Spatially Stable GUI Grounding via Zoom Consistency Loss

Moon Ye-Bin [1] **Jiyeon Son** [2] **Tae-Hyun Oh** [2]

Abstract

GUI grounding, the task of localizing target UI elements from natural language instructions on a screenshot, is a core capability for GUI agents, yet remains challenging due to dense layouts and small elements in high-resolution interfaces. While inference-time zoom methods improve accuracy by re-running inference on cropped regions, they require multiple forward passes per grounding call, making them costly for multi-step agent deployment. Through controlled experiments, we find that models already possess sufficient visual understanding of target elements; what they lack is stable spatial focus under cluttered, high-resolution inputs, a problem we term spatial instability. To address this, we propose a Zoom Consistency Loss, a lightweight auxiliary training objective that enforces agreement between predictions on the original screenshot and on zoomed crops of the same image. At inference time, the model requires only a single forward pass with no additional overhead. Experiments across multiple benchmarks show consistent improvements, with particularly strong gains on the high-resolution ScreenSpot-Pro dataset (+3.80), demonstrating zoom consistency as an effective regularizer for spatially stable grounding.

1. Introduction

Autonomous agents that operate on graphical user interfaces (GUIs) directly from visual observations have emerged as a promising paradigm for task automation. Unlike API-based automation, GUI agents operate on raw screenshots, perceiving and acting on interfaces as humans do, making them applicable across any software environment. Given a natural language instruction, the agent must localize the target UI element on the current screenshot and execute the corresponding action. Since grounding is invoked at every action step, its accuracy directly determines whether the agent proceeds correctly or compounds errors across the task. Improving grounding is therefore not merely a perception sub-problem, but the primary bottleneck for reliable GUI agent deployment. This has motivated a line of work that trains specialized grounding models on large-scale GUI datasets via supervised fine-tuning and reinforcement learning (Xu et al., 2026; Wu et al., 2025b; Pei et al., 2026).

GUI screenshots, however, present unique challenges that natural images do not. A single screen may contain hundreds of icons, menus, and interactive widgets packed into a dense layout, many of which are visually similar and spatially close. This problem is especially severe in high-resolution environments, where target elements can be extremely small relative to the full screen. To address this, inference-time zoom methods have emerged (Tang et al., 2026; Liu et al., 2026; Jiang et al., 2025; Wu et al., 2025a), which crop a region around the initial prediction and re-run inference at higher resolution. While improving grounding accuracy, they introduce a fundamental scalability problem: each grounding call requires two or more forward passes. In a multi-step agent task spanning T action steps, this translates to at least $2T$ forward passes, making inference-time zoom methods costly for real-world deployment.

To understand the root cause of grounding failures, we conduct controlled experiments with spatial hints. We find that simply providing an enlarged ground-truth (GT) bounding box leads to large performance gains without any model modification. In addition, running inference on a zoomed crop of the GT region substantially boosts accuracy. These results suggest that the model already has sufficient visual understanding of the target, but lacks a stable spatial focus when facing a high-resolution full screenshot. The problem is therefore not one of knowledge, but of spatial instability.

Motivated by this, we propose *Zoom Consistency Loss*, a lightweight auxiliary objective that addresses spatial instability. The key idea is simple: a model's prediction on the original screenshot and its prediction on a zoomed crop of the same image should agree. Crucially, zooming is used only during training; at inference time, the model takes a single forward pass on the original screenshot with no additional overhead. Our contributions are as follows:

- We identify *spatial instability* as a core failure mode in

[1]POSTECH [2]KAIST. Correspondence to: Moon Ye-Bin <ybmoon@postech.ac.kr>.

Accepted to the 2nd Workshop on Compositional Learning at ICML 2026, Seoul, South Korea.

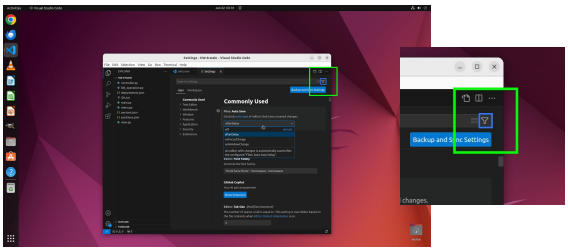

Figure 1. Example of spatial guidance. (Blue: GT bounding box, Green: +50px enlarged GT bounding box).

Table 1. Effect of spatial guidance (enlarged GT bounding box) on Qwen3-VL-8B grounding accuracy.

Method	ScreenSpot-Pro	OSWorld-G	MMBench-GUI-L2
Qwen3-VL-8B	53.51	59.22	82.05
+20px enlarged	**70.21** (+16.71)	**64.01** (+4.79)	**89.34** (+7.29)
+50px enlarged	61.99 (+8.48)	59.22 (+0.00)	87.81 (+5.76)
+100px enlarged	57.24 (+3.73)	59.40 (+0.18)	84.31 (+2.26)

GUI grounding, supported by controlled experiments with spatial hints and zoom-based analysis.

- We propose *Zoom Consistency Loss*, a simple auxiliary loss that encourages stable spatial grounding without any inference-time overhead.
- We show consistent improvements across multiple benchmarks, with particularly strong gains on the high-resolution dataset, highlighting zoom consistency as an effective regularizer for high-resolution GUI grounding.

2. Motivation

Study 1: Spatial Guidance. To examine how sensitive grounding models are to spatial context, we provide an enlarged ground-truth bounding box as a spatial hint at inference time, without modifying the model in any way. Specifically, we draw the GT bounding box in green on the original screenshot and expand it by a fixed margin in each direction (+20px, +50px, +100px), keeping the image resolution unchanged, as shown in Fig. 1. We test Qwen3-VL-8B (Bai et al., 2025) on three benchmarks: OSWorld-G (Xie et al., 2025), ScreenSpot-Pro (Li et al., 2025), and MMBench-GUI-L2 (Wang et al., 2025).

In Table 1, providing spatial guidance consistently improves performance across all benchmarks, with the largest gains observed at the smallest margin (+20px), including a notable improvement +16.71% on ScreenSpot-Pro. As the margin decreases, the performance gain increases, suggesting that tighter spatial hints help the model focus on the relevant region by reducing distracting elements in the surrounding context. Importantly, since no model modification is involved, these results indicate that the model already possesses sufficient visual understanding to localize the target element. What it lacks is stable spatial focus when processing the full screenshot without any guidance.

Table 2. Effect of oracle zoom (crop ratio) on Qwen3-VL-8B grounding accuracy. Crop ratios $r \in \{0.8, 0.7, \ldots, 0.2\}$ correspond to zoom factors of $1.25\times$ to $5.00\times$.

Crop Ratio	ScreenSpot-Pro	OSWorld-G	MMBench-GUI-L2
$r = 1.0$ (w/o crop)	53.51	59.22	82.05
$r = 0.8$	56.55 (+3.04)	61.88 (+2.66)	84.59 (+2.54)
$r = 0.7$	59.96 (+6.45)	63.48 (+4.26)	85.34 (+3.29)
$r = 0.6$	63.00 (+9.49)	64.01 (+4.79)	86.03 (+3.98)
$r = 0.5$	65.59 (+12.08)	65.07 (+5.85)	86.34 (+4.29)
$r = 0.4$	68.75 (+15.24)	64.54 (+5.32)	87.06 (+5.01)
$r = 0.3$	71.79 (+18.28)	67.02 (+7.80)	86.81 (+4.76)
$r = 0.2$	**75.21 (+21.70)**	**69.33 (+10.11)**	**87.31 (+5.26)**

Study 2: Zoomed Inference. To further investigate whether models can correctly localize targets when given zoomed visual context, we crop a fixed-size window around the GT region and resize it back to the original resolution before passing it to the model. Specifically, given a crop ratio $r \in (0, 1]$, we define a window of size $(r \cdot W, r \cdot H)$ where W and H are the width and height of the original screenshot. The window is positioned via translation to ensure the GT bounding box is fully contained within it, and the cropped region is then resized back to (W, H).

In Table 2, this oracle zoom consistently improves performance across all benchmarks, with gains increasing as the crop ratio decreases. Together with Study 1, these results confirm that grounding failures are not due to a lack of visual understanding, but rather an inability to maintain stable spatial focus on cluttered, high-resolution full screenshots.

3. Method: Zoom Consistency Loss

Overview. Given a training sample with a screenshot I of size $W \times H$ and a GT bounding box (x_1, y_1, x_2, y_2), we fine-tune the model with a combination of a standard task loss and our proposed zoom consistency loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{cons}}, \quad \lambda = 0.01 \tag{1}$$

where $\mathcal{L}_{\text{task}}$ is the standard cross-entropy loss over valid tokens, and $\mathcal{L}_{\text{cons}}$ encourages consistent coordinate predictions across different spatial views of the same image.

Zoom View Generation. For each training sample, we generate a zoomed view by cropping a region centered around the GT bounding box and resizing it back to the original resolution (W, H). The crop window is initialized to approximately $4\times$ the bounding box size, then adjusted to preserve the original aspect ratio $r = W/H$, and clamped to the image boundary. The crop is then scaled by a random factor $u \sim \mathcal{U}(1.0, 1.5)$, yielding a zoomed image of size (W_z, H_z) where the target element appears larger and with fewer distracting surroundings.

Pseudo Target Generation. We perform a forward pass on

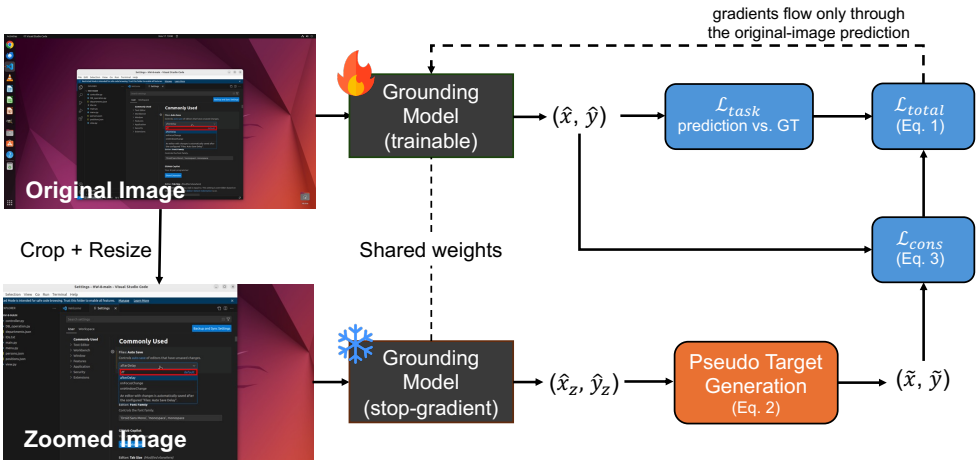

Figure 2. **Overview of Zoom Consistency Training.** For each training sample, we generate a zoomed image by cropping around the GT bounding box. Both original and zoomed images are passed through the same grounding model (shared weights), but the zoomed branch uses stop-gradient. The zoomed prediction $(\hat{x}_z, \hat{y}_z)$ is projected back to the original coordinate space (Eq. 2) to form a pseudo target $(\tilde{x}, \tilde{y})$. The consistency loss $\mathcal{L}_{\text{cons}}$ encourages the original-image prediction $(\hat{x}, \hat{y})$ to match the pseudo target, while gradients flow only through the original branch. At inference, only the original imagee branch is used, requiring a single forward pass.

the zoomed image with stop-gradient to obtain a predicted coordinate $(\hat{x}_z, \hat{y}_z)$. Since Qwen3-VL outputs coordinates in a normalized $(0, 1000)$ space (Bai et al., 2025), we project this prediction back to the original image coordinate space, also in the $(0, 1000)$ scale, via:

$$\tilde{x} = \text{clip}\left(1000 \cdot \frac{x_{\text{crop}} + \frac{\hat{x}_z}{1000} \cdot c_w}{W},\ 0,\ 1000\right) \tag{2}$$

and similarly for $\tilde{y}$, where $(x_{\text{crop}}, y_{\text{crop}})$ is the top-left corner of the crop window and c_w, c_h are the crop dimensions. The resulting $(\tilde{x}, \tilde{y})$ serves as a pseudo target in the original image coordinate space. For grounding models that directly output coordinates in absolute pixel space without such normalization, the factor of 1000 can be omitted, and the projection reduces to a simple affine mapping between the crop and the original image, $\tilde{x} = x_{\text{crop}} + \hat{x}_z \cdot (c_w / W_z)$ (analogously for $\tilde{y}$).

Zoom Consistency Loss. The consistency loss enforces that the model's coordinate prediction on the original screenshot agrees with the pseudo target derived from the zoomed view. We extract the digit token positions from the original image prediction and compute cross-entropy against the pseudo target token ids:

$$\mathcal{L}_{\text{cons}} = \frac{1}{B'} \sum_{i=1}^{B'} \frac{1}{n_i} \sum_{k=1}^{n_i} \text{CE}\left(\mathbf{z}_{i,p_{i,k}},\ t_{i,k}\right) \tag{3}$$

where B' is the number of valid samples in the batch, n_i is the number of digit tokens in sample i, $p_{i,k}$ is the k-th digit token position (tail-aligned), $\mathbf{z}_{i,p_{i,k}}$ is the corresponding logit vector, and $t_{i,k}$ is the pseudo target token id. Gradients flow only through the original image prediction; the zoomed view serves purely as a teacher signal. If no valid samples exist in a batch, $\mathcal{L}_{\text{cons}}$ is skipped and $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}}$.

4. Experiment

4.1. Setup

Implementation Details. We use Qwen3-VL-8B (Bai et al., 2025) as our base model, fine-tuned with LoRA (rank=8, alpha=32, dropout=0.05, target modules: all-linear). Training is conducted for 1 epoch with a batch size of 8 (per-device batch size 1, gradient accumulation steps 8), a learning rate of 2e-5 with a cosine scheduler, weight decay of 0.1, and a maximum sequence length of 2048. All experiments are run on 4 NVIDIA RTX A6000 GPUs.

Training Data. We train on the Salesforce grounding dataset (Yang et al., 2025a) combining 70.7K samples from five GUI interaction datasets: Aria-UI (Yang et al., 2025b), OmniAct (Kapoor et al., 2024), Widget Captioning (Li et al., 2020), UI-Vision (Nayak et al., 2025), and OS-Atlas (Wu et al., 2025b).

Evaluation Benchmarks. We evaluate on three GUI grounding benchmarks with varying resolutions and interface types. ScreenSpot-Pro (Li et al., 2025) contains 1,581 high-resolution professional desktop screenshots with an average resolution of 3267 × 1727 pixels, reaching up to 6016 × 3384 at the 95th percentile. OSWorld-G (Xie et al., 2025) contains 564 desktop screenshots with an average resolution of 1728 × 973 pixels. MMBench-GUI-L2 (Wang et al., 2025) contains 3,594 samples spanning multiple platforms including mobile, desktop, and web, with an average

resolution of 1856×1736 pixels.

4.2. Results

We compare our method against a wide range of existing GUI grounding models on three benchmarks. We report the Qwen3-VL-8B baseline and three fine-tuned variants: standard supervised fine-tuning ($\mathcal{L}_{\text{task}}$), our proposed consistency loss with a fixed crop scale ($\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{cons}}$ (fixed)), and with a randomly sampled crop scale ($\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{cons}}$ (random)).

Strong gains on high-resolution screenshots. The most pronounced gains appear on ScreenSpot-Pro (Table 3), the highest-resolution benchmark in our evaluation. Our consistency loss with random crop scaling improves the Qwen3-VL-8B baseline from 53.51 to 57.31 (+3.80), substantially outperforming standard fine-tuning alone ($\mathcal{L}_{\text{task}}$: +2.91). Compared to existing methods, our 8B model becomes competitive with much larger or specialized models such as Qwen2.5-VL-72B (53.3) and GUI-Owl-7B (54.9), narrowing the gap to GUI-Owl-32B (58.0) despite using a model an order of magnitude smaller. This aligns with our motivation analysis: since spatial instability is most severe under cluttered and high-resolution inputs, training-time zoom consistency provides the largest benefit.

Consistent improvements across benchmarks. On OSWorld-G (Table 4), our method achieves 60.82, surpassing both the strongest baseline GUI-Owl-32B (58.0) and the SFT-only variant (59.93). On MMBench-GUI-L2 (Table 4), the consistency loss further pushes performance to 83.56 (+1.51 over baseline), again exceeding GUI-Owl-32B (82.97). Across all three benchmarks, adding $\mathcal{L}_{\text{cons}}$ on top of $\mathcal{L}_{\text{task}}$ yields consistent improvements over SFT alone, indicating that the gain is not merely an artifact of additional supervision but a genuine effect of the consistency objective.

Fixed vs. random crop scale. We compare two design choices for the zoomed view: a fixed crop scale and a randomly sampled scale $u \sim \mathcal{U}(1.0, 1.5)$. The random variant performs best on ScreenSpot-Pro (57.31 vs. 56.93) and OSWorld-G (60.82 vs. 60.28), which contain a larger proportion of small targets in cluttered layouts. The fixed variant slightly edges out on MMBench-GUI-L2 (83.56 vs. 83.22), where the resolution is lower and target sizes are more uniform. Overall, random scaling tends to help more on the higher-resolution benchmarks, suggesting that exposing the model to a diverse range of zoom levels during training is beneficial when test-time spatial scales vary widely.

Summary. The results show that Zoom Consistency Loss is an effective and lightweight regularizer for GUI grounding. It consistently improves performance with the largest gains on high-resolution benchmarks where spatial instability is most pronounced. All without any inference-time overhead.

Table 3. Performance comparison on ScreenSpot-Pro.

Method	**ScreenSpot-Pro**
Existing Methods	
GPT-4o	0.8
Qwen2.5-VL-3B	25.9
Qwen2.5-VL-7B	27.6
UI-TARS-2B	27.7
UI-TARS-7B	35.7
UI-TARS-72B	38.1
JEDI-7B	39.5
GUI-G2-7B	47.5
Qwen2.5-VL-32B	47.6
Qwen2.5-VL-72B	53.3
GUI-Owl-7B	54.9
GUI-Owl-32B	58.0
Ours	
Baseline (Qwen3-VL-8B)	53.51
+ $\mathcal{L}_{\text{task}}$	56.42
+ $\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{cons}}$ (fixed)	56.93 (+3.42)
+ $\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{cons}}$ (random)	**57.31 (+3.80)**

Table 4. Performance on OSWorld-G and MMBench-GUI-L2.

Method	**OSWorld-G**	**MMBench-GUI-L2**
Existing Methods		
Qwen2.5-VL-7B	31.4	33.85
UGround-V1-7B	36.4	65.68
GUI-Owl-7B	55.9	80.49
GUI-Owl-32B	58.0	82.97
Ours		
Baseline (Qwen3-VL-8B)	59.22	82.05
+ $\mathcal{L}_{\text{task}}$	59.93	83.19
+ $\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{cons}}$ (fixed)	60.28 (+1.06)	**83.56 (+1.51)**
+ $\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{cons}}$ (random)	**60.82 (+1.60)**	83.22 (+1.17)

5. Conclusion

We identified spatial instability as a core failure mode in GUI grounding, where models fail to maintain stable spatial focus on high-resolution screenshots despite possessing sufficient visual understanding of the target. To address this, we proposed Zoom Consistency Loss, a lightweight auxiliary training objective that enforces prediction agreement between the original screenshot and randomly zoomed crops. Our method yields consistent improvements across multiple benchmarks with no inference-time overhead, with particularly strong gains on the high-resolution ScreenSpot-Pro dataset. Despite these promising results, our work has limitations. First, the optimal crop ratio range remains unexplored; we use a fixed random sampling range without systematic analysis of how different ranges affect training stability and performance. Second, our experiments are conducted solely on Qwen3-VL-8B, and it remains unclear whether the proposed method generalizes to other model sizes and architectures. We leave these as directions for future work.

Acknowledgement

This work was supported by Samsung Electronics Co., Ltd (Project Code: IO260114-15161-01), the InnoCORE program of the Ministry of Science and ICT (N10250156, KAIST InnoCore LLM) (50%), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00457882, National AI Research Lab Project) (50%).

References

Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Jiang, Z., Xie, S., Li, W., Zu, W., Li, P., Qiu, J., Pei, S., Ma, L., Huang, T., Wang, M., et al. Zoom in, click out: Unlocking and evaluating the potential of zooming for gui grounding. *arXiv preprint arXiv:2512.05941*, 2025.

Kapoor, R., Butala, Y. P., Russak, M., Koh, J. Y., Kamble, K., AlShikh, W., and Salakhutdinov, R. Omniact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In *European Conference on Computer Vision (ECCV)*, 2024.

Li, K., Meng, Z., Lin, H., Luo, Z., Tian, Y., Ma, J., Huang, Z., and Chua, T.-S. Screenspot-pro: Gui grounding for professional high-resolution computer use. In *Proceedings of the ACM International Conference on Multimedia*, 2025.

Li, Y., Li, G., He, L., Zheng, J., Li, H., and Guan, Z. Widget captioning: Generating natural language description for mobile user interface elements. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, 2020.

Liu, Z., Feng, T., Kang, B., Yang, Y., and Luo, J. Zoom to essence: Trainless gui grounding by inferring upon interface elements. *arXiv preprint arXiv:2603.14448*, 2026.

Nayak, S., Jian, X., Lin, K. Q., Rodriguez, J. A., Kalsi, M., Chapados, N., Özsu, M. T., Agrawal, A., Vazquez, D., Pal, C., Taslakian, P., Gella, S., and Rajeswar, S. UI-Vision: A desktop-centric GUI benchmark for visual perception and interaction. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Pei, S., Tang, L., Duan, T., Chen, L., Li, S., Huang, K., Jing, Y., Yan, Y., Zhang, B., Jiang, C., et al. Adazoom-gui: Adaptive zoom-based gui grounding with instruction refinement. *arXiv preprint arXiv:2603.17441*, 2026.

Tang, F., Chen, B., Lu, Z., Chen, T., Nong, S., Jiang, T., Xu, W., Lu, W., Xiao, J., Zhuang, Y., et al. Ui-zoomer: Uncertainty-driven adaptive zoom-in for gui grounding. *arXiv preprint arXiv:2604.14113*, 2026.

Wang, X., Wu, Z., Xie, J., Ding, Z., Yang, B., Li, Z., Liu, Z., Li, Q., Dong, X., Chen, Z., et al. Mmbench-gui: Hierarchical multi-platform evaluation framework for gui agents. *arXiv preprint arXiv:2507.19478*, 2025.

Wu, H., Chen, H., Cai, Y., Liu, C., Ye, Q., Yang, M.-H., and Wang, Y. Dimo-gui: Advancing test-time scaling in gui grounding via modality-aware visual reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025a.

Wu, Z., Wu, Z., Xu, F., Wang, Y., Sun, Q., Jia, C., Cheng, K., Ding, Z., Chen, L., Liang, P. P., and Qiao, Y. OS-ATLAS: A foundation action model for generalist GUI agents. In *International Conference on Learning Representations (ICLR)*, 2025b.

Xie, T., Deng, J., Li, X., Yang, J., Wu, H., Chen, J., Hu, W., Wang, X., Xu, Y., Wang, Z., et al. Scaling computer-use grounding via user interface decomposition and synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Xu, H., Zhang, X., Liu, H., Wang, J., Zhu, Z., Zhou, S., Hu, X., Gao, F., Cao, J., Wang, Z., et al. Mobile-agent-v3. 5: Multi-platform fundamental gui agents. *arXiv preprint arXiv:2602.16855*, 2026.

Yang, Y., Li, D., Dai, Y., Yang, Y., Luo, Z., Zhao, Z., Hu, Z., Huang, J., Saha, A., Chen, Z., Xu, R., Pan, L., Savarese, S., Xiong, C., and Li, J. Gta1: Gui test-time scaling agent. 2025a. URL https://arxiv.org/abs/2507.05791.

Yang, Y., Wang, Y., Li, D., Luo, Z., Chen, B., Huang, C., and Li, J. Aria-ui: Visual grounding for gui instructions. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025b.