Preprints.org

Review

Not peer-reviewed version

GUI Grounding for Computer-Use Agents: A Survey

Lizheng Zu [†], Ke Huang [†], Gang Xu [†], Jingfan Fan, Baoyuan Wu, Shengfeng He, Hehe Fan, Xiangyu Xu, Chun Li, Yanshan Li, Shihui Ying, Junchi Yan, Bingyi Jing, Ming Li [*], Zheng Zhu

Posted Date: 4 November 2026

Keywords: computer-use agents; GUI agents; GUI grounding; multimodal large language models; survey

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Review

GUI Grounding for Computer-Use Agents: A Survey

Lizheng Zu [1,†], Ke Huang [2,†], Gang Xu [3,†], Jingfan Fan [4], Baoyuan Wu [5], Shengfeng He [6], Hehe Fan [7], Xiangyu Xu [8], Chun Li [9], Yanshan Li [1], Shihui Ying [10], Junchi Yan [11], Bingyi Jing [5], Ming Li [5,*] and Zheng Zhu [12]

[1] Shenzhen University, Shenzhen, China
[2] Sun Yat-sen University, Shenzhen, China
[3] Guangming Laboratory, Shenzhen, China
[4] Beijing Institute of Technology, Beijing, China
[5] The Chinese University of Hong Kong, Shenzhen, China
[6] Singapore Management University, Singapore
[7] Zhejiang University, Hangzhou, China
[8] Xi'an Jiaotong University, Xi'an, China
[9] Shenzhen MSU-BIT University, Shenzhen, China
[10] Shanghai University, Shanghai, China
[11] Shanghai Jiaotong University, Shanghai, China
[12] GigaAI, Beijing, China
* Correspondence: minglisai@cuhk.edu.cn
† Lizheng Zu, Ke Huang, and Gang Xu contributed equally to this work.

We maintain a regularly updated repository for this survey at https://github.com/OmniAI-Lab-CUHKSZ/Awesome-GUI-Grounding.

Abstract

Graphical user interface (GUI) agents automate tasks in digital environments by interpreting user goals and executing interface operations. GUI grounding serves as a critical link between high-level reasoning and low-level interaction execution in GUI agents. It maps users' operation intents to specific interaction targets and their spatial parameters, enabling agents to translate high-level decisions into executable interface actions. Advances in multimodal large language models (MLLMs) have substantially improved GUI agents' capabilities in autonomous operation. However, existing surveys primarily focus on complete GUI-agent systems, leaving grounding methods, data resources, and performance limitations insufficiently synthesized. To the best of our knowledge, this is the first comprehensive survey dedicated to GUI grounding. First, we establish a unified taxonomy along three complementary dimensions–learning paradigms, inference strategies, and target representations–to systematically characterize the core mechanisms and properties of existing methods. Second, we review training datasets, evaluation benchmarks, and metrics, tracing the evolution of data resources and evaluation tasks. Third, we conduct a comparative analysis of representative methods across evaluation benchmarks, identifying their strengths and limitations across target types, instruction complexity, and interaction requirements, while examining trade-offs between accuracy and inference cost. Our analysis reveals that GUI grounding research is expanding from explicit target localization in static interfaces toward implicit intent understanding, target resolution in dynamic interfaces, and spatial-parameter prediction for complex interactions. We discuss the resulting open challenges and future research directions.

Keywords: computer-use agents; GUI agents; GUI grounding; multimodal large language models; survey

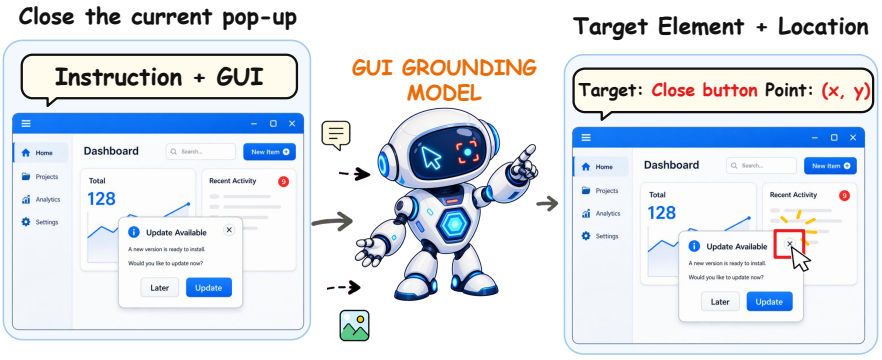
1. Introduction

Graphical user interfaces (GUIs) are the primary medium through which users access digital services and interact with software systems. Recent advances in large language models (LLMs) and

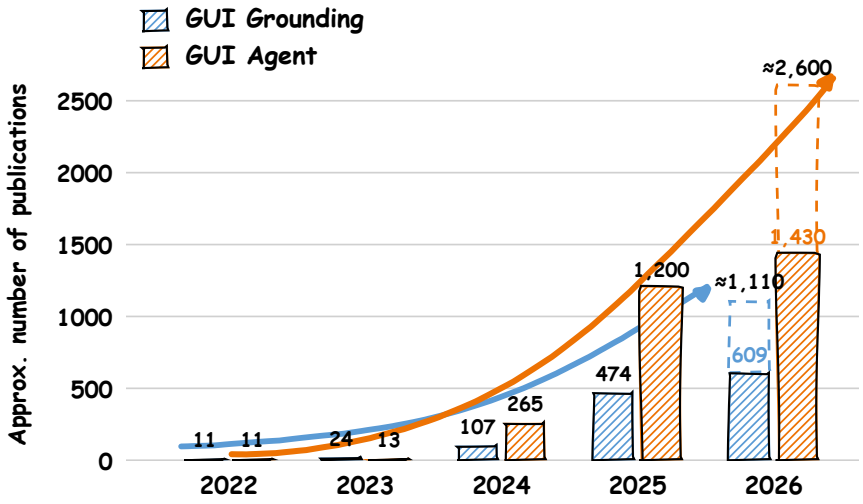
© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

multimodal large language models (MLLMs) have accelerated the shift in GUI automation from rule-based scripts and predefined workflows toward GUI agents [1,2] that understand natural-language goals, perceive interface states, and autonomously execute actions [3]. Regardless of whether an agent relies on explicit planning, hierarchical decision-making, or end-to-end action generation, spatial interactions such as clicking and selecting require a common capability: mapping the current operation intent to a concrete interface target and determining where to act. We refer to this capability as *GUI grounding*. Specifically, given a GUI screenshot and a natural-language query, a model typically needs to predict the click coordinate [4] or bounding box [5] of the target interface element, as illustrated in Figure 1(a).



(a)



(b)

Figure 1. Overview of GUI grounding and related research trends. (a) An illustrative GUI-grounding example, where a model identifies the target interface element from a GUI screenshot based on a natural-language instruction and predicts its interaction location. (b) Annual publication counts related to GUI agents and GUI grounding, showing the growing research interest in these areas.

Within a GUI agent, GUI grounding serves as a critical bridge between high-level task understanding and low-level interface operations. This capability is fundamental across web [6,7], mobile [8,9], desktop [10], and cross-application GUI environments [11]. A grounding error not only causes an immediate incorrect action but may also alter subsequent interface states, allowing the error to propagate through multi-step interactions and ultimately cause task failure [12]. This challenge is compounded by modern GUIs, which often contain numerous small, densely packed text elements, icons, and controls [13] with highly similar appearances [5]. Reliable GUI grounding therefore requires integrating visual, semantic, structural, and layout cues [14] with fine-grained spatial discrimination [15].

As shown in Figure 1(b), GUI grounding research has also increased substantially in recent years alongside the rapid growth of GUI-agent research, gradually developing into a relatively independent

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

research direction [16,17]. Figure 2 traces the development of GUI grounding from early grounding foundations, such as visual grounding, through interface modeling and screenshot-based grounding, to recent directions aimed at refining grounding capabilities. Research objectives are no longer limited to directly predicting target locations, but extend to inference-time search strategies [18], iterative refinement methods [19], and uncertainty-aware prediction mechanisms [20]. These developments have diversified how grounding capabilities are acquired, how predictions are generated and refined, and how interaction targets are represented, motivating a unified framework for systematically organizing and comparing existing methods.

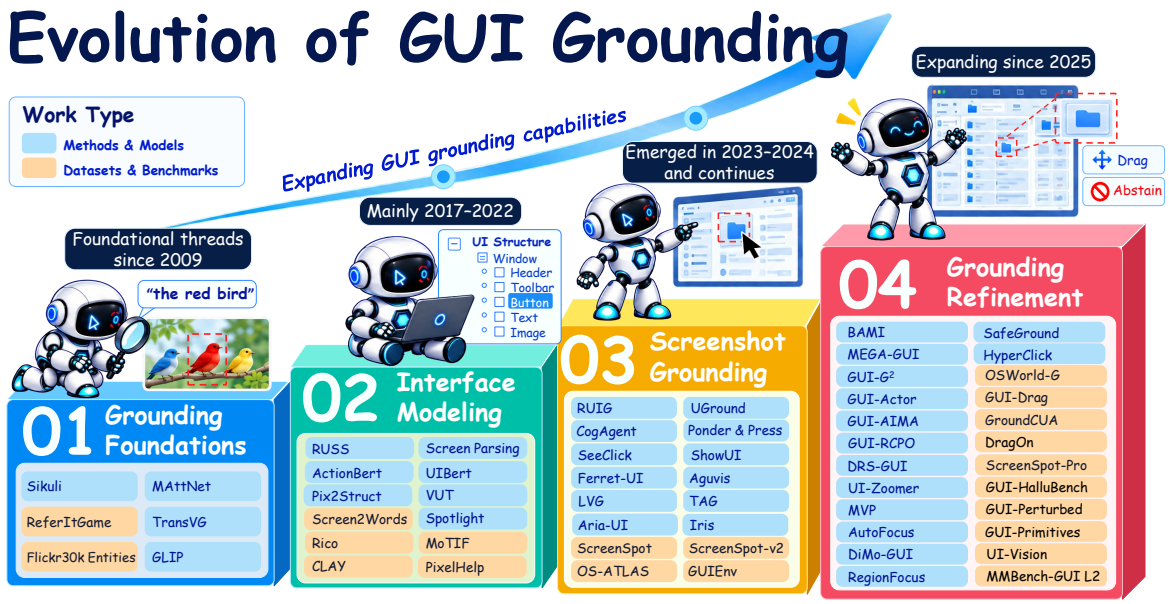


Figure 2. Evolution of GUI grounding. A historical overview of the development of GUI grounding methods, including the progression of learning paradigms, datasets, and evaluation benchmarks.

The development of datasets and benchmarks further reveals both the progress and remaining limitations of GUI grounding. Training resources have evolved from early mobile interface collections to large-scale cross-platform datasets [21], supporting representation learning across increasingly diverse interfaces. Although leading models approach saturation on ScreenSpot-v2 [22], substantial performance gaps persist under more demanding visual and semantic conditions. For example, ScreenSpot-Pro [13] challenges models to distinguish small targets and visually similar controls in dense professional interfaces, while UI-Vision [14] evaluates the ability to identify targets through their functions and spatial relationships. These settings require fine-grained visual discrimination and contextual interpretation beyond matching explicit textual or visual cues. Recent benchmarks further extend evaluation to precise drag operations [23] and the rejection of instructions referring to nonexistent targets [24]. Together, these findings indicate that progress on conventional localization benchmarks does not fully characterize grounding capabilities across diverse interfaces, instructions, and interaction requirements.

However, existing surveys primarily examine GUI agents as complete systems, either reviewing their overall architectures and capabilities, including perception, planning, memory, and action execution [1,2], or examining specific aspects such as trustworthiness [25]. In these surveys, GUI grounding is generally treated as one component of the broader agent system, rather than examined independently as a research problem. Meanwhile, surveys on visual grounding [26,27] review language-conditioned localization in general visual scenes, without systematically addressing the methods and interaction requirements specific to GUI grounding. Table 1 summarizes the differences in scope and coverage between these surveys and ours. To the best of our knowledge, this is the first comprehensive survey

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

dedicated to GUI grounding, systematically reviewing its task formulations, methodological advances, training resources, and evaluation benchmarks, and synthesizing the capabilities and limitations of existing approaches.

Table 1. Comparison of representative surveys. ✓: structured categorization or comparison; △: substantive discussion without such synthesis; –: absent or passing mention. REC: referring expression comprehension; Perf.: performance.

Survey	Scope	Method analysis			Data & evaluation		
		Learn. parad.	Infer. strategy	Target repr.	Train. data	Eval. bench.	Perf. analysis
Qiao et al. [26]	REC	–	–	–	–	–	–
Xiao et al. [27]	Visual grounding	–	–	–	–	–	–
Wang et al. [1]	GUI agents	△	△	–	△	–	–
Zhang et al. [2]	GUI agents	△	△	✓	✓	✓	–
Nguyen et al. [28]	GUI agents	✓	△	△	△	△	–
Tang et al. [29]	GUI agents	△	✓	△	△	△	–
Hu et al. [30]	OS agents	✓	△	△	✓	△	–
Sager et al. [31]	Computer use	△	✓	✓	△	△	–
Shi et al. [25]	Trustworthy GUI	–	△	–	–	△	–
Chen et al. [32]	GUI benchmarks	△	△	△	△	✓	✓
Li & Huang [33]	RL-based GUI	✓	△	–	△	–	–
This survey	**GUI grounding**	✓	✓	✓	✓	✓	✓

We present a systematic survey of GUI grounding, covering its task formulations, methodological advances, training resources, and evaluation frameworks. We organize existing methods along three complementary dimensions that capture how grounding capability is acquired, how grounding is performed , and how interaction targets are represented: (1) *Learning paradigms* characterize how grounding capabilities are acquired, covering GUI-grounding pretraining, fine-tuning, and training-free approaches; (2) *Inference strategies* characterize how grounding predictions are produced and refined at inference time, including direct, guided, multi-hypothesis, and iterative grounding; (3) *Target representations* characterize how interaction targets are encoded and converted into executable spatial parameters, including coordinate, spatial-response, and element representations. Building on this taxonomy, we further examine complex interactions, cross-task generalization, and trustworthy grounding to identify the capability boundaries of existing methods and the challenges that remain.

This survey focuses on GUI grounding research publicly available on or before October 31, 2026. Broader GUI-agent studies are included only when they explicitly investigate grounding capabilities. We also maintain a regularly updated github repository to track developments beyond the survey's cutoff date. Our main contributions are summarized as follows:

- To the best of our knowledge, this is the first systematic survey dedicated to GUI grounding. We establish a unified task definition and research scope, and clarify its distinctions from and connections to conventional visual grounding and complete GUI-agent research.
- We establish a unified and decomposable taxonomy of GUI-grounding methods along learning paradigms, inference strategies, and target representations, enabling systematic comparison of their commonalities, differences, combinations, and applicable conditions.
- We systematically review training resources and evaluation benchmarks, compare representative methods, and analyze their capability limitations and accuracy–efficiency trade-offs.
- Based on our methodological and evaluation analyses, we identify the key challenges facing GUI grounding and highlight promising directions for future research.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

The remainder of this paper is organized as follows. Section 2 formalizes GUI grounding and introduces its observation and target representations. Section 3 reviews existing methods through a taxonomy of learning paradigms, inference strategies, and target representations. Section 4 reviews training resources, benchmarks, and evaluation metrics, and analyzes the performance and limitations of representative methods. Section 5 discusses open challenges and future research directions in semantic, interactive, reliable, and efficient GUI grounding. Finally, Section 6 concludes the survey.

2. Background

This section introduces the basic concepts and task formulation of GUI grounding, describes its observation and input representations, and defines the target space and validity conditions.

2.1. Concepts and Problem Formulation

Visual grounding. Visual grounding (VG), commonly known as referring expression comprehension (REC), aims to locate a specific object or region in an image based on a natural-language description [26]. Given an image I and a textual query q, a model identifies the referred target by interpreting its appearance, attributes, and relationships with other objects, and predicts its bounding box. Classical visual grounding assumes that the description refers to a single object present in the image. Generalized visual grounding relaxes this assumption to accommodate descriptions referring to multiple objects or no object [27].

GUI agents. GUI agents autonomously complete digital tasks by perceiving graphical user interfaces, understanding user goals, and executing interface operations [28]. In a partially observable interactive environment, a policy π_θ with parameters θ produces an action $a_t \sim \pi_\theta(\cdot \mid u, o_t, h_t)$ given a high-level task u, the current observation o_t, and the interaction history h_t. Here, t indexes interaction steps, and h_t contains preceding observations and actions. The action a_t typically specifies its type and arguments, such as a click location, drag endpoints, scroll direction, or text input. A complete GUI agent combines task understanding, state perception, long-horizon planning, action selection, and execution.

Conventional GUI grounding. Within a GUI agent, GUI grounding focuses on identifying interface targets and determining their spatial action arguments. A local operation intent q_t specifies the goal of a single interface operation, such as opening a settings menu, whereas the high-level task u may require multiple operations. Existing studies commonly formulate GUI grounding as language-conditioned interface-element localization. Given a GUI observation o and a query q describing a target element, a model predicts its location:

$$\hat{y} = g_\theta^{\text{conv}}(o, q), \qquad \hat{y} \in \mathcal{L}(o). \tag{1}$$

Here, g_θ^{conv} is a grounder parameterized by θ. The observation o may be a screenshot or include structured interface data such as the DOM or an accessibility tree. $\mathcal{L}(o)$ is the target space, typically comprising click coordinates, bounding boxes, or element identifiers. In the screenshot-only setting ($o = I$), $\hat{y}$ is a click coordinate or target bounding box. This formulation generally assumes the target is present in o and the action type is given or fixed by the evaluation protocol (e.g., clicking) [4,5,16,34].

Generalized GUI grounding. Conventional GUI grounding covers many click-based tasks. However, user instructions do not always identify the target explicitly; they may instead refer to its intended function or spatial relations [14], or require interaction context to resolve the target [34]. Moreover, GUI interactions are not limited to clicking, and some operations require multiple spatial parameters. Drag-based interactions, for example, require joint localization of start and end positions, as in text selection [23] and spreadsheet range selection [35]. Multiple interface elements may also satisfy the same user intent, so grounding should accommodate alternative valid targets rather than assume a unique correct location [17]. Accordingly, we define generalized GUI grounding as action-type-conditioned interface-target resolution and spatial-argument determination. Given the current interface observation o_t, a local operation intent q_t, an action type τ_t, and an optional interaction history h_t, a GUI grounder predicts a response:

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

$$\hat{r}_t = g_\theta(o_t, q_t, h_t; \tau_t),$$
$$\hat{r}_t \in \mathcal{Y}_{\tau_t}(o_t) \cup \mathcal{Z}. \tag{2}$$

Here, g_θ denotes the complete grounder, $\hat{r}_t$ its final response, and $\hat{y}_t$ a candidate target prediction that may be withheld. The action type and chosen representation determine the output space $\mathcal{Y}_{\tau_t}(o_t)$. It may contain interface elements or structured nodes, points or regions, or target-bound spatial arguments. We define $\mathcal{Z} = \{\text{NoValidTarget}, \text{Abstain}\}$ to represent two types of non-execution responses. NoValidTarget reports that no target satisfies the instruction in the current interface state, whereas Abstain withholds a prediction due to insufficient confidence.

2.2. Observation and Input Representations

Observations for GUI grounding may contain visual information, structured information, or both. We write $o_t = (\mathcal{I}_t, \mathcal{S}_t)$, where $\mathcal{I}_t = \{I_t^{(v)}\}_{v=1}^{V_t}$ contains the V_t rendered views available at time t, and $\mathcal{S}_t$ contains structured or symbolic information about the current interface state. Unavailable components are represented by the empty set.

Visual observations. Visual observations typically capture the rendered GUI as full screenshots, current viewports, or local regions. Screenshots preserve interface text, icons, controls, layout, and visual state, providing a common input for grounding across web, mobile, and desktop interfaces [4,13,16]. Figure 3 shows representative GUI screenshots across platforms and application domains.

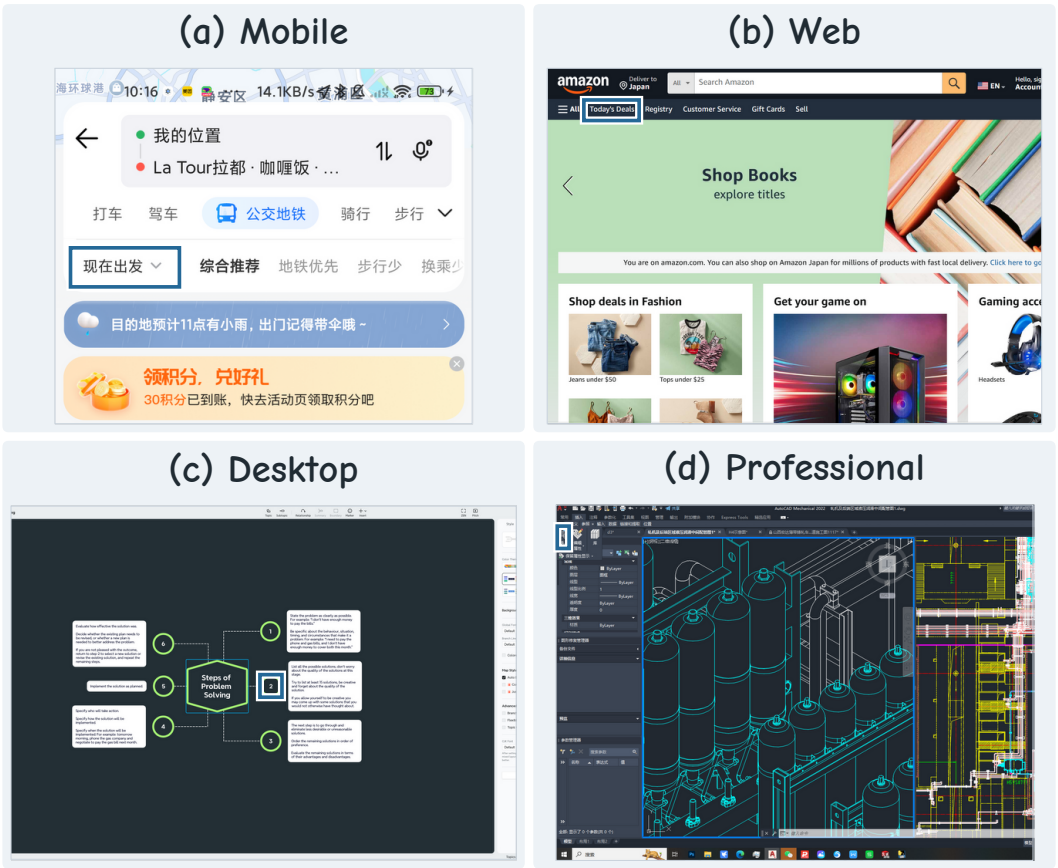


Figure 3. Representative GUI environments: (a) an Android transit application, (b) an Amazon webpage, and (c) an XMind desktop workspace from MMBench-GUI L2 [36]; and (d) an AutoCAD workspace from ScreenSpot-Pro [13]. Contextual crops retain the original target annotations (blue boxes); short labels identify the selected controls.

Structured and symbolic observations. GUI grounding may also use machine-readable interface structures from the platform or symbolic information from interface parsing. Common forms in-

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

clude HTML/DOM nodes on web pages [6,37], accessibility representations in WebArena [7] and AndroidWorld [9], and mobile view hierarchies [38].

Hybrid observations. Hybrid observations combine rendered pixels with structured or symbolic interface information, including upstream screenshot parsing results. OmniParser [39], for example, provides annotated screenshots, OCR text, and generated icon descriptions. Models can refer to candidate elements by region ID.

2.3. Target Space and Representations

The target space $\mathcal{Y}_{\tau_t}(o_t)$ depends on the interface representation and action type. Targets may be represented as spatial targets, symbolic targets, or composite spatial arguments. We omit the step subscript t for a single grounding instance.

Spatial targets. Spatial targets specify location or extent in interface coordinates. Let Ω_I denote the spatial domain of the current interface. A point target $p = (x, y) \in \Omega_I$ uses pixel, normalized, or discretely encoded coordinates. Points specify click or touch locations in screenshot-based GUI grounding [4,16]. A target's extent can be represented by a bounding box $B = (x_1, y_1, x_2, y_2)$ or a region $R \subseteq \Omega_I$.

Symbolic targets. A grounding target can be represented as a node reference, element identifier, or candidate index rather than explicit coordinates [6,7]. Such outputs depend on the available interface structure or candidate set, with interaction locations obtained from the geometry of the selected element when needed.

Action-conditioned targets. For interaction type τ, a target output may pair an element or region c with spatial arguments ϕ_τ: $y = (c, \phi_\tau)$. The referent c can be omitted when ϕ_τ identifies the target. For example, a click uses $\phi_{\text{click}} = p$; an endpoint-only drag uses the ordered pair $\phi_{\text{drag}} = (p_{\text{src}}, p_{\text{dst}}) \in \Omega_I \times \Omega_I$ for its source and destination [23,35].

Target validity. Let $\mathcal{Y}^*(o, q, h; \tau)$ denote outputs satisfying the current intent and action requirements for observation o, instruction q, optional history h, and action type τ. A prediction is valid exactly when this condition holds:

$$\hat{y} \in \mathcal{Y}^*(o, q, h; \tau) \subseteq \mathcal{Y}_\tau(o). \tag{3}$$

A single target may admit multiple valid spatial outputs. For example, under point-in-region evaluation, any point within the reference region R^* is valid. Different interface elements may also satisfy the same intent, such as a shopping-cart icon and a clickable text link that both provide access to the cart [17]. When $\mathcal{Y}^*(o, q, h; \tau) = \varnothing$, returning NoValidTarget correctly indicates that no valid target exists. Figure 4 illustrates target absence, precise positioning within controls, and drag endpoint localization.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

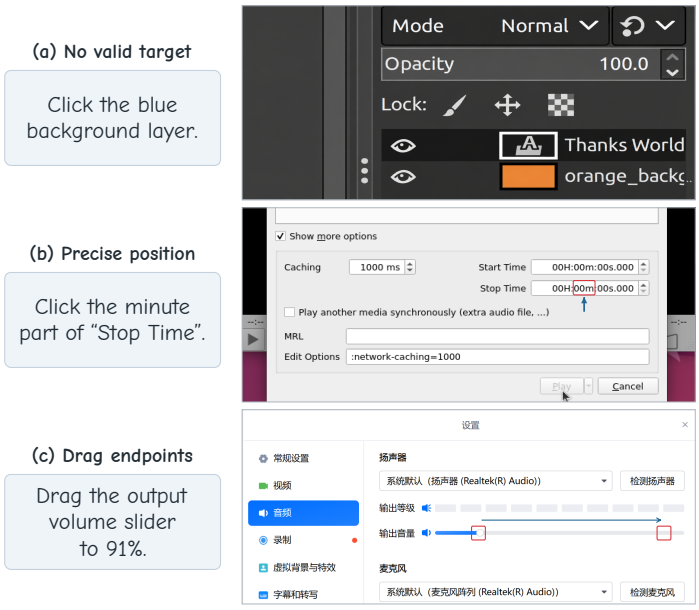


Figure 4. Grounding requirements beyond element-level localization: (a) no valid target, (b) precise within-control positioning, and (c) drag endpoint localization. Examples are adapted from OSWorld-G [40] in (a) and (b), and FineState-Bench [41] in (c). In (c), amber and blue boxes mark the start and target regions, respectively; the arrow indicates the requested movement.

3. Method Taxonomy

This section systematically reviews GUI grounding methods along three complementary dimensions: learning paradigms, inference strategies, and target representations. As illustrated in Figure 5, these dimensions address capability acquisition, evidence organization during inference, and the spatial representation of interaction targets, respectively. Table 2 provides an overview of representative methods across the three dimensions.

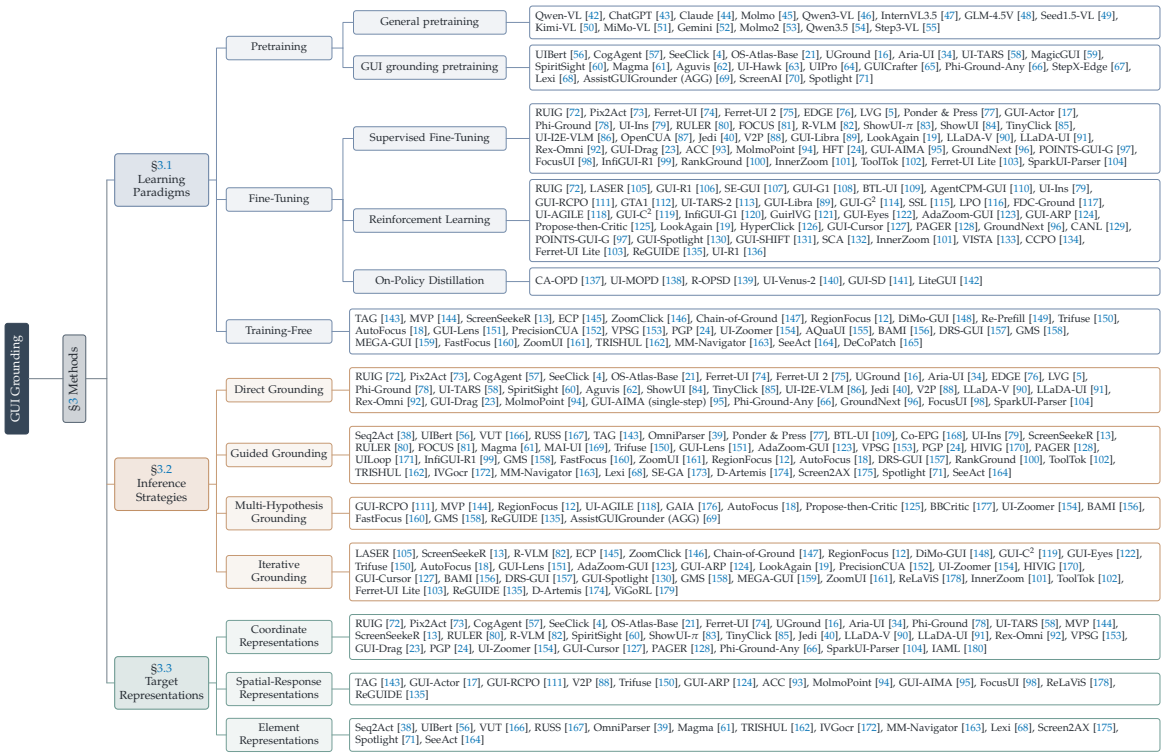


Figure 5. GUI grounding methods organized by learning paradigms, inference strategies, and target representations. Within each dimension, methods are assigned by their principal grounding contribution and configuration; multiple categories are retained only for multiple principal mechanisms.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

Table 2. Comparison of representative GUI-grounding methods. Entries characterize grounding-specific configurations, with inference and target representations reported for the trained or adapted grounder.

Method	Year	Backbone	Inference	Target representation	Key mechanism
GUI pretraining					
UI-TARS [58]	2025	Qwen2-VL	Direct	Coordinate: point	Combines description-to-coordinate grounding with GUI perception tasks and unified action supervision.
SpiritSight [60]	2025	InternVL2	Direct	Coordinate: block-local point	Combines multi-level GUI pretraining with block-wise positional encoding and block-local coordinates.
Phi-Ground-Any [66]	2026	Phi-3.5-Vision	Direct	Coordinate: action keypoints	Pretrains on synthetic click, drag, and draw tasks with rendered layouts and annotated image regions.
Fine-tuning: supervised learning					
GUI-Actor [17]	2025	Qwen2-VL; Qwen2.5-VL	Multi-hypothesis	Spatial-response: patch scores	Supervises action-token attention over target patches, then verifies marked candidate locations.
FocusUI [98]	2026	Qwen2.5-VL	Direct	Spatial-response: patch scores	Learns instruction-guided token selection and uses PosPad markers to preserve positional continuity.
GUI-Drag [23]	2025	Jedi	Direct	Coordinate: drag endpoints	Synthesizes text-dragging labels from word-level OCR and trains jointly with click data.
Fine-tuning: reinforcement learning					
GTA1 [112]	2025	UI-TARS-1.5; OpenCUA	Direct	Coordinate: point	Optimizes grounding with GRPO and binary rewards for clicks inside the target region.
SE-GUI [107]	2025	Qwen2.5-VL	Direct	Coordinate: point	Uses dense distance rewards and attention-guided loss filtering across self-evolution stages.
GUI-Eyes [122]	2026	Qwen2.5-VL	Iterative	Coordinate: point	Learns whether and how to crop or zoom using accuracy, proximity, and target-coverage rewards.
Fine-tuning: on-policy distillation					
GUI-SD [141]	2026	Qwen3-VL	Direct	Coordinate: point	Distills from a target-highlighted self-teacher, weighting tokens by digit significance and teacher entropy.
LiteGUI [142]	2026	Qwen3-VL	Direct	Coordinate: point	Distills from a teacher conditioned on matched valid actions, followed by action- and plan-level GRPO.
R-OPSD [139]	2026	Qwen2.5-VL; Qwen3-VL	Direct	Coordinate: point or box	Adapts at test time via reflection-guided self-distillation and contrastive calibration of failed rollouts.
Training-free enhancement					
GUI-RC [111]	2025	Qwen2.5-VL; UI-TARS-1.5	Multi-hypothesis	Spatial-response: voting grid	Aggregates predictions into pixel votes and selects the largest connected region of maximally voted pixels.
MVP [144]	2025	Qwen3-VL; GTA1	Multi-hypothesis	Coordinate: points	Clusters coordinates from attention-guided crops and the full image, then averages the largest cluster.
UI-Zoomer [154]	2026	Qwen2.5-VL; UI-Venus	Multi-hypothesis + iterative	Coordinate: box	Gates zooming by spatial agreement and token confidence; sizes crops by box spread and extent.
PGP [24]	2026	Qwen3-VL; InternVL3.5	Guided	Coordinate: box	Prompts the model to parse element semantics and boxes before grounding within a single inference round.

3.1. Learning Paradigms

We categorize GUI grounding methods into three learning paradigms: *pretraining*, which establishes transferable grounding capabilities; *fine-tuning*, which adapts pretrained models to GUI grounding; and *training-free* methods, which improve localization without parameter updates.

3.1.1. Pretraining

Pretraining develops transferable visual, semantic, and spatial representations for GUI grounding. Existing approaches fall into two categories: general MLLM pretraining, which acquires grounding through broad multimodal learning, and GUI-specific pretraining, which explicitly incorporates interface data and localization supervision.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

General pretraining. General MLLMs acquire grounding capabilities through multimodal pretraining. Qwen-VL [42] introduces region-level image-text-box alignment, while Molmo [45] incorporates pointing into multimodal generation. Recent general-purpose MLLMs, including Qwen3-VL [46], Qwen3.5 [54], InternVL3.5 [47], GLM-4.5V [48], Step3-VL [55], Seed1.5-VL [49], Kimi-VL [50], and MiMo-VL [51], provide visual grounding and spatial reasoning capabilities that support GUI interaction. Proprietary model families, including GPT-6 Astra [43], Claude [44], and Gemini [52], provide general multimodal foundations for GUI agents.

GUI grounding pretraining. GUI-specific pretraining explicitly learns semantic-spatial alignment from interface data. Grounding-focused approaches include SeeClick [4], which uses automatically collected web and mobile element descriptions and locations; UGround [16], which scales referring-expression supervision on web data and transfers to mobile and desktop interfaces; and OS-Atlas-Base [21], which trains on over 13 million elements across Windows, Linux, macOS, Android, and web environments.

Other approaches integrate grounding with broader GUI understanding. CogAgent [57] combines text recognition, grounding, and interface understanding using screenshots and DOM supervision. GUICourse [181] jointly trains on OCR and grounding tasks on GUIEnv-global before incorporating GUI knowledge and action data. SpiritSight [60] integrates text and icon recognition with position- and function-based grounding, while UIPro [64] learns grounding through large-scale GUI understanding pretraining. UI-TARS [58] and MagicGUI [59] similarly incorporate grounding into broader GUI perception and understanding.

3.1.2. Fine-Tuning

Supervised fine-tuning. Supervised fine-tuning (SFT) adapts pretrained models to GUI grounding through instruction-target supervision. One direction improves training data coverage and diversity. Ferret-UI [74] jointly learns referring, grounding, and interface understanding from region-annotated mobile screens, while Ferret-UI 2 [75] extends training across platforms and resolutions. EDGE [76] synthesizes multi-granularity GUI supervision, and TinyClick [85] combines auxiliary tasks and augmentation to adapt a compact vision-language model. ShowUI [84] balances heterogeneous grounding and navigation data through interleaved training and resampling.

Other methods refine localization supervision. R-VLM [82] introduces an IoU-aware objective that emphasizes predictions with greater target overlap. GUI-Actor [17] trains an action head to identify target-related visual patches, while GUI-AIMA [95] aligns intrinsic multimodal attention with patch-level grounding signals. V2P [88] sharpens localization by suppressing off-target responses and emphasizing target centers. LVG [5] incorporates UI element relationships through layout-guided contrastive learning. These methods refine grounding supervision through spatial overlap, patch-level alignment, or interface structure.

SFT also provides an initial policy for subsequent reinforcement learning (RL) optimization. UI-Ins [79] trains on diverse synthesized instructions before applying RL to optimize reasoning pathways. POINTS-GUI-G [97] combines curated data and resolution-consistent SFT with verifiable grounding rewards. GUI-Libra [89] uses action-aware SFT with mixed reasoning and direct-action examples and action-token reweighting, followed by RL policy optimization.

Reinforcement learning. We consider reward-based and preference-based policy optimization, including direct preference optimization (DPO), under RL. Early approaches use verifiable action rewards. GUI-R1 [106] rewards action types, click positions, and action parameters, while GTA1 [112] applies GRPO with binary rewards for clicks inside target bounding boxes. To address sparse feedback, SE-GUI [107] introduces distance-based point rewards. GUI-G^2 [114] models targets using Gaussian point and coverage rewards, while LPO [116] combines spatial distance with region-level information preferences.

Beyond reward design, methods improve exploration and policy stability. GUI-G1 [108] uses direct-output templates, constrained rewards, and difficulty-aware updates to mitigate excessive reasoning and reward exploitation. InfiGUI-G1 [120] encourages multiple coordinate candidates through adaptive exploration rewards. FDC-Ground [117] combines distance feedback with fact-

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

aligned completion pruning to select consistent rollouts. GuirlVG [121] introduces adaptive KL constraints to limit policy deviation. UI-AGILE [118] combines continuous rewards with cropping-based resampling, while VISTA [133] constructs GRPO groups from target-preserving views for more informative relative advantages.

RL also optimizes visual evidence acquisition and refinement. Ferret-UI Lite [103] refines initial predictions using local crops through RL and visual tool use. GUI-Eyes [122] learns adaptive cropping and zooming policies by rewarding both tool selection and grounding accuracy. LookAgain [19] optimizes a predict–inspect–refine process that verifies and revises initial coordinates using visual feedback.

Recent methods reduce dependence on annotated targets through self-generated supervision. GUI-RCPO [111] derives rewards from spatial agreement among sampled predictions for unlabeled test-time adaptation. CANL [129] constructs pseudo targets from coordinate-token confidence and uses distance-based feedback. Its negative-learning variant suppresses predictions far from pseudo targets to limit reinforcement of incorrect predictions. Such approaches enable annotation-free adaptation but depend on the reliability of self-generated signals.

On-policy distillation. On-policy distillation (OPD) provides token-level teacher supervision over student-generated trajectories. GUI-SD [141] introduces on-policy self-distillation (OPSD) with privileged visual guidance for coordinate generation, while LiteGUI [142] uses an external teacher before RL optimization. To improve supervision reliability, Huang et al. [182] apply correctness-aware gating and confidence scaling, while CA-OPD [137] uses teacher confidence to retain or correct student tokens during generation. R-OPSD [139] extends self-distillation to unlabeled test-time grounding through reflective feedback and calibrated updates.

3.1.3. Training-Free

Training-free methods improve GUI grounding without parameter updates by leveraging visual feedback, internal model signals, and additional inference-time observations.

Feedback and internal signals. Existing model predictions and internal representations provide grounding cues without further training. Chain-of-Ground [147] exposes intermediate predictions as visual references for correction while retaining global context. PrecisionCUA [152] similarly uses visual feedback to refine cursor predictions in dense code-editor interfaces. Other methods recover localization signals from the model itself: TAG [143] extracts instruction-related attention, while Trifuse [150] combines attention with OCR text and icon semantics. Re-Prefill [149] reuses target-related visual representations during a second prefill stage to guide coordinate prediction.

Visual augmentation. Local inspection and multiple views enrich the visual evidence available to an existing grounder. Iterative Narrowing (IN) [183] progressively narrows the view around predicted locations, while RegionFocus [12] combines focused views with visual landmarks to retain search context. DiMo-GUI [148] refines text- and icon-focused predictions through separate processing, and ZoomClick [146] examines zooming and stopping strategies. MVP [144] complements local inspection with spatial aggregation of predictions from instruction-guided crops. These approaches use additional visual processing to recover local detail or obtain complementary predictions, with benefits that depend on preserving relevant context and combining predictions appropriately.

Adaptive inference. Adaptive region inspection and search allocate additional inference computation according to intermediate grounding evidence. UI-Zoomer [154] uses prediction agreement and confidence to decide whether to zoom, while BAMI [156] combines coarse-to-fine focusing with candidate assessment. DRS-GUI [157] organizes region exploration through Monte Carlo tree search. GUI-Lens [151] uses OCR and detected UI components as spatial references for successive region inspection, while AutoFocus [18] uses uncertainty across coordinate hypotheses to guide view selection. Their effectiveness depends on whether intermediate grounding evidence reliably guides further inspection.

Summary. Pretraining builds transferable grounding capabilities from diverse interfaces, while fine-tuning improves localization and action prediction through targeted supervision. Both depend on

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

data coverage and supervision quality. Training-free methods enhance predictions without parameter updates, often at additional inference cost. These paradigms are complementary and can be combined, with their suitability determined by supervision availability, model access, and inference budget.

3.2. Inference Strategies

Inference strategies characterize how grounding evidence is organized and processed. We distinguish four complementary patterns: *direct grounding* predicts targets directly from the input; *guided grounding* introduces auxiliary evidence; *multi-hypothesis grounding* aggregates or selects among candidate predictions; and *iterative grounding* updates evidence based on intermediate hypotheses to refine localization. Figure 6 illustrates their evidence flows through representative methods.

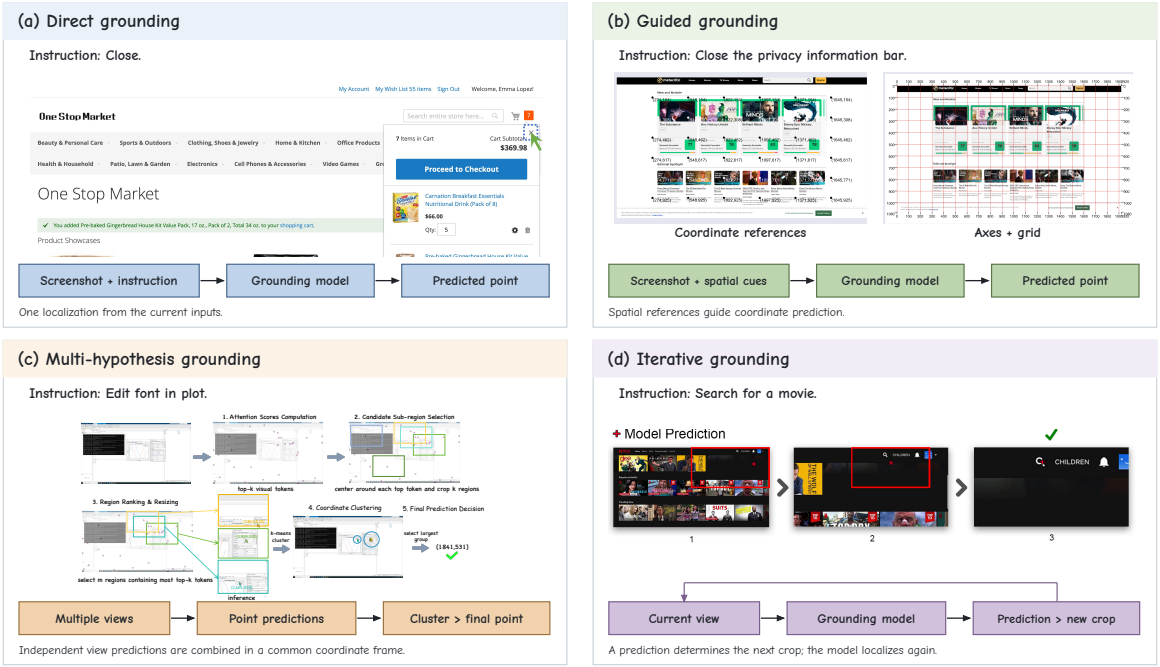


Figure 6. Representative GUI-grounding inference strategies: (a) direct grounding, (b) guided grounding, (c) multi-hypothesis grounding, and (d) iterative grounding. Examples are adapted from SeeClick [4], Li et al. [184], MVP [144], and Iterative Narrowing [183], respectively.

3.2.1. Direct Grounding

Direct grounding predicts spatial targets from the current GUI and instruction without explicitly constructing auxiliary grounding evidence or refining intermediate target hypotheses. This definition concerns evidence organization and allows multiple internal decoding steps.

Coordinate generation. A common implementation is autoregressive coordinate generation. SeeClick [4], UGround [16], and OS-Atlas-Base [21] map screenshots and referring expressions to target coordinates, while UI-TARS [58] directly predicts target points in its single-step grounding configuration. Direct prediction also supports richer spatial outputs: GUI-Drag [23] predicts both drag endpoints for text selection. Other methods use diffusion-based decoding. GUI-adapted LLaDA-V [90] generates target boxes through masked-token denoising, while LLaDA-UI [91] uses block-wise diffusion to produce target coordinates in its single-turn grounding setting.

Direct methods also differ in how they preserve visual detail before prediction. CogAgent [57] combines low- and high-resolution features for target localization. Ferret-UI [74] combines full-screen and enlarged subimage features, while Ferret-UI 2 [75] uses adaptive gridding across screen shapes and platforms. Such input processing supplies detailed visual features without requiring an intermediate target prediction to guide subsequent inspection.

Spatial feature readout. Other methods derive locations from spatial responses over visual features. GUI-Actor [17], without its verifier, selects target patches through an attention-based action head. GUI-

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

AIMA [95] aggregates query-relevant attention for patch localization in its single-step configuration. MolmoPoint [94] uses existing image features to select a patch, a subpatch, and a point within it. These methods illustrate that direct grounding can use spatial scoring or hierarchical readout as well as coordinate generation.

3.2.2. Guided Grounding

Guided grounding uses auxiliary evidence before target selection to narrow the search space, support coordinate prediction, or clarify instructions.

Candidate-based guidance. Visual parsing and explicit marks provide candidate-based guidance from screenshots. OmniParser [39] provides marked element candidates with textual and icon semantics and spatial information to support target selection. Magma [61] uses Set-of-Mark (SoM) prompting with numerically labeled candidate regions for action grounding. Parsing-guided Prompt (PGP) [24] lists visible icons and text with their semantics and coordinates, then localizes the target using this information, all in one inference round.

Spatial references and decoding guidance. Spatial guidance supports coordinate prediction through explicit positional references or corrections during decoding. RULER [80] aligns coordinate-bearing tokens with visual patches via shared positional encodings, enabling prediction relative to known positions. Li et al. [184] overlay axes, grids, and labeled intersections on screenshots as spatial references. Vision-PE Shuffle Guidance (VPSG) [153] suppresses coordinate biases during digit generation using auxiliary decoding with shuffled visual positional encodings.

Semantic reformulation. Semantic guidance clarifies targets by reformulating instructions or describing relevant interface content. Ponder & Press [77] uses an interpreter to describe targets for a separate visual locator. Semantic guidance can also be generated within the grounding process through target descriptions or intermediate reasoning. TAG [143] generates target descriptions and uses attention from their tokens for localization. In its slow route, FOCUS [81] summarizes the interface and analyzes visual details before predicting coordinates. UI-Ins [79] reasons from perspectives such as appearance, functionality, location, and intent.

3.2.3. Multi-Hypothesis Grounding

Multi-hypothesis grounding determines the final prediction by aggregating or selecting among multiple predicted target locations. These hypotheses may arise from repeated sampling, different visual views, or distinct reasoning branches.

Spatial consensus. Spatial consensus determines the target from agreement among predicted locations. GUI-RCPO [111] maps sampled points or boxes onto a spatial voting grid to identify a consensus region. MVP [144] clusters predictions from attention-guided crops and returns the centroid of the densest spatial cluster. UI-Zoomer [154] votes among sampled boxes based on their overlap and uses token confidence to select among equally supported candidates.

Candidate assessment. Candidate assessment evaluates how well alternative predictions match the instruction and visual content. RegionFocus [12] marks predictions from focal regions on an image-as-map and uses these visual references to select among candidates. UI-AGILE [118] generates predictions from overlapping subimages and scores each candidate using a crop centered on its predicted location. AutoFocus [18] selects among predictions from global and local crops by visually comparing the marked candidate locations. BAMI [156] uses an external correction model to select among alternative target boxes. GUI-Actor [17] uses its verifier to assess locations proposed by its action head, while Propose-then-Critic [125] ranks coordinate proposals with a visual critic.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

3.2.4. Iterative Grounding

Iterative grounding uses intermediate target hypotheses to update the evidence for subsequent localization. Let $e^{(0)}$ denote the initial evidence constructed from the observation and task inputs. At iteration k, the model produces a hypothesis $z^{(k)}$ and updates the evidence:

$$\begin{aligned} z^{(k)} &= f_\theta\Big(e^{(k)}\Big), \\ e^{(k+1)} &= \Phi\Big(e^{(k)}, z^{(k)}\Big). \end{aligned} \tag{4}$$

Here, Φ updates the evidence using predicted coordinates, candidate regions, or target-related internal representations. The index k denotes refinement within a grounding instance.

Region refinement and search. Revisiting regions around initial predictions reveals finer visual detail. IN [183] iteratively crops around predicted coordinates, while R-VLM [82] refines its initial prediction in a zoomed-in region. ECP [145] uses a coarse prediction from a downsampled image to crop the high-resolution original. ZoomClick [146] checks agreement between full-image and tile predictions before progressively narrowing the view. DiMo-GUI [148] refines text- and icon-focused predictions separately through dynamic zooming, stopping when successive predicted coordinates are sufficiently close. BAMI [156] iteratively crops around selected target boxes for further localization.

Attention signals can also guide region selection. Trifuse [150] fuses attention, OCR text, and icon semantics for coarse localization before zooming in. ZoomUI [161] iteratively crops high-attention regions for further localization. ScreenSeekeR [13] performs cascaded search over regions proposed by a planner. GMS [158] combines recursive region selection by a generalist scanner with specialist localization and local verification. DRS-GUI [157] searches instruction-relevant regions through focusing, shifting, and expansion before predicting coordinates once. Region hypotheses can thus guide inspection without provisional coordinate predictions. Narrowing may lose context or exclude the target, motivating context preservation and search recovery. Context-Aware Cropping [185] repeats grounding on a large crop around the initial prediction. MEGA-GUI [159] narrows the view toward predictions within the current region and can expand it when predictions fall outside. RegionFocus [12] marks failed search locations as visual landmarks to guide subsequent searches.

Adaptive observation control. Beyond choosing where to inspect, adaptive methods decide whether to inspect further and at what scale. GUI-ARP [124] learns to choose between attention-guided refinement and direct prediction. When visual self-verification rejects an initial prediction, AutoFocus [18] maps sampled coordinates and uncertainty to a spatial field for view selection. UI-Zoomer [154] uses spatial consensus and token confidence to trigger a single zoom-in refinement, sizing the crop by prediction uncertainty. GUI-C^2 [119] bases further cropping on predicted box area and whether the box touches the crop boundary. AdaZoom-GUI [123] conditions zooming on predicted target size after instruction refinement. GUI-Lens [151] selects successive crop regions and scales using coordinate references from OCR and detected UI elements. GUI-Eyes [122] learns cropping and zooming policies. GUI-Spotlight [130] adaptively selects quadrant, color-guided, or rectangular crops and adds them to the localization context.

Visual feedback and correction. Beyond selecting new views, methods can expose prior predictions for visual verification and correction. LookAgain [19] inspects marked local crops to refine or confirm predictions. Chain-of-Ground [147] uses marked predictions as visual references for correction, while PrecisionCUA [152] marks prior coordinates on screenshots for iterative refinement in code editors. GUI-Cursor [127] renders a virtual cursor and uses cursor–target spatial relations for correction.

Summary. The four inference strategies differ in how they use grounding evidence. Direct grounding predicts targets without additional guidance or refinement, while guided grounding incorporates auxiliary cues. Multi-hypothesis grounding compares candidate predictions, whereas iterative grounding refines predictions through updated visual evidence. However, unreliable guidance, correlated prediction errors, and context loss during refinement can limit their effectiveness. Thus, additional candidates or refinement steps do not necessarily improve accuracy.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

3.3. Target Representations

In this section, we distinguish three types: *coordinate representations*, which encode targets as numerical geometry; *spatial-response representations*, which derive locations from scores over pixels, visual tokens, or regions; and *element representations*, which select targets from candidate interface elements and retrieve their geometry. This taxonomy focuses on final target readout rather than intermediate spatial structures. Figure 7 illustrates representative examples.

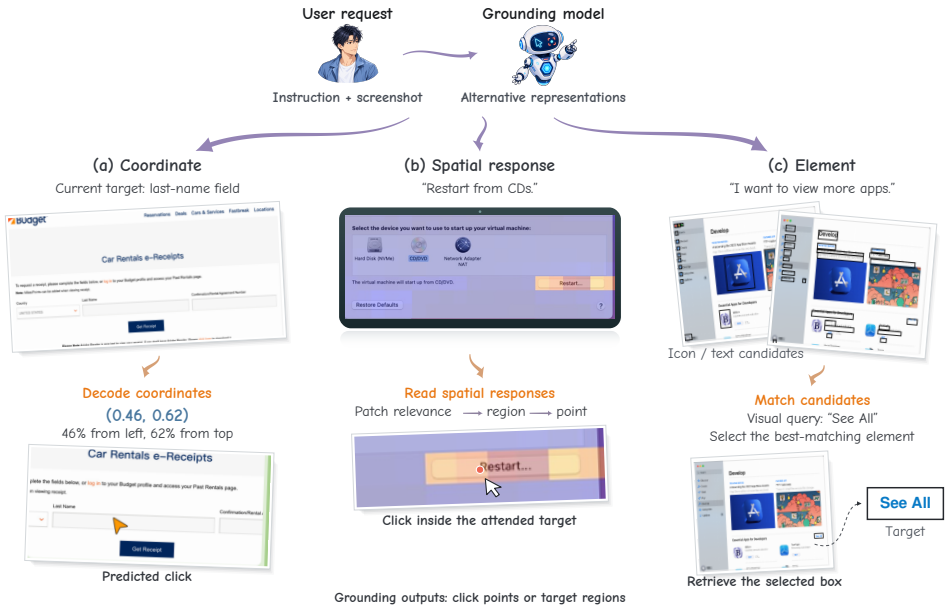


Figure 7. Target representations and spatial readout in GUI grounding: (a) decoding coordinates, (b) deriving a click location from spatial responses, and (c) retrieving the geometry of a selected element. Examples are adapted from SeeClick [4], GUI-Actor [17], and Layout-Aware GUI Grounding [186], respectively. Coordinates in (a) are normalized to the full input image. The red point and cursor in (b) illustrate spatial readout.

3.3.1. Coordinate Representations

Coordinate representations encode targets as explicit numerical geometry. They can represent a single location, a spatial region, or multiple locations required by more complex interactions. Their design involves both how this geometry is encoded and how the resulting coordinates are mapped to the screen.

Output forms. A click point (x, y) directly specifies an executable location, whereas a bounding box (x_1, y_1, x_2, y_2) preserves target extent. GUI-Drag [23], for example, extends grounding from a single click point to start and end coordinates for text dragging, while ShowUI-π [83] represents continuous drag trajectories through incremental cursor adjustments. Coordinate representations therefore range from points and regions to multi-point or sequential outputs, depending on the geometry required for execution.

Coordinate encoding and decoding. Coordinate outputs differ in both their serialization and decoding mechanisms. SeeClick [4] directly generates normalized point or box coordinates, while OS-Atlas-Base [21] unifies point and box outputs within a normalized coordinate range. Rex-Omni [92] instead represents quantized coordinates using dedicated tokens. Phi-Ground [78] adopts direct textual coordinate outputs after finding limited benefits from coordinate-digit label smoothing and reweighting at larger training scales. Coordinate decoding can also use diffusion-based generation. GUI-adapted LLaDA-V [90] formulates single-turn action and bounding-box prediction using discrete diffusion generation, while LLaDA-UI [91] uses block-wise diffusion decoding to generate structured GUI actions with normalized coordinate outputs.

Geometric mapping. Coordinates may use pixel values, normalized intervals, or discrete model-specific ranges, and may refer to the full image or local regions defined by crops or image partitions.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

SpiritSight [60] represents a target point using a block index and within-block coordinates, then reconstructs its global location during post-processing. UI-Zoomer [154] maps bounding boxes predicted within a zoomed crop back to the global coordinate frame. For a crop with top-left corner (x_0, y_0) and size (w_c, h_c) in the original image, a normalized local point $(u, v) \in [0, 1]^2$ is mapped to global coordinates as follows:

$$x = x_0 + uw_c, \qquad y = y_0 + vh_c. \tag{5}$$

Final accuracy therefore depends on both local prediction and correct transformation to the global reference frame.

Finer coordinate quantization cannot recover visual detail lost during image encoding. Accurate coordinate prediction also requires reliable alignment between visual positions and numerical coordinates. VPSG [153] uses auxiliary decoding with shuffled visual positional encodings to estimate and suppress coordinate-prediction biases. RULER [80] introduces explicit coordinate markers that share positional embeddings with image patches, making the position-to-coordinate mapping explicit.

3.3.2. Spatial-Response Representations

Spatial-response representations maintain target-related scores over visual patches, pixels, or spatial regions and derive the final location from these scores.

Spatial scores. GUI-Actor [17] introduces an attention-based action head that aligns a dedicated action token with relevant visual patches, while GUI-AIMA [95] uses intrinsic multimodal attention to score target patches. Other methods improve the quality of these spatial responses. V2P [88] suppresses background responses and encourages peaks near target centers.

Spatial scores can also be extracted from pretrained models without task-specific grounding fine-tuning. TAG [143] aggregates attention maps associated with generated target-description tokens to derive localization signals. Trifuse [150] maps OCR-derived textual relevance and icon-caption similarity onto a visual patch grid and combines them with attention responses.

Location readout. Spatial scores must be converted into an executable location. A correct response at coarse patch resolution may still be insufficient for precise interaction. MolmoPoint [94] addresses spatial granularity through hierarchical selection of a patch, a subpatch, and a point within it. ReLaViS [178] reads out the click location from its final spatial search distribution by taking a probability-weighted average of visual-token centers near the highest-scoring token.

Spatial readout can also operate on fields constructed from coordinate predictions. GUI-RCPO [111] projects sampled predictions onto a spatial voting grid and selects the largest connected region among maximally voted pixels. In its test-time aggregation stage, ReGUIDE [135] constructs a density field from coordinate predictions using kernel density estimation and selects its peak as the final location.

3.3.3. Element Representations

Element representations identify the target from a finite candidate set $\mathcal{E} = \{c_i\}_{i=1}^{M}$, where each candidate c_i has element attributes and associated geometry B_i. A scoring-based readout selects a single target:

$$\hat{i} \in \arg\max_{1 \le i \le M} s(q, c_i), \qquad \hat{B} = B_{\hat{i}}, \tag{6}$$

where $s(q, c_i)$ scores the instruction–candidate match and $\hat{B}$ is the geometry retrieved for the selected element.

Candidate structure. Candidates can be obtained from native interface structure or reconstructed from visual input. DOM and accessibility nodes associate element identities with textual or functional attributes and available geometry. Screenshot-based parsing provides analogous information: OmniParser [39] associates detected regions with text and functional descriptions, while Screen2AX [175] organizes detected and described elements into an accessibility hierarchy. TRISHUL [162] associates

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

local elements with broader interface regions and spatially informed functional descriptions to support action grounding.

Element selection and readout. Candidate attributes such as text, appearance, neighboring context, and layout relations provide complementary cues for discrimination. IVGocr [172] matches the target type and role extracted from the instruction to detected elements enriched with OCR text and returns the matched element's coordinates. In its referring-expression grounding task, Lexi [68] scores candidate UI regions against the expression and selects the highest-scoring region. Candidate identifiers provide another way to specify the selected target. In its SoM-based ScreenSpot configuration, Magma [61] selects among numerically marked candidate regions proposed by OmniParser. In the grounding evaluation of Screen2AX [175], GPT-4 selects an element identifier from the generated accessibility tree. Selection can also return the candidate's geometry directly: Layout-Aware GUI Grounding [186] matches structured target descriptions against layout-prior candidates and returns the bounding box of the selected candidate.

Representation constraints. Element representations depend on candidate coverage, selection accuracy, and geometry quality. If detection or filtering excludes the target, selection within the retained candidate set cannot recover it. Even when the target is present, selection can remain ambiguous: the oracle two-candidate marking experiment in GUI-Primitives [15] shows that errors persist when the target and distractor are explicitly supplied. Correct element selection can still yield an inaccurate interaction location when the associated geometry is incorrect, particularly for candidates reconstructed from screenshots [39,175]. Actions requiring a precise position within an element may also need additional spatial parameters.

Summary. The three target representations differ in how they turn a grounding decision into an interaction location. Coordinate representations predict the required geometry directly, so errors can arise in both coordinate prediction and mapping back to the screen. Spatial-response representations derive locations from spatial scores, but coarse responses may not support precise clicks. Element representations retrieve the geometry of a selected candidate, but cannot recover a target missing from the candidate set. Thus, accurate target identification alone is insufficient: the predicted coordinates, spatial readout, or retrieved element geometry must also be precise enough for the intended action.

4. Datasets and Evaluation

This section reviews GUI grounding datasets, benchmarks, and model performance. We first examine the evolution of training datasets, focusing on platform coverage, scale, and supervision signals. We then compare model performance across benchmarks to assess progress and cross-scenario generalization. Finally, we analyze benchmark difficulty and discriminative power, identifying saturated evaluations and remaining challenges to guide future research.

Table 3. Representative training resources for GUI grounding. M/D/W: mobile/desktop/web; trajs.: trajectories.

Resource	Year	Plat.	Scale
MobileViews [187]	2024	M	>1.2M screens
SeeClick [4]	2024	M/W	1M samples
AutoGUI [188]	2024	W	702K samples
UGround [16]	2024	M/W	10M elements
OS-ATLAS [21]	2024	M/D/W	>13M elements
MultiUI [189]	2024	W	7.3M samples
AgentTrek [190]	2024	W	10.4K trajs.
ScaleCUA-Data [191]	2025	M/D/W	>17.1M samples
GroundCUA [96]	2025	D	>3.56M elements
GUI-Drag [23]	2025	D/W	161K samples
Jedi [40]	2025	M/D/W	4M examples
AUTO-Explorer [192]	2025	D/W	17.5K samples
GUI-360° [193]	2025	D	79.5K samples
ScreenParse [194]	2026	W	21.1M elements
CMGUI [195]	2026	M	18.9K samples
MolmoPoint-GUISyn [94]	2026	M/D/W	>2M points
DragOn [35]	2026	D/W	3.5M tasks
DeskForge-1M [196]	2026	D	159.7M elements

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

4.1. Training Data and Resources

GUI grounding training data have evolved from mobile interface collections to large-scale cross-platform resources with richer annotations and interaction tasks. Early mobile resources established these foundations: RICO [197] provided Android screenshots and view hierarchies with element geometry and text; Widget Captioning [198] paired UI elements with crowdsourced descriptions; and UI RefExp [56] linked referring expressions to target elements. CLAY [199] improved the underlying layout annotations by removing visually invalid nodes and assigning semantic types.

From 2024 onward, datasets expanded in scale and platform coverage. Table 3 summarizes representative datasets by release year, platform coverage, and scale. SeeClick [4] integrated automatically collected web data, existing mobile annotations, and general vision-language instructions. GUIEnv-global [181] provided webpage text, element locations, and layout order, while MobileViews [187] expanded Android screenshot–view-hierarchy pairs. OS-ATLAS [21] and ScaleCUA-Data [191] further extended grounding data across mobile, desktop, and web interfaces.

Annotation methods increasingly enrich grounding supervision with semantic descriptions. UGround [16] generated referring expressions covering appearance, function, position, and element relationships. AMEX [200] combined model-generated screen and function descriptions with human verification, while AutoGUI [188] inferred element functions from interface changes. MultiUI [189] used accessibility-tree and page information to construct grounding and auxiliary UI-understanding tasks. CMGUI [195] added grounding instructions with rationales for Chinese mobile interfaces.

Recent work has introduced synthetic interfaces, dense annotations, and complex interactions. Jedi [40] combined collected icons, code-rendered components, application layouts, and filtered existing data. MolmoPoint-GUISyn [94] generated and rendered HTML interfaces across platforms, extracting element locations and multiple pointing instructions. GroundCUA [96] provided dense, human-verified element boxes and labels for desktop applications, while ScreenParse [194] annotated webpage elements with bounding boxes, semantic types, and text through structural extraction and model-based refinement. Beyond point grounding, GUI-Drag [23] synthesized text-selection examples with drag endpoints, while DragOn [35] extended supervision to cell selection, element resizing, and slider manipulation.

4.2. Evaluation Benchmarks and Metrics

Benchmarks. Table 4 summarizes representative benchmarks. ScreenSpot [4] established evaluation of text and icon/widget localization across mobile, desktop, and web interfaces. ScreenSpot-v2 [21] refined its instructions and bounding boxes while resolving ambiguous examples. WinSpot [201] focuses on Windows applications, while ScreenSpot-Pro [13] evaluates high-resolution professional software across six categories, emphasizing small targets and dense layouts.

Evaluation has since expanded to more complex instructions. UI-I2E-Bench [86] distinguishes explicit and implicit references, while UI-Vision [14] covers basic, functional, and spatial grounding. MMBench-GUI L2 [36] separates basic visual and positional descriptions from advanced functional instructions. OSWorld-G [40] provides original instructions alongside refined variants with more explicit grounding cues to reduce reliance on software-specific knowledge. CAGUI [110] evaluates grounding on Chinese Android interfaces through Text2Point and Fun2Point, while VenusBench-GD [202] provides bilingual, cross-platform evaluation of basic, reasoning, functional, and refusal grounding.

Recent benchmarks emphasize robustness and interaction complexity. Jandial et al. [203] assess prediction consistency across alternative descriptions of the same target, while GUI-Primitives [15] tests spatial reasoning through contrastive instruction pairs. GUI-Perturbed [204] varies webpage appearance and instruction formulation, and WinDeskGround [205] evaluates multi-window scenes with controlled occlusion, density, and distractors. GUI-HalluBench [24] examines confusable elements and nonexistent targets. For region prediction and precise interaction, UI-Vision [14] evaluates region grounding, ScreenDrag [23] requires text-selection drag endpoints, and FineState-Bench [41] assesses

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

state-conditioned control localization and interaction. Figure 8 illustrates representative tasks and outputs.

Evaluation metrics. Following the ScreenSpot benchmarks [4,13,21], point accuracy measures the proportion of predicted coordinates falling within the annotated target boxes:

$$\mathrm{Acc}_{\mathrm{point}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[\hat{p}_i \in B_i], \tag{7}$$

where N is the number of test instances, $\hat{p}_i$ is the predicted point, B_i is the target bounding box, and $\mathbb{I}[\cdot]$ is the indicator function. Region-based grounding uses IoU to measure spatial overlap, as in UI-Vision [14]. Element-based evaluation measures whether the correct candidate UI element is selected, as in UIBert [56] and Mind2Web [6].

Interaction-oriented metrics distinguish localization from precise interaction. ScreenDrag [23] reports Drag Trigger Rate (DTR) for valid drag generation and Success Rate (SR) for exact text selection. FineState-Bench [41] distinguishes target-control localization through SR@Loc from interactable-region localization through SR@Int, and further evaluates exact target-state achievement together with localization requirements.

Reliability metrics assess prediction consistency, rejection, and uncertainty. Consistency is evaluated through prediction dispersion across descriptions of the same target [203] and joint correctness on contrastive instruction pairs [15]. GUI-HalluBench [24] reports Localization Accuracy for confusable targets and Rejection Rate for absent targets. SafeGround [20] evaluates uncertainty-based error detection and selective grounding, measuring errors among accepted predictions and retention of correct predictions.

Table 4. Representative GUI grounding benchmarks grouped by primary evaluation focus. M/D/W: mobile/desktop/web. Sizes refer to the listed grounding evaluation sets.

Benchmark	Year	Plat.	Size	Key feature
Interface coverage				
ScreenSpot [4]	2024	M/D/W	1,272	Text and icons
ScreenSpot-v2 [21]	2024	M/D/W	1,272	Text and icons
ScreenSpot-Pro [13]	2025	D	1,581	High-resolution GUIs
WinSpot [201]	2025	D	5,000	Windows apps
GUI-360° [193]	2025	D	19,780	Office applications
MedSPOT [206]	2026	D/W	597	Clinical workflows
Instruction design				
MUG [207]	2024	M	6,208	Multi-turn instructions
UI-I2E-Bench [86]	2025	M/D/W	1,477	Explicit/implicit instructions
UI-Vision [14]	2025	D	5,479	Functional/spatial references
OSWorld-G [40]	2025	D	564	Software knowledge
MMBench-GUI [36]	2025	M/D/W	3,574	Goal-based instructions
CAGUI [110]	2025	M	3,000	Chinese instructions
VenusBench-GD [202]	2025	M/D/W	6,166	English/Chinese instructions
AutoGUI-v2 [208]	2026	M/D/W	1,518	Functional regions/elements
Robustness				
TransBench [209]	2025	M/W	Variable	Cross-domain transfer
GUI-Perturbed [204]	2026	W	3,120	Image/text perturbations
WinDeskGround [205]	2026	D	Variable	Occlusion and distractors
GUI-HalluBench [24]	2026	M/D/W	2,000	Confusable/absent targets
GUI-Primitives [15]	2026	D/W	994	Paired spatial instructions
ScreenHaystack [210]	2026	D	48,000	Spatial blind zones
Region and interaction grounding				
UI-Vision [14]	2025	D	311	Functional regions
ScreenDrag [23]	2025	D	5,333	Text selection
DragOn [35]	2026	D/W	2,000	Diverse drag actions
FineState-Bench [41]	2026	M/D/W	2,209	State-conditioned grounding

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

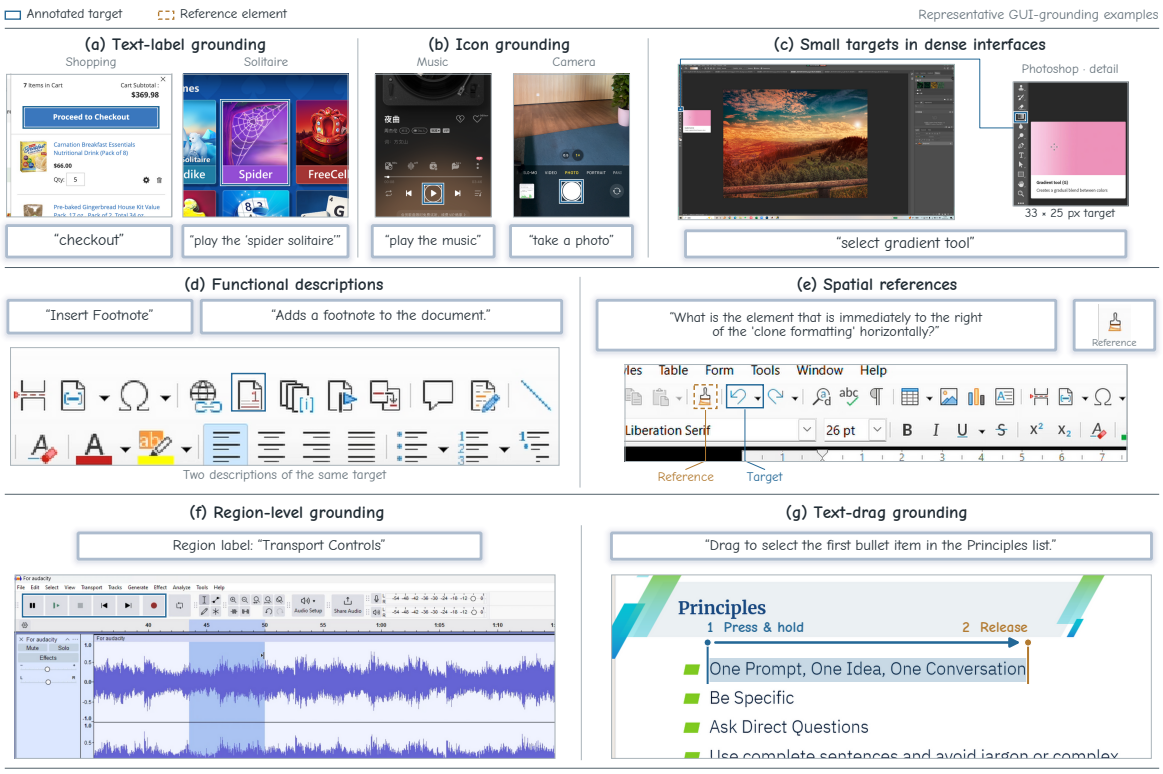


Figure 8. Representative GUI-grounding tasks: (a) text-label and (b) icon grounding from ScreenSpot-v2 [21]; (c) small-target grounding from ScreenSpot-Pro [13]; (d) functional descriptions, (e) spatial references, and (f) region-level grounding from UI-Vision [14]; and (g) text-drag grounding from ScreenDrag [23]. Blue boxes mark targets, and the dashed amber box marks the reference element. In (g), shading marks the target text span and the arrow indicates the drag direction.

4.3. Performance Comparison and Analysis

Table 5 compares reported results for general-purpose models, GUI-specific models, and models with inference-time enhancements. Grounding accuracy has improved substantially. On ScreenSpot-v2 [21], accuracy rises from 55.1% for SeeClick [4,17] to 97.5% for Qwen-UI-Agent-27B [22], indicating near-saturation. ScreenSpot-Pro [13] remains more challenging: the same Qwen-UI-Agent model achieves 76.6% without zooming. GPT-6 Astra [43] reaches 92.7% without tools, the highest reported result in Table 5.

Aggregate accuracy, however, masks important capability gaps. Figure 9 compares performance across target types, instruction complexity, interaction precision, and refusal tasks. Among the 99 ScreenSpot-Pro [13] configurations compiled in the figure, text accuracy exceeds icon accuracy in 97, with a median gap of 31.8 percentage points. Accuracy also declines for smaller targets [86]. ReGround experiments show that overly small crops lose useful context [13], while WinDeskGround [205] reveals sensitivity to partial occlusion.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

Table 5. Reported accuracy (%) for representative models on GUI grounding benchmarks. SSP: ScreenSpot-Pro; G/G-R: OSWorld-G with original/refined instructions; UIV: UI-Vision element grounding; MM: MMBench-GUI L2. Bold and underlined values indicate the best and second-best reported results, respectively, in each column across all model groups.

Model / configuration	SSP	G	G-R	UIV	MM
General-purpose models					
Qwen2.5-VL-7B [211]	27.6[a]	31.4[a]	–	0.9[a]	33.9[b]
Step3-VL-10B [55]	51.6	59.0	–	–	81.5
Qwen3-VL-8B [46]	54.6	58.2	67.4[c]	25.2[c]	83.0[c]
Qwen3.5-27B [54]	68.4[d]	–	67.9[d]	46.6[d]	89.1[d]
Gemini-3-Pro [212]	72.7	–	–	–	–
GPT-6 Astra [43]	**92.7**	–	–	–	–
GUI-specific models					
SeeClick [4]	1.1[a]	–	–	5.4[a]	–
OS-ATLAS-7B [21]	18.9[e]	27.7[e]	–	9.0[e]	41.4[e]
UGround-V1-7B [16]	31.1[e]	36.4[e]	–	12.9[e]	65.7[e]
UI-TARS-7B [58]	35.7	47.5[a]	–	17.6[a]	–
Jedi-7B [40]	39.5	54.1	–	24.8	–
GUI-Actor-7B [17]	44.6	–	56.6[f]	–	76.5[g]
SE-GUI-7B [107]	47.3	–	–	–	76.6[g]
GUI-AIMA-3B-lite [95]	49.8	–	58.3	27.9	76.3
InfiGUI-G1-7B [120]	51.9	–	–	26.1	80.8
V2P-7B [88]	52.5	52.5	–	28.8	79.9
MolmoPoint-GUI-8B [94]	60.2	<u>70.0</u>	–	–	–
GUI-SD-8B [141]	60.7	64.0	70.9	33.3	86.7
GTA1-32B [112]	63.6	65.2[g]	72.2[g]	–	83.4[g]
MAI-UI-8B [169]	65.8	60.1	68.6	40.7	88.8
UI-Venus-1.5-8B [213]	68.4	69.7	74.1	46.5	88.1
GUI-Owl-1.5-8B [3]	71.1	65.8	69.3	–	82.5
Qwen-UI-Agent-27B [22]	76.6	–	<u>78.5</u>	**70.0**	**92.6**
Models with inference-time enhancements					
UI-Venus-1.5-8B + RankGround [100]	71.0	**78.0**	**84.7**	–	89.8
Qwen3-VL-32B + MVP [144]	74.0	–	–	44.1	–
GUI-Owl-1.5-8B + Zoom + Re-Prefill [149]	80.1	68.8	75.0	41.6	85.1
Qwen-UI-Agent-27B + Zoom [22]	81.5	–	–	–	–
GPT-5.5 + GUI-Lens [151]	<u>87.9</u>	–	–	<u>68.6</u>	<u>91.5</u>

Model references identify the original work; superscripts indicate additional evaluation sources: [a][40]; [b][36]; [c][141]; [d][22]; [e][88]; [f][95]; [g][169].

Complex instructions remain challenging. On the Windows subset of MMBench-GUI L2 [36], MAI-UI-32B [169] drops from 93.0% on basic instructions to 78.7% on advanced instructions. On UI-Vision [14], spatial accuracy trails both basic and functional accuracy in 23 of 24 configurations. GUI-Primitives [15] further reports that 60–92% of predictions fall outside both candidate regions, identifying candidate localization as a major source of failure in relational grounding.

Correct target localization does not ensure precise interaction. On ScreenDrag [23], GUI-Drag-3B generates valid drag actions for all instances but achieves only 38.1% exact text-selection success. On FineState-Bench [41], platform-average SR@Int is lower than SR@Loc for all eight models under static evaluation, with a median gap of 20.9 percentage points. Reliability remains limited: equivalent target descriptions can yield inconsistent coordinates [203], while ten of 29 OSWorld-G [40] configurations achieve zero refusal accuracy under original instructions.

Inference-time enhancements offer substantial gains without model retraining. On ScreenSpot-Pro, RegionFocus [12] improves Qwen2.5-VL-72B from 47.8% to 61.6% through region refinement, MVP [144] raises Qwen3-VL-32B from 55.3% to 74.0% through multi-view aggregation, and GUI-Lens [151] improves GPT-5.5 from 74.8% to 87.9% through iterative cropping and verification.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

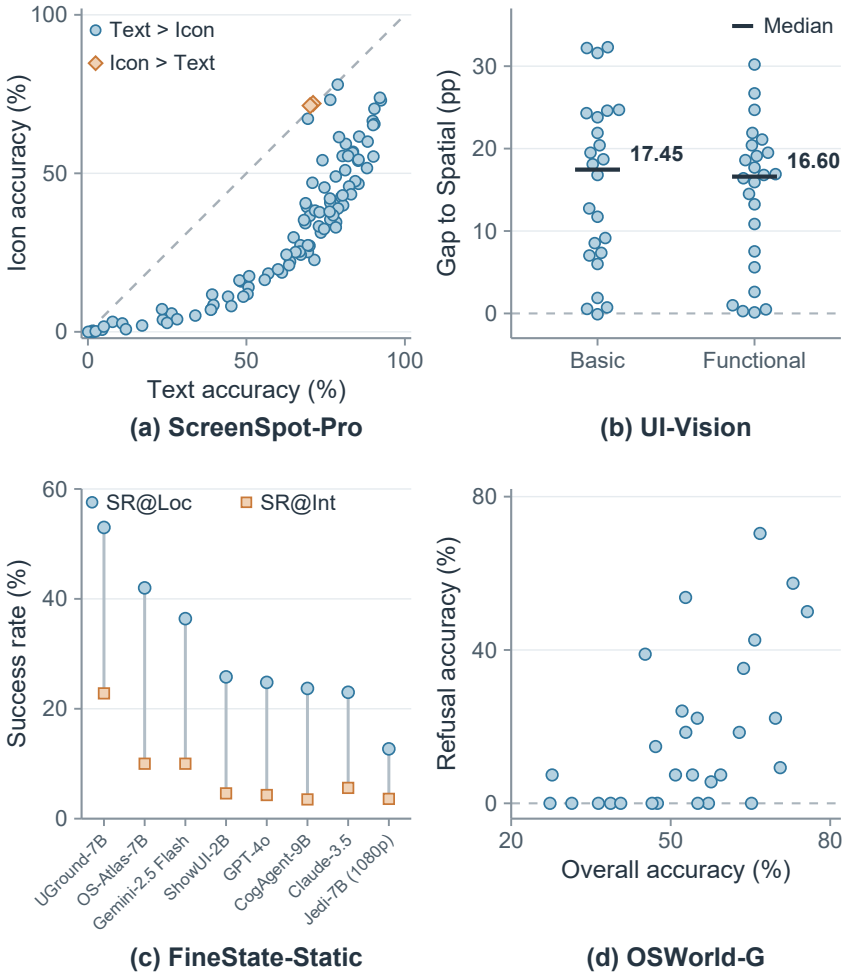


Figure 9. GUI-grounding capability gaps. (a) Text versus icon accuracy across 99 ScreenSpot-Pro [13] configurations; the dashed diagonal indicates equal accuracy. (b) Basic-minus-Spatial and Functional-minus-Spatial accuracy gaps, in percentage points, across 24 UI-Vision [14] configurations; black bars indicate medians. (c) Target-control (SR@Loc) and interactable-region (SR@Int) localization success rates, averaged across platforms, for eight models under the FineState-Bench [41] static protocol; vertical segments connect each model's two scores. (d) Overall versus refusal accuracy across 29 OSWorld-G [40] configurations using original instructions.

Figure 10 shows diminishing returns from increased inference budgets, with occasional performance declines. MVP [144] using GTA1-7B achieves 61.7% with four additional views but drops to 60.6% with ten. For GUI-Lens [151], switching from Balanced to Quality increases average latency from 87.7 to 118.0 seconds per sample, while accuracy rises only from 90.00% to 90.33% on its 300-example ScreenSpot-Pro subset. Thus, inference-time improvements must be weighed against their computational and latency costs.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

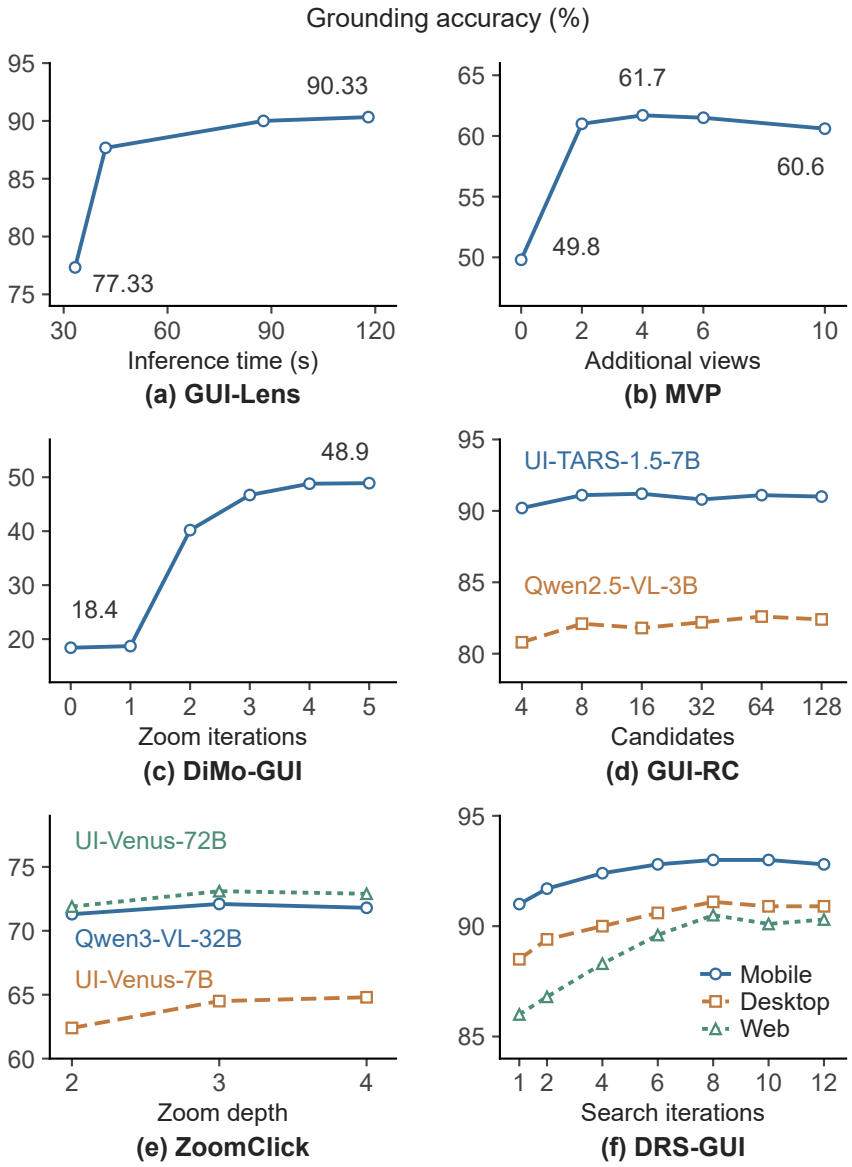


Figure 10. Grounding accuracy versus inference budget: (a) GUI-Lens [151], (b) MVP [144], (c) DiMo-GUI [148], (d) GUI-RC [111], (e) ZoomClick [146], and (f) DRS-GUI [157]. Panels (a), (b), (c), and (e) use ScreenSpot-Pro; (d) and (f) use ScreenSpot-v2. Panel (a) reports mean latency for GPT-5.5 on a 300-example subset; (b) uses GTA1-7B. The zooming strategy is fixed in (e), and search depth is fixed at three in (f).

5. Challenges and Future Directions

Despite advances on screenshot-based benchmarks, GUI grounding still faces challenges in semantic understanding, dynamic interfaces, prediction reliability, and inference efficiency. We discuss four corresponding research directions.

5.1. Semantic Grounding

GUI grounding must resolve targets described by their functions or relationships rather than explicit visual cues. Existing benchmarks reveal persistent difficulties with implicit and relational instructions [15,86], while coordinate-based metrics cannot distinguish target misidentification from localization errors. Future work should improve semantic reasoning across unfamiliar layouts through training with paraphrased and contrastive instructions, promoting consistency for equivalent descriptions and discrimination between different targets. Evaluation should separate target identification from coordinate precision, using controlled candidate-region tests to isolate semantic and spatial errors.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

5.2. Interactive Grounding

Most benchmarks evaluate grounding on isolated screenshots, overlooking how interaction history and interface changes affect target identity and location. Recent methods incorporate preceding actions and screen sequences [34,63], but determining which historical information remains valid is challenging. Future work should track targets across interface states, update their locations, and revise predictions when new observations contradict prior evidence. Evaluation should test target relocation under controlled changes, such as scrolling or pop-up activation, with action types and target descriptions fixed to isolate grounding from action selection and downstream execution.

5.3. Reliable Grounding

High localization accuracy does not guarantee reliability when targets are absent, ambiguous, or difficult to identify. Benchmarks expose weaknesses in invalid-target detection [24,40], while recent methods investigate prediction verification [19] and uncertainty estimation [20,214]. Future work should integrate these capabilities while distinguishing target absence from abstention due to uncertainty, since low confidence does not imply that no valid target exists. Evaluation should measure error correction, invalid-target detection, calibration, and robustness across interfaces and instruction variations, alongside risk-coverage trade-offs that quantify errors among accepted predictions and the proportion of requests accepted.

5.4. Efficient Grounding

Improving grounding accuracy often requires additional computation, particularly for high-resolution interfaces where image resizing obscures small targets [13]. Existing approaches employ multi-view processing [144], adaptive refinement [18], and visual-token compression [155], yet increased inference budgets do not consistently improve accuracy. Future work should allocate computation according to instance difficulty, determining when finer visual evidence or further refinement is beneficial while preserving small-target details. Evaluation should jointly report accuracy, end-to-end latency, and memory usage, with model size, hardware, input resolution, and inference budgets specified to enable comparisons under matched resource constraints.

6. Conclusion

This survey presents a unified formulation of GUI grounding and a taxonomy spanning learning paradigms, inference strategies, and target representations. By connecting these dimensions with training resources, evaluation benchmarks, and reported performance, it systematically characterizes existing methods and their capability boundaries.

Our analysis shows that advances in semantic–spatial alignment and inference-time visual refinement have improved grounding performance, yet strong results on static benchmarks do not necessarily translate into reliable interaction. Implicit instructions, small targets, and state-dependent operations continue to challenge semantic understanding, spatial precision, and interaction validity. Moreover, inference-time improvements depend on the quality of intermediate evidence and incur additional computational costs.

Future research should integrate semantic reasoning, interaction history, and visual feedback with uncertainty-aware prediction and adaptive computation. Evaluation should distinguish grounding errors from planning and execution failures while accounting for task conditions and inference budgets. These advances are essential for developing GUI grounders that accurately resolve interaction targets, produce valid spatial parameters, and recognize when predictions are insufficiently supported.

References

1. Wang, S.; Liu, W.; Chen, J.; Zhou, Y.; Gan, W.; Zeng, X.; Che, Y.; Yu, S.; Hao, X.; Shao, K.; et al. Gui agents with foundation models: A comprehensive survey. *arXiv preprint arXiv:2411.04890* **2024**.
2. Zhang, C.; He, S.; Qian, J.; Li, B.; Li, L.; Qin, S.; Kang, Y.; Ma, M.; Liu, G.; Lin, Q.; et al. Large language model-brained gui agents: A survey. *arXiv preprint arXiv:2411.18279* **2024**.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

3. 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**.
4. Cheng, K.; Sun, Q.; Chu, Y.; Xu, F.; YanTao, L.; Zhang, J.; Wu, Z. Seeclick: Harnessing gui grounding for advanced visual gui agents. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 9313–9332.
5. Qian, Y.; Lu, Y.; Hauptmann, A.G.; Riva, O. Visual grounding for user interfaces. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), 2024, pp. 97–107.
6. Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; Su, Y. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems* **2023**, *36*, 28091–28114.
7. Zhou, S.; Xu, F.F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. Webarena: A realistic web environment for building autonomous agents. In Proceedings of the International Conference on Learning Representations, 2024, Vol. 2024, pp. 15585–15606.
8. Rawles, C.; Li, A.; Rodriguez, D.; Riva, O.; Lillicrap, T. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems* **2023**, *36*, 59708–59728.
9. Rawles, C.; Clinckemaillie, S.; Chang, Y.; Waltz, J.; Lau, G.; Fair, M.; Li, A.; Bishop, W.; Li, W.; Campbell-Ajala, F.; et al. Androidworld: A dynamic benchmarking environment for autonomous agents. In Proceedings of the International Conference on Learning Representations, 2025, Vol. 2025, pp. 406–441.
10. Xie, T.; Zhang, D.; Chen, J.; Li, X.; Zhao, S.; Cao, R.; Hua, T.J.; Cheng, Z.; Shin, D.; Lei, F.; et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems* **2024**, *37*, 52040–52094.
11. Lu, Q.; Shao, W.; Liu, Z.; Du, L.; Meng, F.; Li, B.; Chen, B.; Huang, S.; Zhang, K.; Luo, P. Guiodyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. In Proceedings of the 2025 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2025, pp. 22404–22414.
12. Luo, T.; Logeswaran, L.; Johnson, J.; Lee, H. Visual test-time scaling for gui agent grounding. In Proceedings of the 2025 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2025, pp. 19989–19998.
13. Li, K.; Meng, Z.; Lin, H.; Luo, Z.; Tian, Y.; Ma, J.; Huang, Z.; Chua, T.S. Screenspot-pro: Gui grounding for professional high-resolution computer use. In Proceedings of the Proceedings of the 33rd ACM International Conference on Multimedia, 2025, pp. 8778–8786.
14. Nayak, S.; Jian, X.; Lin, K.Q.; Rodriguez, J.A.; Kalsi, M.; Awal, R.; Chapados, N.; Özsu, M.T.; Agrawal, A.; Vazquez, D.; et al. Ui-vision: A desktop-centric gui benchmark for visual perception and interaction. *arXiv preprint arXiv:2503.15661* **2025**.
15. Jahin, M.A.; Parvez, M.R. GUI-Primitives: Diagnosing Spatial Reasoning Failures in Vision-Language GUI Grounding. *arXiv preprint arXiv:2608.21832* **2026**.
16. Gou, B.; Wang, D.R.; Zheng, B.; Xie, Y.; Chang, C.; Shu, Y.; Sun, H.; Su, Y. Navigating the digital world as humans do: Universal visual grounding for gui agents. In Proceedings of the International Conference on Learning Representations, 2025, Vol. 2025, pp. 30851–30883.
17. Wu, Q.; Cheng, K.; Yang, R.; Zhang, C.; Yang, J.; Jiang, H.; Mu, J.; Peng, B.; Qiao, B.; Tan, R.; et al. Gui-actor: Coordinate-free visual grounding for gui agents. *Advances in Neural Information Processing Systems* **2026**, *38*, 15101–15128.
18. Yao, R.; Xiong, S.; Zou, T.; Xiong, S.; Rong, Y. AutoFocus: Uncertainty-Aware Active Visual Search for GUI Grounding. *arXiv preprint arXiv:2605.02630* **2026**.
19. Zhang, R.; Meng, H.; He, Y.; Shao, R.; Liu, A.H.; Nie, L. LookAgain: Closed-Loop GUI Grounding with Visually Grounded Reflection. *arXiv preprint arXiv:2608.09723* **2026**.
20. Wang, Q.; Fan, Y.; Wang, X.E. SafeGround: Know When to Trust GUI Grounding Models via Uncertainty Calibration. *arXiv preprint arXiv:2602.02419* **2026**.
21. Wu, Z.; Wu, Z.; Xu, F.; Wang, Y.; Sun, Q.; Jia, C.; Cheng, K.; Ding, Z.; Chen, L.; Liang, P.P.; et al. OS-ATLAS: Foundation action model for generalist GUI agents. In Proceedings of the International Conference on Learning Representations, 2025, Vol. 2025, pp. 5090–5108.
22. Zhou, H.; Tong, P.; Zhang, X.; Kong, Q.; Cai, C.; Xia, T.; Zhang, G.; Zhang, J.; Li, L.; Chen, L.; et al. Qwen-UI-Agent Technical Report: Toward Next-Generation Real-World Centric Foundation GUI Agents. *arXiv preprint arXiv:2607.28227* **2026**.
23. Liao, Z.; Lu, Y.; Gou, B.; Sun, H.; Awadallah, A. Beyond clicking: A step towards generalist gui grounding via text dragging. *arXiv preprint arXiv:2601.06031* **2025**.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

24. Zhang, Z.; Jing, H.; Lv, R.; Fang, S.; Zhu, S.; Wang, J.; Li, C.; Liu, X.; Ma, C.; Zhai, G. Exposing and Evaluating Hallucinations for GUI Grounding. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026, pp. 40208–40223.
25. Shi, Y.; Yu, W.; Huang, J.; Yao, W.; Chen, W.; Liu, N. Towards trustworthy gui agents: A survey. *arXiv preprint arXiv:2503.23434* **2025**.
26. Qiao, Y.; Deng, C.; Wu, Q. Referring expression comprehension: A survey of methods and datasets. *IEEE Transactions on Multimedia* **2020**, *23*, 4426–4440.
27. Xiao, L.; Yang, X.; Lan, X.; Wang, Y.; Xu, C. Towards visual grounding: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2025**.
28. Nguyen, D.; Chen, J.; Wang, Y.; Wu, G.; Park, N.; Hu, Z.; Lyu, H.; Wu, J.; Aponte, R.; Xia, Y.; et al. Gui agents: A survey. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 22522–22538.
29. Tang, F.; Xu, H.; Zhang, H.; Chen, S.; Wu, X.; Shen, Y.; Zhang, W.; Hou, G.; Tan, Z.; Yan, Y.; et al. A survey on (m) llm-based gui agents. *arXiv preprint arXiv:2504.13865* **2025**.
30. Hu, X.; Xiong, T.; Yi, B.; Wei, Z.; Xiao, R.; Chen, Y.; Ye, J.; Tao, M.; Zhou, X.; Zhao, Z.; et al. Os agents: A survey on mllm-based agents for computer, phone and browser use. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 7436–7465.
31. Sager, P.J.; Meyer, B.; Yan, P.; von Wartburg-Kottler, R.; Etaiwi, L.; Enayati, A.; Nobel, G.; Abdulkadir, A.; Grewe, B.F.; Stadelmann, T. A comprehensive survey of agents for computer use: Foundations, challenges, and future directions. *Journal of Artificial Intelligence Research* **2026**, *85*.
32. Chen, Y.; Wang, S.; Wang, Y.; Yao, Q. A Survey on Benchmarks of LLM-based GUI Agents. *Transactions on Machine Learning Research* **2025**.
33. Li, J.; Huang, K. A survey on gui agents with foundation models enhanced by reinforcement learning. *arXiv preprint arXiv:2504.20464* **2025**.
34. Yang, Y.; Wang, Y.; Li, D.; Luo, Z.; Chen, B.; Huang, C.; Li, J. Aria-ui: Visual grounding for gui instructions. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 22418–22433.
35. Bout, N.; Langevin, M.; Riochet, R. DragOn: A Benchmark and Dataset for Drag-Based GUI Interactions. *arXiv preprint arXiv:2606.06322* **2026**.
36. Wang, X.; Wu, Z.; Xie, J.; Ding, Z.; Yang, B.; Li, Z.; Liu, Z.; Li, Q.; Dong, X.; Chen, Z.; et al. MMBench-GUI: A Unified Hierarchical Evaluation Framework for Multi-Platform GUI Agents. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026, pp. 6239–6248.
37. Pasupat, P.; Jiang, T.S.; Liu, E.; Guu, K.; Liang, P. Mapping natural language commands to web elements. In Proceedings of the Proceedings of the 2018 conference on empirical methods in natural language processing, 2018, pp. 4970–4976.
38. Li, Y.; He, J.; Zhou, X.; Zhang, Y.; Baldridge, J. Mapping natural language instructions to mobile UI action sequences. In Proceedings of the Proceedings of the 58th annual meeting of the association for computational linguistics, 2020, pp. 8198–8210.
39. Lu, Y.; Yang, J.; Shen, Y.; Awadallah, A. Omniparser for pure vision based gui agent. *arXiv preprint arXiv:2408.00203* **2024**.
40. 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. *Advances in Neural Information Processing Systems* **2026**, *38*.
41. Ji, F.; Yang, J.; Song, Z.; Wang, Y.; Cui, Z.; Li, Y.; Jiang, Q.; Chen, X. FineState-Bench: Benchmarking State-Conditioned Grounding for Fine-grained GUI State Setting. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2026, 2026, pp. 43073–43088.
42. Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; Zhou, J. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond, 2023, [arXiv:cs.CV/2308.12966].
43. OpenAI. GPT-6 Astra: A New Generation of Intelligence, 2026. Accessed: Oct. 2, 2026.
44. Anthropic. Developing a Computer Use Model, 2024. Accessed: Sep. 19, 2026.
45. Deitke, M.; Clark, C.; Lee, S.; Tripathi, R.; Yang, Y.; Park, J.S.; Salehi, M.; Muennighoff, N.; Lo, K.; Soldaini, L.; et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2025, pp. 91–104.

46. 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**.
47. Wang, W.; Gao, Z.; Gu, L.; Pu, H.; Cui, L.; Wei, X.; Liu, Z.; Jing, L.; Ye, S.; Shao, J.; et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265* **2025**.
48. Hong, W.; Yu, W.; Gu, X.; Wang, G.; Gan, G.; Tang, H.; Cheng, J.; Qi, J.; Ji, J.; Pan, L.; et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006* **2025**.
49. Guo, D.; Wu, F.; Zhu, F.; Leng, F.; Shi, G.; Chen, H.; Fan, H.; Wang, J.; Jiang, J.; Wang, J.; et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062* **2025**.
50. Team, K.; Du, A.; Yin, B.; Xing, B.; Qu, B.; Wang, B.; Chen, C.; Zhang, C.; Du, C.; Wei, C.; et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491* **2025**.
51. Yue, Z.; Lin, Z.; Song, Y.; Wang, W.; Ren, S.; Gu, S.; Li, S.; Li, P.; Zhao, L.; Li, L.; et al. Mimo-vl technical report. *arXiv preprint arXiv:2506.03569* **2025**.
52. Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261* **2025**.
53. Clark, C.; Zhang, J.; Ma, Z.; Park, J.S.; Tripathi, R.; Lee, S.; Salehi, M.; Ren, J.; Kim, C.D.; Yang, Y.; et al. Molmo2: Open weights and data for vision-language models with video understanding and grounding. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026, pp. 28652–28668.
54. Qwen Team. Qwen3.5: Towards Native Multimodal Agents, 2026.
55. Huang, A.; Yao, C.; Han, C.; Wan, F.; Guo, H.; Lv, H.; Zhou, H.; Wang, J.; Zhou, J.; Sun, J.; et al. Step3-vl-10b technical report. *arXiv preprint arXiv:2601.09668* **2026**.
56. Bai, C.; Zang, X.; Xu, Y.; Sunkara, S.; Rastogi, A.; Chen, J.; et al. Uibert: Learning generic multimodal representations for ui understanding. *arXiv preprint arXiv:2107.13731* **2021**.
57. Hong, W.; Wang, W.; Lv, Q.; Xu, J.; Yu, W.; Ji, J.; Wang, Y.; Wang, Z.; Dong, Y.; Ding, M.; et al. Cogagent: A visual language model for gui agents. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2024, pp. 14281–14290.
58. Qin, Y.; Ye, Y.; Fang, J.; Wang, H.; Liang, S.; Tian, S.; Zhang, J.; Li, J.; Li, Y.; Huang, S.; et al. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326* **2025**.
59. Tang, L.; Dong, S.; Huang, Y.; Xiang, M.; Ruan, H.; Wang, B.; Li, S.; Xi, Z.; Cao, Z.; Pang, H.; et al. Magicgui: A foundational mobile gui agent with scalable data pipeline and reinforcement fine-tuning. *arXiv preprint arXiv:2508.03700* **2025**.
60. Huang, Z.; Cheng, Z.; Pan, J.; Hou, Z.; Zhan, M. Spiritsight agent: Advanced gui agent with one look. In Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2025, pp. 29490–29500.
61. Yang, J.; Tan, R.; Wu, Q.; Zheng, R.; Peng, B.; Liang, Y.; Gu, Y.; Cai, M.; Ye, S.; Jang, J.; et al. Magma: A foundation model for multimodal ai agents. In Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2025, pp. 14203–14214.
62. Xu, Y.; Wang, Z.; Wang, J.; Lu, D.; Xie, T.; Saha, A.; Sahoo, D.; Yu, T.; Xiong, C. Aguvis: Unified pure vision agents for autonomous gui interaction. *arXiv preprint arXiv:2412.04454* **2024**.
63. Zhang, J.; Yu, Y.Q.; Liao, M.; Li, W.; Wu, J.; Wei, Z. UI-hawk: Unleashing the screen stream understanding for mobile GUI agents. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, 2025, pp. 18217–18236.
64. Li, H.; Su, J.; Chen, J.; Ju, Z.; Chen, Y.; Li, Q.; Zhang, Z. UIPro: Unleashing Superior Interaction Capability For GUI Agents. In Proceedings of the 2025 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2025, pp. 1613–1623.
65. Fan, S.; Chen, L.; Yin, R.; Liu, Q.; Rao, Y.; Guo, M.H.; Hu, S.M. GUICrafter: Weakly-Supervised GUI Agent Leveraging Massive Unannotated Screenshots. *arXiv preprint arXiv:2606.29705* **2026**.
66. Zhang, M.; Zhao, X.; Tan, Z.; Huoshen, Z.; Fan, Y.; Yang, Y.; Qiu, K.; Liu, B.; Wagle, J.; Yin, C.; et al. Covering Human Action Space for Computer Use: Data Synthesis and Benchmark. *arXiv preprint arXiv:2605.12501* **2026**.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

67. Wang, Y.; Hu, H.; Han, J.; Qiu, W.; Ge, Z.; Tang, L.; Wang, F. StepX-Edge: An On-Device UI Vision-Language Model via Architecture-Training-Deployment Co-Design. *arXiv preprint arXiv:2607.22708* **2026**.
68. Banerjee, P.; Mahajan, S.; Arora, K.; Baral, C.; Riva, O. Lexi: Self-supervised learning of the ui language. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022, 2022, pp. 6992–7007.
69. Lei, W.; Gao, D.; Shou, M.Z. Grounding multimodal large language model in gui world. In Proceedings of the International Conference on Learning Representations, 2025, Vol. 2025, pp. 19742–19765.
70. Baechler, G.; Sunkara, S.; Wang, M.; Zubach, F.; Mansoor, H.; Etter, V.; Cărbune, V.; Lin, J.; Chen, J.; Sharma, A. Screenai: A vision-language model for ui and infographics understanding. *arXiv preprint arXiv:2402.04615* **2024**.
71. Li, G.; Li, Y. Spotlight: Mobile ui understanding using vision-language models with a focus. *arXiv preprint arXiv:2209.14927* **2022**.
72. Zhang, Z.; Xie, W.; Zhang, X.; Lu, Y. Reinforced ui instruction grounding: Towards a generic ui task automation api. *arXiv preprint arXiv:2310.04716* **2023**.
73. Shaw, P.; Joshi, M.; Cohan, J.; Berant, J.; Pasupat, P.; Hu, H.; Khandelwal, U.; Lee, K.; Toutanova, K.N. From pixels to ui actions: Learning to follow instructions via graphical user interfaces. *Advances in Neural Information Processing Systems* **2023**, *36*, 34354–34370.
74. You, K.; Zhang, H.; Schoop, E.; Weers, F.; Swearngin, A.; Nichols, J.; Yang, Y.; Gan, Z. Ferret-ui: Grounded mobile ui understanding with multimodal llms. In Proceedings of the European Conference on Computer Vision. Springer, 2024, pp. 240–255.
75. Li, Z.; You, K.; Zhang, H.; Feng, D.; Agrawal, H.; Li, X.; Moorthy, M.; Nichols, J.; Yang, Y.; Gan, Z. Ferret-ui 2: Mastering universal user interface understanding across platforms. In Proceedings of the International Conference on Learning Representations, 2025, Vol. 2025, pp. 99783–99806.
76. Chen, X.; Li, H.; Liang, J.; Jiang, S.; Yang, D. Edge: Enhanced grounded gui understanding with enriched multi-granularity synthetic data. *arXiv preprint arXiv:2410.19461* **2024**.
77. Wang, Y.; Zhang, H.; Tian, J.; Tang, Y. Ponder & press: Advancing visual gui agent towards general computer control. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 1461–1473.
78. Zhang, M.; Xu, Z.; Zhu, J.; Dai, Q.; Qiu, K.; Yang, Y.; Luo, C.; Chen, T.; Wagle, J.; Franklin, T.; et al. Phi-ground tech report: Advancing perception in gui grounding. *arXiv preprint arXiv:2507.23779* **2025**.
79. Chen, L.; Zhou, H.; Zhang, J.; Tong, P.; Zhang, X.; Kong, Q.; Liu, C.; Liu, Y.; Wang, W.; Wang, Y.; et al. UI-Ins: Enhancing GUI grounding with multi-perspective instruction as reasoning. In Proceedings of the International Conference on Learning Representations, 2026, Vol. 2026, pp. 54418–54444.
80. Wang, S.; Zhang, T.; Masry, A.; Pal, C.; Gella, S.; Liu, B.; Taslakian, P. Improving GUI Grounding with Explicit Position-to-Coordinate Mapping. *arXiv preprint arXiv:2510.03230* **2025**.
81. Tang, F.; Shen, Y.; Zhang, H.; Chen, S.; Hou, G.; Zhang, W.; Zhang, W.; Song, K.; Lu, W.; Zhuang, Y. Think twice, click once: Enhancing gui grounding via fast and slow systems. *arXiv preprint arXiv:2503.06470* **2025**.
82. Park, J.; Tang, P.; Das, S.; Appalaraju, S.; Singh, K.Y.; Manmatha, R.; Ghadar, S. R-vlm: Region-aware vision language model for precise gui grounding. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 9669–9685.
83. Hu, S.; Lin, K.Q.; Shou, M.Z. ShowUI-p: Flow-based Generative Models as GUI Dexterous Hands. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026, pp. 8130–8140.
84. Lin, K.Q.; Li, L.; Gao, D.; Yang, Z.; Wu, S.; Bai, Z.; Lei, S.W.; Wang, L.; Shou, M.Z. Showui: One vision-language-action model for gui visual agent. In Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2025, pp. 19498–19508.
85. Pawlowski, P.; Zawistowski, K.; Lapacz, W.; Wiacek, A.; Skorupa, M.; Postansque, S.; Hoscilowicz, J. Tinyclick: Single-turn agent for empowering gui automation. *arXiv preprint arXiv:2410.11871* **2024**.
86. Liu, X.; Zhang, X.; Zhang, Z.; Lu, Y. Ui-e2i-synth: Advancing gui grounding with large-scale instruction synthesis. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 15668–15684.
87. Wang, X.; Wang, B.; Lu, D.; Yang, J.; Xie, T.; Wang, J.; Deng, J.; Guo, X.; Xu, Y.; Wu, C.; et al. Opencua: Open foundations for computer-use agents. *Advances in Neural Information Processing Systems* **2026**, *38*, 139756–139806.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

88. Chen, J.; Chen, L.; Wang, D.; Su, Q.; Chu, Z.; Hao, B.; Gan, L.; Zhuang, C.; Gu, J. V2P: Visual Attention Calibration for GUI Grounding via Background Suppression and Center Peaking. *arXiv preprint arXiv:2508.13634* **2025**.
89. Yang, R.; Wu, Q.; Wang, Z.; Chen, H.; Yang, K.; Cheng, H.; Yao, H.; Peng, B.; Zhang, H.; Gao, J.; et al. Gui-libra: Training native gui agents to reason and act with action-aware supervision and partially verifiable rl. *arXiv preprint arXiv:2602.22190* **2026**.
90. Kumbhar, S.; Liao, H.; Appalaraju, S.; Singh, K.Y. Towards GUI Agents: Vision-Language Diffusion Models for GUI Grounding. *arXiv preprint arXiv:2603.26211* **2026**.
91. Gu, Z.; Chen, H.; Qin, Q.; Xin, Y.; Gan, K.; Liu, L.; Cui, L.; Wang, X.; Zhou, B.; Zhang, Y.; et al. LLaDA-UI: Bringing Block-wise Diffusion to Vision-Language GUI Agents. *arXiv preprint arXiv:2609.13287* **2026**.
92. Jiang, Q.; Huo, J.; Chen, X.; Xiong, Y.; Zeng, Z.; Chen, Y.; Ren, T.; Yu, J.; Zhang, L. Detect anything via next point prediction. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026, pp. 25472–25483.
93. Kim, S.; Chen, Z.; Li, X.; Liu, Q. Look Where You're Told: Instruction-Consistent Attention for GUI Grounding. In Proceedings of the Proceedings of the 4th Workshop on Advances in Language and Vision Research (ALVR), 2026, pp. 155–163.
94. Clark, C.; Yang, Y.; Park, J.S.; Ma, Z.; Zhang, J.; Tripathi, R.; Salehi, M.; Lee, S.; Anderson, T.; Han, W.; et al. MolmoPoint: Better pointing for VLMs with grounding tokens. In Proceedings of the European Conference on Computer Vision. Springer, 2026, pp. 665–684.
95. Zhou, S.; Lai, V.D.; Tan, H.; Kil, J.; Zhu, W.; Chen, C.; Zhang, R. Gui-aima: Aligning intrinsic multimodal attention with a context anchor for gui grounding. In Proceedings of the European Conference on Computer Vision. Springer, 2026, pp. 79–97.
96. Feizi, A.; Nayak, S.; Jian, X.; Lin, K.Q.; Li, K.; Awal, R.; Lù, X.H.; Obando Ceron, J.S.; Rodriguez, J.A.; Chapados, N.; et al. Grounding computer use agents on human demonstrations. In Proceedings of the International Conference on Learning Representations, 2026, Vol. 2026, pp. 5181–5215.
97. Zhao, Z.; Liu, Y.; Liu, Y.; Wang, H.; Tian, L.; Zhou, X.; You, Y.; Yu, Z.; Yu, Y.; Zhou, J. POINTS-GUI-G: GUI-Grounding Journey. *arXiv preprint arXiv:2602.06391* **2026**.
98. Ouyang, M.; Lin, K.Q.; Shou, M.Z.; Ng, H.T. Focusui: Efficient ui grounding via position-preserving visual token selection. *arXiv preprint arXiv:2601.03928* **2026**.
99. Liu, Y.; Li, P.; Xie, C.; Hu, X.; Han, X.; Zhang, S.; Yang, H.; Wu, F. Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative reasoners. *arXiv preprint arXiv:2504.14239* **2025**.
100. Fan, L.; Bi, X.; Li, Y.; Li, S.; Li, H.; Yang, M. RankGround: Efficient High-Resolution GUI Grounding via Lightweight Reranker-Guided Crop Selection. *arXiv preprint arXiv:2609.18690* **2026**.
101. Liu, C.; Chen, L.; Zhou, H.; Chen, L.; Cai, C.; Yu, X.; Hoi, S.; Wang, Y. One Forward Beats Two: InnerZoom for Accurate and Efficient GUI Grounding. *arXiv preprint arXiv:2606.30084* **2026**.
102. Wang, X.; Zhang, G.; Li, J.; Tu, J.; Li, C.; Li, M. Tooltok: Tool tokenization for efficient and generalizable gui agents. *arXiv preprint arXiv:2602.02548* **2026**.
103. Yang, Z.; Dou, Z.Y.; Feng, D.; Huang, F.; Nguyen, A.; You, K.; Attia, O.; Yang, Y.; Feng, M.; Zhang, H.; et al. Ferret-ui lite: Lessons from building small on-device gui agents. *arXiv preprint arXiv:2509.26539* **2025**.
104. Jing, H.; Chen, J.; Rao, C.; Dang, Z.; Teng, J.; Chu, T.; Mo, J.; Fang, S.; Lin, H.; Lv, R.; et al. Sparkui-parser: Enhancing gui perception with robust grounding and parsing. *arXiv preprint arXiv:2509.04908* **2025**.
105. Wang, W.; Huang, Q.; Xue, G.; Liang, X.; Li, J. Learning active perception via self-evolving preference optimization for gui grounding. *arXiv preprint arXiv:2509.04243* **2025**.
106. Luo, R.; Wang, L.; He, W.; Chen, L.; Li, J.; Xia, X. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458* **2025**.
107. Yuan, X.; Zhang, J.; Li, K.; Cai, Z.; Yao, L.; Chen, J.; Wang, E.; Hou, Q.; Chen, J.; Jiang, P.T.; et al. Enhancing visual grounding for gui agents via self-evolutionary reinforcement learning. *arXiv preprint arXiv:2505.12370* **2025**.
108. Zhou, Y.; Dai, S.; Wang, S.; Zhou, K.; Jia, Q.; Xu, J. Gui-g1: Understanding r1-zero-like training for visual grounding in gui agents. *Advances in Neural Information Processing Systems* **2026**, *38*, 95683–95705.
109. Zhang, S.; Zhang, R.; Fu, P.; Wang, S.; Yang, J.; Du, X.; Qin, B.; Huang, Y.; Luo, Z.; Luan, J. Btl-ui: Blink-think-link reasoning model for gui agent. *Advances in Neural Information Processing Systems* **2026**, *38*, 56035–56056.
110. Zhang, Z.; Lu, Y.; Fu, Y.; Huo, Y.; Yang, S.; Wu, Y.; Si, H.; Cong, X.; Chen, H.; Lin, Y.; et al. Agentcpm-gui: Building mobile-use agents with reinforcement fine-tuning. In Proceedings of the Proceedings of the 2025

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2025, pp. 155–180.
111. Du, Y.; Yan, Y.; Tang, F.; Lu, Z.; Zong, C.; Lu, W.; Jiang, S.; Shen, Y. Test-time reinforcement learning for gui grounding via region consistency. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2026, Vol. 40, pp. 30593–30601.
112. Yang, Y.; Li, D.; Dai, Y.; Yang, Y.; Luo, Z.; Zhao, Z.; Hu, Z.; Huang, J.; Saha, A.; Chen, Z.; et al. Gta1: Gui test-time scaling agent. In Proceedings of the International Conference on Learning Representations, 2026, Vol. 2026, pp. 48295–48323.
113. Wang, H.; Zou, H.; Song, H.; Feng, J.; Fang, J.; Lu, J.; Liu, L.; Luo, Q.; Liang, S.; Huang, S.; et al. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544* **2025**.
114. Tang, F.; Gu, Z.; Lu, Z.; Liu, X.; Shen, S.; Meng, C.; Wang, W.; Zhang, W.; Shen, Y.; Lu, W.; et al. GUI-G^2: Gaussian Reward Modeling for GUI Grounding. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2026, Vol. 40, pp. 33214–33222.
115. Wu, J.; Yang, C.; Shen, Y.; Xu, F.; Ni, B.; Liao, C.; Liu, Y.; Wang, H.; Nie, S.; Zhang, S.; et al. SSL: Sweet Spot Learning for Differentiated Guidance in Agentic Optimization. *arXiv preprint arXiv:2601.22491* **2026**.
116. Tang, J.; Xia, Y.; Wu, Y.F.; Hu, Y.; Yuhui, C.; Chen, Q.G.; Xu, X.; Wu, X.; Lu, H.; Ma, Y.; et al. Lpo: Towards accurate gui agent interaction via location preference optimization. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2026, 2026, pp. 14617–14628.
117. Zeng, X.; Li, W.; Wu, Q.; Zhang, L. Fdc-ground: Improving grpo for gui grounding via exponential rewards and fact-aligned pruning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2026, Vol. 40, pp. 28122–28130.
118. Lian, S.; Wu, Y.; Ma, J.; Ding, Y.; Song, Z.; Chen, B.; Zheng, X.; Li, H.; Ji, R. Ui-agile: Advancing gui agents with effective reinforcement learning and precise inference-time grounding. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026, pp. 8575–8584.
119. Li, J.; Hao, C.; Chau, L.P.; Wang, Y. GUI-C^2: Coarse-to-Fine GUI Grounding via Difficulty-Aware Reinforcement Learning, 2026, [arXiv:cs.CV/2605.30884].
120. Liu, Y.; Liu, Z.; Zhu, S.; Li, P.; Xie, C.; Wang, J.; Hu, X.; Han, X.; Yuan, J.; Wang, X.; et al. Infigui-g1: Advancing gui grounding with adaptive exploration policy optimization. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2026, Vol. 40, pp. 32267–32275.
121. Kang, W.; Lei, B.; Liu, G.; Ding, C.; Yan, Y. Guirlvg: Incentivize gui visual grounding via empirical exploration on reinforcement learning. In Proceedings of the International Conference on Learning Representations, 2026, Vol. 2026, pp. 63861–63875.
122. Chen, C.; Shao, J.; Lu, D.; Hu, H.; Liu, X.; Yao, H.; Liu, W. Gui-eyes: Tool-augmented perception for visual grounding in gui agents. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2026, Vol. 40, pp. 29350–29358.
123. 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**.
124. Ye, X.; Li, Y.; Dai, W.; Liu, M.; Chen, Z.; Han, Z.; Min, H.; Ren, J.; Zhang, X.; Yang, W.; et al. Gui-arp: Enhancing grounding with adaptive region perception for gui agents. In Proceedings of the ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2026, pp. 3676–3680.
125. Wang, W.; Li, X.; Guo, H.; Yu, W.; Fang, T.; Mi, H.; Yu, D.; Zhang, S. Measure Twice, Click Once: Co-evolving Proposer and Visual Critic via Reinforcement Learning for GUI Grounding. In Proceedings of the Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2026, pp. 20964–20984.
126. Zhang, S.; Fu, P.; Zhang, R.; Yang, J.; Du, A.; Xi, X.; Wang, S.; Huang, Y.; Qin, B.; Luo, Z.; et al. Enhancing Trustworthy GUI Grounding via Self-Critiqued Reinforcement Learning. *arXiv preprint arXiv:2510.27266* **2025**.
127. Zhao, Y.; Chen, W.N.; Inan, H.A.; Kessler, S.; Wang, L.; Wutschitz, L.; Yang, F.; Zhang, C.; Minervini, P.; Rajmohan, S.; et al. Learning gui grounding with spatial reasoning from visual feedback. *arXiv preprint arXiv:2509.21552* **2025**.
128. Wei, J.; Bai, X.; Liu, S.; Jia, C.; Sun, Z.; Xu, X.; Li, S.; Sun, L.; Yu, B.; He, C.; et al. PAGER: Bridging the Semantic-Execution Gap in Point-Precise Geometric GUI Control. *arXiv preprint arXiv:2605.15963* **2026**.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

129. Liu, Y.; Tang, F.; Yan, Y.; Lu, Z.; Nong, S.; Jiang, T.; Xu, W.; Zhang, W.; Lu, W.; Xiao, J.; et al. Learning from Reliable Negatives: Confidence-Anchored Test-Time Adaptation for GUI Grounding. In Proceedings of the European Conference on Computer Vision. Springer, 2026, pp. 227–244.
130. Lei, B.; Xu, N.; Payani, A.; Hong, M.; Liao, C.; Cao, Y.; Ding, C. GUI-Spotlight: Adaptive Iterative Focus Refinement for Enhanced GUI Visual Grounding. *arXiv preprint arXiv:2510.04039* **2025**.
131. Gao, L.; Zhang, L.; Gao, P.; Liu, W.; Luan, J.; Xu, M. Gui-shift: Enhancing vlm-based gui agents through self-supervised reinforcement learning. In Proceedings of the International Conference on Learning Representations, 2026, Vol. 2026, pp. 39236–39255.
132. Yang, S.; Xiong, Z.; Yang, K.; Cai, G.; Li, Y.; Jiang, P.; Kun, G.; Cai, Q.; Feng, L. SCA: Spatial Credit Assignment for Reinforcement Learning of GUI Agents. *arXiv preprint arXiv:2609.36939* **2026**.
133. Qiu, X.; Zhang, Y.; Jia, H.; Shen, S.; Meng, C.; Zhu, L. VISTA: View-Consistent Self-Verified Training for GUI Grounding. *arXiv preprint arXiv:2606.14579* **2026**.
134. Song, Y.; Yin, J.; Zhang, R.; Harris, I.G. Compress to Focus: Efficient Coordinate Compression for Policy Optimization in Multi-Turn GUI Agents. *arXiv preprint arXiv:2601.11631* **2026**.
135. Lee, H.; Kim, J.; Kim, B.; Tack, J.; Jo, C.; Lee, J.; Park, C.; In, S.; Shin, J.; Yoo, K.M. ReGUIDE: Data Efficient GUI Grounding via Spatial Reasoning and Search. *arXiv preprint arXiv:2505.15259* **2025**.
136. Lu, Z.; Chai, Y.; Guo, Y.; Yin, X.; Liu, L.; Wang, H.; Xiao, H.; Ren, S.; Zhao, P.; Liu, G.; et al. Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2026, Vol. 40, pp. 17608–17616.
137. Li, M.; Mu, L.; Hu, H.; Gu, Y.; Xue, L.; Tang, L.; Zhang, Y.; Wang, F. CA-OPD: Confidence-Aware On-Policy Distillation for Structured Visual Prediction. *arXiv preprint arXiv:2609.02401* **2026**.
138. Lian, N.; Chen, A.; Yu, Z.; Duan, C.; Liu, F.; Liu, H.; Fu, P.; Luan, J.; Wang, Y.; Xia, S.T.; et al. UI-MOPD: Multi-Platform On-Policy Distillation for Continual GUI Agent Learning. *arXiv preprint arXiv:2607.04425* **2026**.
139. Xuan, S.; Li, Z. Test-Time Self-Evolving GUI Visual Grounding via Reflection-Guided On-Policy Self-Distillation. *arXiv preprint arXiv:2608.11191* **2026**.
140. Team, V.; Cai, Z.; Chen, H.; Chen, J.; Chen, W.; Gao, C.; Gu, Z.; Guo, Y.; Hu, Y.; Jiang, J.; et al. UI-Venus-2 Technical Report. *arXiv preprint arXiv:2609.00028* **2026**.
141. Zhang, Y.; Wu, D.; Shen, H.; Ma, C.; Zhou, Y. Learn where to click from yourself: On-policy self-distillation for gui grounding. *arXiv preprint arXiv:2605.00642* **2026**.
142. Wu, Y.; Cai, Z.; Ning, L.; Wang, H.; Chen, Z.; Tang, Y.; Chen, H. LiteGUI: Distilling Compact GUI Agents with Reinforcement Learning. *arXiv preprint arXiv:2605.07505* **2026**.
143. Xu, H.M.; Chen, Q.; Wang, L.; Liu, L. Attention-driven gui grounding: Leveraging pretrained multimodal large language models without fine-tuning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025, Vol. 39, pp. 8851–8859.
144. Zhang, Y.; Pan, Z.; Zeng, Z.; Shen, S.; Meng, C.; Zhu, L. Mvp: Multiple view prediction improves gui grounding. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026, pp. 27482–27492.
145. Lee, J.; Choi, Y.; Choi, H.; Kim, H.; Kim, S. A training-free, task-agnostic framework for enhancing mllm performance on high-resolution images. *arXiv preprint arXiv:2507.10202* **2025**.
146. 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**.
147. Li, A.Y.; Yu, B.; Lei, D.; Ren, T.; Liu, S. Chain-of-Ground: Improving GUI Grounding via Iterative Reasoning and Reference Feedback. *arXiv preprint arXiv:2512.01979* **2025**.
148. Wu, H.; Chen, H.; Cai, Y.; Liu, C.; Ye, Q.; Yang, M.H.; Wang, Y. Dimo-gui: Advancing test-time scaling in gui grounding via modality-aware visual reasoning. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, 2025, pp. 26246–26256.
149. Lin, J.; Shen, F.; Li, J.; Nie, P.; Yu, F.; Li, M.; Li, H. What Happens Before Decoding? Prefill Determines GUI Grounding in VLMs. *arXiv preprint arXiv:2605.12549* **2026**.
150. Ma, L.; Zhao, D.; Wang, S.; Lv, Z.; Wang, M. Trifuse: Enhancing attention-based gui grounding via multimodal fusion. *arXiv preprint arXiv:2602.06351* **2026**.
151. Fu, Z.; Wang, S.; Zhang, W.; Li, G.; Deng, Y.; Gao, J.; Qi, J.; Yan, H.; Zheng, Y.; Li, X.; et al. GUI-Lens: Coarse-to-Fine Cropping for GUI Grounding with General-Purpose VLMs. *arXiv preprint arXiv:2608.03270* **2026**.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

152. Mittal, H.; Mittal, G.; Troncoso, N.D.; Hu, Y. PrecisionCUA: Iterative Visual Refinement for Pixel-Precise Cursor Grounding in Code Editors. *arXiv preprint arXiv:2604.13019* **2026**.
153. Tao, X.; Wang, Y.; Cai, Y.; Luo, Y.; Han, K.; Tang, J. Mitigating Coordinate Prediction Bias from Positional Encoding Failures. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2026, 2026, pp. 20635–20650.
154. 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**.
155. Li, Y.; Zhu, T.; Son, H.M.; Zhao, Z.; Liu, X.; Chen, M. AQuaUI: Visual Token Reduction for GUI Agents with Adaptive Quadtrees. *arXiv preprint arXiv:2605.19260* **2026**.
156. Zhang, B.; Zhang, B.; Wang, B.; Zheng, W.; Cheng, Y.; Tang, L.; Yan, Y.; Zhou, J.; Lu, J. BAMI: Training-Free Bias Mitigation in GUI Grounding. *arXiv preprint arXiv:2605.06664* **2026**.
157. Liu, Y.; Shen, H.; Yu, L.; Liu, S.; Chen, Z.; Zhou, Y. DRS-GUI: Dynamic Region Search for Training-Free GUI Grounding. *arXiv preprint arXiv:2605.15542* **2026**.
158. Li, Z.; Song, G.; Wang, Y.; Xiong, Z.; Yuan, J.; Cai, Y. Generalist Scanner Meets Specialist Locator: A Synergistic Coarse-to-Fine Framework for Robust GUI Grounding. *arXiv preprint arXiv:2509.24133* **2025**.
159. Kwak, S.; Kim, J.; Kim, B.; Yoon, J.J.; Jang, W.; Hong, J.; Yang, J.; Kwon, Y.D. MEGA-GUI: Multi-stage Enhanced Grounding Agents for GUI Elements. *arXiv preprint arXiv:2511.13087* **2025**.
160. Yang, Z.; Feng, X.; Zhong, W.; Liu, L.; Feng, X.; Qin, B. Focus Like a Human: Efficient GUI Grounding via Coarse-to-Fine Visual Attention and Parallel Verification. In Proceedings of the Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-26; Calvanese, D., Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2026, pp. 428–436. Main Track, https://doi.org/10.24963/ijcai.2026/49.
161. Liu, Z.; Feng, T.; Kang, B.; Yang, Y.; Luo, J. Zoom to essence: Trainless gui grounding by inferring upon interface elements. *arXiv preprint arXiv:2603.14448* **2026**.
162. Singh, K.; Singh, S.; Khanna, M. Trishul: Towards region identification and screen hierarchy understanding for large vlm based gui agents. In Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2025, pp. 170–179.
163. Yan, A.; Yang, Z.; Zhu, W.; Lin, K.; Li, L.; Wang, J.; Yang, J.; Zhong, Y.; McAuley, J.; Gao, J.; et al. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. *arXiv preprint arXiv:2311.07562* **2023**.
164. Zheng, B.; Gou, B.; Kil, J.; Sun, H.; Su, Y. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614* **2024**.
165. Zhang, Y.; Kang, L.; Wang, Z.; Wu, D.; Wang, Z. DeCoPatch: Revealing Causal Latent Subspaces in Vision-Language Models for GUI Grounding. In Proceedings of the European Conference on Computer Vision. Springer, 2026, pp. 20–37.
166. Li, Y.; Li, G.; Zhou, X.; Dehghani, M.; Gritsenko, A. Vut: Versatile ui transformer for multi-modal multi-task user interface modeling. *arXiv preprint arXiv:2112.05692* **2021**.
167. Xu, N.; Masling, S.; Du, M.; Campagna, G.; Heck, L.; Landay, J.; Lam, M. Grounding open-domain instructions to automate web support tasks. In Proceedings of the Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 1022–1032.
168. Zhao, Y.; Zhu, H.; Jiang, T.; Li, S.; Xu, X.; Wang, H.H. Co-EPG: A Framework for Co-Evolution of Planning and Grounding in Autonomous GUI Agents. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2026, Vol. 40, pp. 36582–36590.
169. Zhou, H.; Zhang, X.; Tong, P.; Zhang, J.; Chen, L.; Kong, Q.; Cai, C.; Liu, C.; Wang, Y.; Zhou, J.; et al. Mai-ui technical report: Real-world centric foundation gui agents. *arXiv preprint arXiv:2512.22047* **2025**.
170. Lee, J.; Khan, Z.; Prasad, A.; Chen, J.C.Y.; Chakraborty, S.; Balasubramaniam, K.; Sahu, S.; Stengel-Eskin, E.; Lee, H.; Bansal, M. A History-Aware Visually Grounded Critic for Computer Use Agents. *arXiv preprint arXiv:2606.11078* **2026**.
171. Li, S.; Guo, X.; Liu, T.; Yi, B.; Gong, Z.; Liu, Z.; Chen, H.; Zhang, W. What's missing in screen-to-action? towards a ui-in-the-loop paradigm for multimodal gui reasoning. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2026, 2026, pp. 17674–17690.
172. Ettifouri, E.H.; Espejel, J.L.; Minkova, L.; Dardouri, T.; Dahhane, W. Visual Grounding Methods for Efficient Interaction with Desktop Graphical User Interfaces. *arXiv preprint arXiv:2407.01558* **2024**.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

173. Jin, S.; Wang, L.; Zhang, Z. SE-GA: Memory-Augmented Self-Evolution for GUI Agents. *arXiv preprint arXiv:2605.16883* **2026**.
174. Mi, H.; Feng, Y.; Lu, W.; Wang, Y.; Li, J.; Cao, S.; Cui, H.; Tian, T.; Zhang, X.; Luo, H.; et al. D-Artemis: A Deliberative Cognitive Framework for Mobile GUI Multi-Agents. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2026, 2026, pp. 13905–13931.
175. Muryn, V.; Sumyk, M.; Hirna, M.; Garkot, S.; Shamrai, M. Screen2AX: Vision-Based Approach for Automatic macOS Accessibility Generation. *arXiv preprint arXiv:2507.16704* **2025**.
176. Wang, S.; Fu, P.; Zhang, R.; Zhang, S.; Xi, X.; Yang, J.; Qin, B.; Huang, Y.; Luo, Z.; Luan, J. GAIA: A data flywheel system for training GUI test-time scaling critic models. In Proceedings of the European Conference on Computer Vision. Springer, 2026, pp. 386–404.
177. Sun, Y.; Fu, P.; Zhang, S.; Du, A.; Xi, X.; Zhang, R.; Luo, Z.; Luan, J.; Zhang, C. Beyond Binary: Reframing GUI Critique as Continuous Semantic Alignment. *arXiv preprint arXiv:2605.14311* **2026**.
178. Wu, K.; Zheng, B.; Huang, W.; Wang, K. Recurrent Latent Visual Search for GUI Grounding. *arXiv preprint arXiv:2610.05185* **2026**.
179. Sarch, G.; Saha, S.; Khandelwal, N.; Jain, A.; Tarr, M.; Kumar, A.; Fragkiadaki, K. Grounded reinforcement learning for visual reasoning. *Advances in Neural Information Processing Systems* **2026**, *38*, 150977–151013.
180. Xu, Y.; Zhang, Y.; Liu, J.; Chen, J. Structuring GUI Elements through Vision Language Models: Towards Action Space Generation. *arXiv preprint arXiv:2508.16271* **2025**.
181. Chen, W.; Cui, J.; Hu, J.; Qin, Y.; Fang, J.; Zhao, Y.; Wang, C.; Liu, J.; Chen, G.; Huo, Y.; et al. Guicourse: From general vision language model to versatile gui agent. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 21936–21959.
182. Huang, J.; Huang, Z.; Shi, Y.; Yang, T.; Zhai, X.; Chu, W.; Liu, N. Trust the Right Teacher: Quality-Aware Self-Distillation for GUI Grounding. *arXiv preprint arXiv:2606.18101* **2026**.
183. Nguyen, A. Improved gui grounding via iterative narrowing. *arXiv preprint arXiv:2411.13591* **2024**.
184. Li, W.; Shao, Y.; Yang, J.; Lu, Y.; Zhong, L.; Wang, Y.; Yu, M.; Ruan, T.; Duan, M. How auxiliary reasoning unleashes gui grounding in vlms. *arXiv preprint arXiv:2509.11548* **2025**.
185. Tao, X.; Wang, Y.; Cai, Y.; Yang, Z.; Tang, J. Understanding gui agent localization biases through logit sharpness. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2025, 2025, pp. 23361–23374.
186. Li, Y.; Hou, X. Hallucination-Free GUI Grounding via Regression-Free Layout-Aware Matching. *arXiv preprint arXiv:2608.09654* **2026**.
187. Gao, L.; Zhang, L.; Wang, S.; Gao, P.; Liu, W.; Luan, J.; Wang, S.; Li, Y.; Xu, M. MobileViews: A Million-scale and Diverse Mobile GUI Dataset. *arXiv preprint arXiv:2409.14337* **2024**.
188. Li, H.; Chen, J.; Su, J.; Chen, Y.; Li, Q.; Zhang, Z. Autogui: Scaling gui grounding with automatic functionality annotations from llms. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 10323–10358.
189. Liu, J.; Ou, T.; Song, Y.; Qu, Y.; Lam, W.; Xiong, C.; Chen, W.; Neubig, G.; Yue, X. Harnessing webpage uis for text-rich visual understanding. In Proceedings of the International Conference on Learning Representations, 2025, Vol. 2025, pp. 17971–18005.
190. Xu, Y.; Lu, D.; Shen, Z.; Wang, J.; Wang, Z.; Mao, Y.; Xiong, C.; Yu, T. Agenttrek: Agent trajectory synthesis via guiding replay with web tutorials. In Proceedings of the International Conference on Learning Representations, 2025, Vol. 2025, pp. 79822–79843.
191. Liu, Z.; Xie, J.; Ding, Z.; Li, Z.; Yang, B.; Wu, Z.; Wang, X.; Sun, Q.; Liu, S.; Wang, W.; et al. Scalecua: Scaling open-source computer use agents with cross-platform data. In Proceedings of the International Conference on Learning Representations, 2026, Vol. 2026, pp. 62707–62766.
192. Guo, X.; Gao, D.; Shou, M.Z. Auto-explorer: Automated data collection for gui agent. *arXiv preprint arXiv:2511.06417* **2025**.
193. Mu, J.; Zhang, C.; Ni, C.; Wang, L.; Qiao, B.; Mathur, K.; Wu, Q.; Xie, Y.; Ma, X.; Zhou, M.; et al. GUI-360°: A Comprehensive Dataset and Benchmark for Computer-Using Agents, 2025, [arXiv:cs.AI/2511.04307].
194. Gurbuz, A.S.; Hong, S.; Nassar, A.; Pollefeys, M.; Staar, P. ScreenParse: Moving Beyond Sparse Grounding with Complete Screen Parsing Supervision. *arXiv preprint arXiv:2602.14276* **2026**.
195. Xie, Y.; Chen, S.; Xing, J.; Jiang, W.; Zhu, Z.; Wang, Y.; Bu, P.; Song, J.; Jiang, Y.; Zheng, B. SecAgent: Efficient Mobile GUI Agent with Semantic Context. *arXiv preprint arXiv:2603.08533* **2026**.
196. Gurbuz, A.S.; Nassar, A.; Hong, S.; Pollefeys, M.; Staar, P.W. DeskForge: Dense Supervision from Desktop Environments for Computer-Use Agents. *arXiv preprint arXiv:2610.02320* **2026**.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 4 November 2026 https://doi.org/10.20944/preprints202611.0431.v1

197. Deka, B.; Huang, Z.; Franzen, C.; Hibschman, J.; Afergan, D.; Li, Y.; Nichols, J.; Kumar, R. Rico: A mobile app dataset for building data-driven design applications. In Proceedings of the Proceedings of the 30th annual ACM symposium on user interface software and technology, 2017, pp. 845–854.
198. Li, Y.; Li, G.; He, L.; Zheng, J.; Li, H.; Guan, Z. Widget captioning: Generating natural language description for mobile user interface elements. In Proceedings of the Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), 2020, pp. 5495–5510.
199. Li, G.; Baechler, G.; Tragut, M.; Li, Y. Learning to denoise raw mobile UI layouts for improving datasets at scale. In Proceedings of the Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022, pp. 1–13.
200. Chai, Y.; Huang, S.; Niu, Y.; Xiao, H.; Liu, L.; Wang, G.; Zhang, D.; Ren, S.; Li, H. Amex: Android multi-annotation expo dataset for mobile gui agents. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 2138–2156.
201. Hui, Z.; Li, Y.; Zhao, D.; Banbury, C.; Chen, T.; Koishida, K. WinSpot: GUI grounding benchmark with multimodal large language models. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2025, pp. 1086–1096.
202. Zhou, B.; Huang, Z.; Guo, Y.; Gu, Z.; Xia, T.; Luo, Z.; Tang, F.; Kong, D.; Shang, Y.; Ou, S.; et al. VenusBench-GD: A Comprehensive Multi-Platform GUI Benchmark for Diverse Grounding Tasks. *arXiv preprint arXiv:2512.16501* **2025**.
203. Jandial, S.; Li, Y.; Wagle, J.; Koishida, K. Do GUI Grounders Truly Understand UI Elements? In Proceedings of the Findings of the Association for Computational Linguistics: EACL 2026, 2026, pp. 2772–2785.
204. Wang, Y.; Sikka, H.; Mathur, Y.; Zhou, T.; Nyachhyon, J.; Guruprasad, P. GUI-Perturbed: Domain Randomization Reveals Systematic Brittleness in GUI Grounding Models. *arXiv preprint arXiv:2604.14262* **2026**.
205. Zhao, H.; Chen, T.; Wang, Z. WinDeskGround: A Benchmark for Robust GUI Grounding in Complex Multi-Window Desktop Environments. *arXiv preprint arXiv:2605.16402* **2026**.
206. Shakeel, R.; Ali, A.R.M.; Mushtaq, M.; Saleem, T.J.; Ashraf, T. Medspot: A workflow-aware sequential grounding benchmark for clinical gui. In Proceedings of the European Conference on Computer Vision. Springer, 2026, pp. 240–257.
207. Li, T.; Li, G.; Zheng, J.; Wang, P.; Li, Y. Mug: Interactive multimodal grounding on user interfaces. In Proceedings of the Findings of the Association for Computational Linguistics: EACL 2024, 2024, pp. 231–251.
208. Li, H.; Wang, X.; Su, J.; Ju, Z.; Chen, Y.; Li, Q.; Zhang, Z. AutoGUI-v2: A Comprehensive Multi-Modal GUI Functionality Understanding Benchmark. *arXiv preprint arXiv:2604.24441* **2026**.
209. Lu, Y.; Yu, Q.; Wang, H.; Liu, Z.; Su, W.; Liu, Y.; Guo, Y.; Liang, M.; Wang, Y.; Wang, H. TransBench: Breaking barriers for transferable graphical user interface agents in dynamic digital environments. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 12464–12478.
210. Li, C.; Sun, X.; Deng, Y.; Zhao, Q.; Yeung-Levy, S.; Zhang, Y. ScreenHaystack: Finding Blind Zones in GUI Grounding. *arXiv preprint arXiv:2609.32036* **2026**.
211. Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. Qwen2.5-VL Technical Report, 2025, [arXiv:cs.CV/2502.13923].
212. Google DeepMind. Model Evaluation—Approach, Methodology & Results: Gemini 3 Pro. Model evaluation report, 2025. Accessed: Oct. 2, 2026.
213. Team, V.; Gao, C.; Gu, Z.; Liu, Y.; Qiu, X.; Shen, S.; Wen, Y.; Xia, T.; Xu, Z.; Zeng, Z.; et al. Ui-venus-1.5 technical report. *arXiv preprint arXiv:2602.09082* **2026**.
214. Kumar, D.; Tayebati, S.; Naik, D.; Rios, A.S.; Ahuja, N.; Tickoo, O.; Krishnan, R.; Trivedi, A.R. Uncertainty Quantification for Computer-Use Agents: A Benchmark across Vision-Language Models and GUI Grounding Datasets. *arXiv preprint arXiv:2606.25760* **2026**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.