters, freezing the rest of the model. This method provided a significantly smaller memory footprint and faster training convergence while still yielding non-trivial performance improvements in grounding precision. LoRA ranks were set to 8 across all adapted modules, and dropout was applied with a probability of 0.1.

All experiments were conducted using 4 × NVIDIA H100 GPUs in a distributed setting using mixed-precision training (fp16). We used DeepSpeed and HuggingFace Accelerate to handle gradient accumulation, checkpointing, and parallelization. Training convergence was achieved within 5 epochs, with early stopping based on validation click accuracy on a held-out subset of WinSpot.

B More Training Data Construct Examples

To build a robust and diverse training corpus, our data pipeline aggregated screenshots from various sources, including real-world application states, software demos, and open benchmarks (e.g., CoVA (Kumar et al., 2022), WebSight (Laurençon et al., 2024)). After visual filtering and quality assessment using Phi3-Vision, selected images were passed through a multi-stage annotation pipeline. Figure 6 illustrates a sample of the training data. These examples span interaction types such as: Single-button confirmation dialogs (e.g., "Click OK to continue"), Multi-option menus (e.g., "Choose 'Save As' from the File menu"), Toolbar item selection (e.g., "Click the printer icon to print the document"), Search or input field interaction (e.g., "Type your query in the search box at the top right"). Each image contains between 5 interactable regions, and both the instruction and bounding box data were validated for semantic alignment by our human annotators (see Section C).

C Human Annotation

The annotation process follow similar settings as Hui et al. (2024b). For WinSpot involved a group of carefully selected annotators, all of whom were undergraduate, master's, or Ph.D. students, proficient in GUI operations and familiar with the Windows operating system. The annotation team consisted of individuals with diverse academic backgrounds, ensuring a broad understanding of GUI interactions across different applications. Each annotator was tasked with identifying and marking interactable regions within various Windows applications, focusing on elements such as buttons, icons, menus, and other clickable UI components. For the annotation process, annotators were provided with a set of Windows screenshots. These screenshots were then annotated using a custom tool that allowed them to create bounding boxes around each interactable element. Annotators were also required to provide corresponding descriptions of the elements, ensuring that both the visual and functional aspects of each UI component were documented. The entire annotation process was conducted in English to maintain consistency across all samples. To ensure data privacy, all screenshots were reviewed and post-processed to remove any personal information or sensitive content. The final dataset includes over 1,000 images and 5,000 instruction-click pairs, representing a comprehensive set of interactions across a variety of Windows applications.

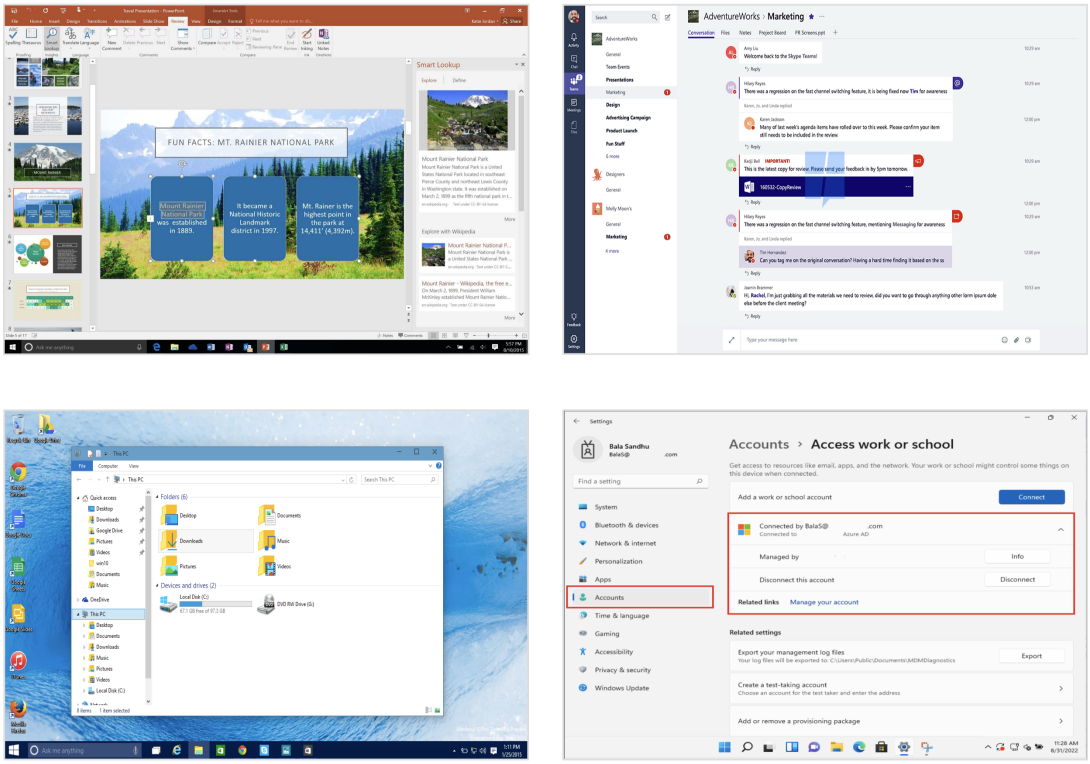

Figure 6: Examples of GUI grounding data generated during training set construction. Each box is annotated with the action-relevant region and its aligned instruction.