

TRISHUL: Towards Region Identification and Screen Hierarchy Understanding for Large VLM based GUI Agents

Kunal Singh*

Fractal AI Research

kunal.singh@fractal.ai

Shreyas Singh*

Fractal AI Research

shreyas.singh@fractal.ai

Mukund Khanna

Fractal AI Research

mukund.khanna@fractal.ai

Abstract

Recent advancements in Large Vision Language Models (LVLMs) have led to the emergence of LVLM-based Graphical User Interface (GUI) agents developed under various paradigms. Training-based approaches, such as CogAgent and SeeClick, suffer from poor cross-dataset and cross-platform generalization due to their reliance on dataset-specific training. Generalist LVLMs, such as GPT-4V, utilize Set-of-Marks (SoM) for action grounding; however, obtaining SoM labels requires metadata like HTML source, which is not consistently available across platforms. Additionally, existing methods often specialize in singular GUI tasks rather than achieving comprehensive GUI understanding. To address these limitations, we introduce TRISHUL, a novel, training-free agentic framework that enhances generalist LVLMs for holistic GUI comprehension. Unlike prior works that focus on either action grounding (mapping instructions to GUI elements) or GUI referring (describing GUI elements given a location), TRISHUL seamlessly integrates both. At its core, TRISHUL employs Hierarchical Screen Parsing (HSP) and the Spatially Enhanced Element Description (SEED) module, which work synergistically to provide multi-granular, spatially, and semantically enriched representations of GUI elements. Our results demonstrate TRISHUL's superior performance in action grounding across the ScreenSpot, VisualWebBench, AITW, and Mind2Web datasets. Additionally, for GUI referring, TRISHUL surpasses the ToL agent on the ScreenPR benchmark, setting a new standard for robust and adaptable GUI comprehension.

1. Introduction

Developing AI agents capable of operating digital devices through natural language commands has been a longstanding research goal [11, 26, 33]. These agents can enhance productivity by automating tasks through Graphical User Interface

(GUI). Early studies explored simplified settings [11, 26, 33], while later efforts [2, 5, 6, 14, 23, 23, 24, 35, 37, 45] leveraged GUI understanding to build more sophisticated agents. Recent approaches [8, 12, 34, 41, 47] incorporate LLMs alongside structured GUI representations (e.g., HTML, DOM trees, View Hierarchy) to enhance comprehension.

With advances in LVLMs, studies [8, 10, 13, 43, 46] have integrated visual perception to improve performance on benchmarks like Mind2Web [8] and WebArena [47]. However, these models struggle with visual grounding [40], relying heavily on structured metadata, which is often unavailable, noisy, or misaligned. SeeAct [46] improves action grounding in GPT-4V [29] via set-of-marks (SoM) [40], but its dependency on structured data introduces limitations.

1.1. Related Works & Motivation

Recent research has focused on developing agents that rely solely on visual perception to interact with GUIs in a human-like manner. These works on purely vision-based GUI agents using LVLMs have evolved along 2 main approaches:

End to End Training based GUI Agents: Multiple studies [3, 7, 15, 32, 42] have trained LVLMs on GUI navigation tasks for various platforms/device-types.

Test-time assistance with visual perception tools: Studies have leveraged visual perceptions tools to assist generalist LVLMs like GPT-4V. MM-Navigator [39] leverages pre-trained icon detector module. A concurrent work to ours, Omniparser [28], trains a YOLO-v8 [18] based icon detection & BLIPv2 [22] based icon captioner modules for action grounding. Tree-of-Lens (ToL) Agent [9] trains a perception module for GUI referring task of generating region description based on user selected point.

Multiple GUI navigation-related benchmarks [27, 38] and studies [7, 46] have highlighted two major weaknesses among pure vision-based GUI navigation agents. Firstly, the performance of these methods trained on certain distribution of user interfaces don't generalize well across platforms/device types. Given the rapid pace with which new user interfaces are introduced every day, the generalizability

*Equal contribution from the authors.

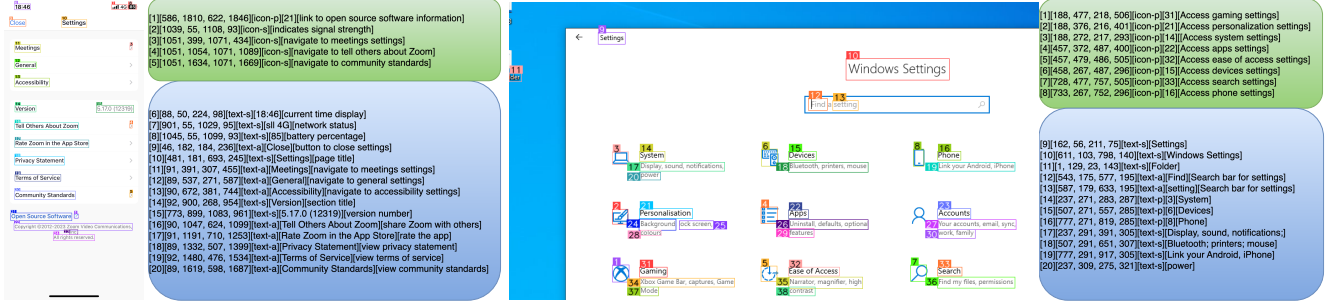


Figure 1. Screen parsing results showing detected GUI elements and their function descriptors leveraging our HSP and SEED modules

of training based approaches to Out-Of-Distribution samples remains a challenge. Secondly, most of the GUI agents such as DigiRL [3], SeeClick [7], MM-Navigator [39] are optimized for specialized GUI related tasks (majorly action prediction & grounding), and often evaluate on diversely sourced but thematically similar tasks and metrics, hence they lack proper GUI comprehension capabilities across different tasks and interfaces.

Algorithm 1 Hierarchical Screen Parsing

Require: Image I , $A_{\text{thresh-GROI}}$, $A_{\text{thresh-Icon}}$, IOU_{thresh} , SAM , OCR

- 1: Initialize: SAM , OCR , A_{thresh} , IOU_{thresh}
- 2: Sample N points $\mathcal{P} \leftarrow \mathcal{U}(0, W) \times \mathcal{U}(0, H)$ \triangleright Image Size (W, H)
- 3: $\mathcal{B} \leftarrow SAM(I, \mathcal{P})$, $\mathcal{T} \leftarrow OCR(I)$ \triangleright SAM boxes \mathcal{B} and OCR boxes \mathcal{T}
- 4: Initialize $\mathcal{G} \leftarrow \emptyset$, $\mathcal{I} \leftarrow \emptyset$ \triangleright GROI candidates and Icon candidates
- 5: **for** each $b \in \mathcal{B}$ **do**
- 6: **if** $\text{Area}(b) > A_{\text{thresh-GROI}}$ **then**
- 7: $\mathcal{G} \leftarrow \mathcal{G} \cup \{b\}$ \triangleright Add to GROI candidates
- 8: **end if**
- 9: **if** $\text{Area}(b) < A_{\text{thresh-Icon}}$ **then**
- 10: $\mathcal{I} \leftarrow \mathcal{I} \cup \{b\}$ \triangleright Add to Icon candidates
- 11: **end if**
- 12: **end for**
- 13: Initialize $\mathcal{S} \leftarrow \emptyset$ \triangleright Information Scores for Non Max Suppression (NMS)
- 14: $\mathcal{I}_{\text{filtered}}, \mathcal{T}_{\text{filtered}} \leftarrow \text{Overlap Removal and Filtering}(\mathcal{I}, \mathcal{T})$
- 15: **for** each $b \in \mathcal{G}$ **do**
- 16: $\mathcal{N}_{\text{inside}} = |\{\mathcal{T}_b^{\text{inside}}\}| + |\{\mathcal{I}_b^{\text{inside}}\}|$ \triangleright Number of boxes inside b
- 17: $\mathcal{N}_{\text{inter}} = |\{\mathcal{T}_b^{\text{intersect}}\}| + |\{\mathcal{I}_b^{\text{intersect}}\}|$ \triangleright Number of boxes intersecting b
- 18: $\mathcal{S} \leftarrow \mathcal{S} \cup \left\{ \frac{\mathcal{N}_{\text{inside}}}{\sqrt{1 + \mathcal{N}_{\text{inter}} \cdot \text{Area}(b)}} \right\}$ \triangleright Information Score for b
- 19: **end for**
- 20: $\mathcal{G}_{\text{filtered}} \leftarrow \text{NMS}(\mathcal{G}, \mathcal{S}, IOU_{\text{thresh}})$ \triangleright Apply NMS to get Filtered GROIs
- 21: **return** $\mathcal{G}_{\text{filtered}}, \mathcal{I}_{\text{filtered}}, \mathcal{T}_{\text{filtered}}$

1.2. Contribution

To address these challenges, we introduce TRISHUL, a training-free, agentic framework for comprehensive GUI screen understanding. TRISHUL equips LVLMs with the capabilities required to perform diverse GUI interaction tasks, it utilizes foundational models to parse and build a rich hierarchical understanding of the GUI screens, to enhance their action grounding and GUI referring capabilities.

Hierarchical Screen Parsing (HSP): The HSP module organizes GUI elements across two distinct levels of granularity: broad regions called Global Regions of Interest (GROIs) which cluster related components and local

elements like icons, text, and images. This hierarchical structuring captures spatial and semantic relationships between different GUI components, providing a multi-layered comprehensive GUI screen understanding.

Spatially Enhanced Element Description (SEED): As depicted in fig. 1, SEED generates contextually aware and spatially informed functionality descriptions for local elements by analyzing their relative positioning with respect to other elements in the GUI. By associating nearby icons and text, SEED enables the generation of high-fidelity functionality descriptions for GUI elements, facilitating a more nuanced understanding of each element’s role.

We evaluate TRISHUL on ScreenSpot [7], VisualWebBench [27], Mind2Web [8], and AITW [31], demonstrating that GPT-4V [29] and GPT-4o [30] using TRISHUL surpass prior state-of-the-art methods in action grounding and episodic instruction-following tasks. Additionally, we validate TRISHUL’s effectiveness in GUI referring via the Screen PR dataset, improving accessibility applications and user interaction feedback

2. Methodology

To overcome limitations of the existing methods mentioned in Sec. 1, we focus on designing screen understanding components, in a training free manner, to enhance the grounding and referring abilities of the generalist LVLMs leading to TRISHUL agent. We discuss the different modules below along with an end-to-end agentic framework resulting from leveraging these modules.

2.1. Hierarchical Screen Parsing

The hierarchical screen parsing process is formalized in Algorithm 1. Initially, the screen image I is passed through SAM [20] and EasyOCR [17]. The generated bounding boxes are filtered based on predefined area thresholds $A_{\text{thresh-GROI}}$ and $A_{\text{thresh-LE}}$ to generate GROI candidates and Local Elements (LE). Local Elements collectively refer to bounding boxes for text, icon, buttons and images in the GUI. We then apply an overlap removal and filtering func-

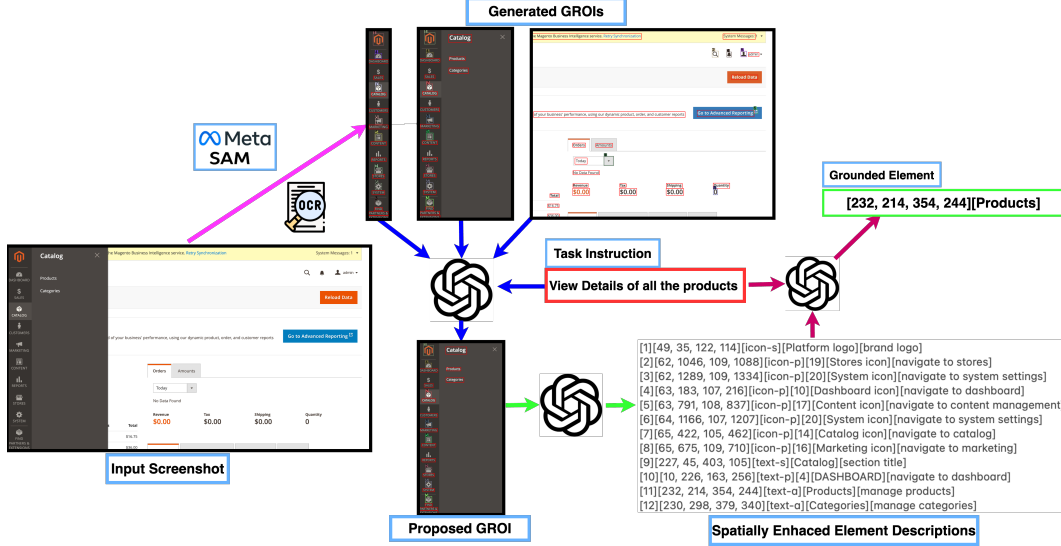


Figure 2. TRISHUL: Agentic Action Grounding Framework, Pink arrow, denotes our Hierarchical Screen Parsing (HSP) method, to generate GROIs and local element annotations, Green arrows represent our Spatially Enhanced Element Descriptor (SEED) workflow, Blue arrows represent our GROI proposal framework and Magenta Arrow shows, the Set of Marks (SoM) based Grounding workflow.

tion to refine the icon and text bounding boxes by removing redundant and unwanted local elements.

For each GROI candidate, the number of boxes inside and intersecting with the GROI is calculated. An Information Score \mathcal{S} is then computed for each candidate based on the ratio of the number of bounding boxes inside, to the area of the GROI, adjusted by the number of intersecting boxes. This score provides a measure of the GROI’s information content, helping the system to prioritize larger and more informative regions for inclusion in the hierarchical tree.

Finally, a Non-Max-Suppression (NMS) algorithm is applied to the GROI candidates based on their Information Scores. The resulting filtered set of GROIs, icons, and text boxes are returned as the final hierarchical structure, which contains all the relevant GUI elements grouped together through GROIs. For specific details on the Overlap Removal, Filtering and NMS algorithm refer to the supplementary.

2.2. SEED: Spatially Enhanced Element Description Generation

Accurately describing the functionality of local GUI elements is essential for effective understanding of GUI and action grounding. Relying solely on visual appearance is unreliable since identical icons can serve different purposes in different contexts, and distinct icons may represent similar functions, leading to ambiguity. Textual and semantic cues around GUI elements help clarify functionality. Pairing icons with nearby text enables precise descriptions, while semantic associations (e.g., text linked to input fields or buttons) aid in identifying actionable elements.

We introduce SEED (Spatially Enhanced Element De-

scription), a prompting framework that employs Chain of Thought (CoT) [36] and In-Context Learning (ICL) [4] to generate spatially and semantically informed functional descriptions for all GUI elements. SEED processes an image I annotated with SoM-style ID tags, and a prompt with bounding boxes for detected elements (via our HSP module), and OCR-extracted text descriptors:

$$\mathcal{B}_{\text{icon}} = \{(i, b_{\text{icon},i})\}_{i=1}^{N_{\text{icon}}} \quad (1)$$

$$\mathcal{B}_{\text{text}} = \{(i, b_{\text{text},i}, d_i)\}_{i=N_{\text{icon}}+1}^{N_{\text{total}}}, \quad (2)$$

where $b_{\text{icon},i}$ and $b_{\text{text},i}$ are bounding boxes, and d_i represents OCR-derived text descriptors.

SEED outputs a spatially enhanced descriptor set \mathcal{A} :

$$\mathcal{A} = \{b, \ell, a, d \mid b \in \mathcal{B}_{\text{icon}} \cup \mathcal{B}_{\text{text}}\} \quad (3)$$

Each element’s attributes include bounding box b , label $\ell \in \{\text{paired, standalone, picture, actionable} - \text{text}\}$, set of associated elements a , and a spatially enhanced functional description d .

SEED classifies elements as paired or standalone based on semantics and positioning. Paired elements combine descriptors from nearby text/icons for a unified description, while standalone elements rely on visual cues alone. Text elements linked to interactive components (e.g., input fields, search bars, buttons) are labeled as actionable, and embedded icons are classified as {picture}.

We use ICL [4] with six examples from the ScreenSpot [19] dataset, The full SEED prompt with specific details about the SEED module is available in supplementary.

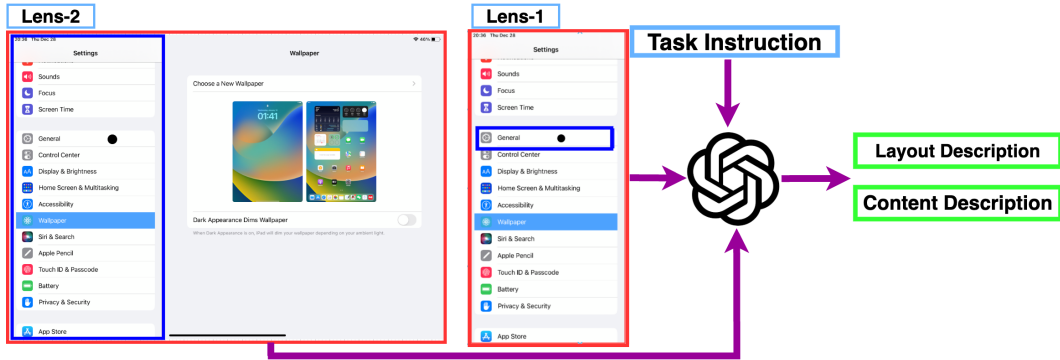


Figure 3. TRISHUL: Agentic GUI Referring Framework, the 2 Lenses created using our HSP module for local and global context. Lens-1 contains the local element (blue) in the cropped GROI (red), Lens-2 contains the GROI (blue) in the full input screenshot (red). The selected point is represented as the black dot. Both lenses are fed to the LVLm to generate Layout and Task description.

2.3. Agentic Formulation of Action Grounding

Platform	ScreenSpot		VisualWebBench	
	GPT-4o	GPT-4V	GPT-4o	GPT-4V
Mobile	0.91	0.81	-	-
Web	0.96	0.83	0.93	0.86
PC	0.92	0.83	-	-
Overall	0.93	0.82	0.93	0.86

Table 1. GROI proposal accuracy.

This section explains how the hierarchical nature of GUIs is leveraged for enhanced SoM style action grounding in LVLms as explained in fig. 2. Given an image I with Global Regions of Interest (GROIs) \mathcal{G} , bounding boxes for icons $\mathcal{B}_{\text{icon}}$ and text $\mathcal{B}_{\text{text}}$, OCR-derived text descriptors d_j , and an instruction I_s , the task is to identify the bounding box \mathcal{B} corresponding to the correct element required to complete the instruction in a single step.

TRISHUL performs action grounding in two stages. First, it proposes the most relevant GROI by passing the full annotated image $I_{\text{annotated}}$, cropped GROIs $\mathcal{G}_{\text{cropped}}$, and instruction I_s to the LVLm. The model outputs descriptions $\mathcal{D}_{\mathcal{G}}$ for each GROI and the ID of the most relevant one:

$$\{I_{\text{annotated}}, \mathcal{G}_{\text{cropped}}, I_s\} \longrightarrow \{\mathcal{D}_{\mathcal{G}}, \text{ID}_{\text{GROI}}\}. \quad (4)$$

GROI proposal accuracy is evaluated by checking if the ground truth bounding box midpoint lies inside the proposed GROI. Results with GPT-4o and GPT-4V on ScreenSpot [19] and VisualWebBench [27] (Table 1) confirm the effectiveness of our GROI ranking module.

Next, we use SEED (Section 2.2) to generate functionality descriptors for all local elements in the proposed GROI. The annotated image and descriptors are then used in a Set

of Marks [40] framework to predict the bounding box for grounding the instruction.

2.4. Agentic Formulation of GUI referring task

In this section we describe how the hierarchical screen parsing module can be leveraged to increase the ability of LVLms on the GUI referring task as explained in fig. 3. Given the input GUI screenshot I , the task involves describing the content and layout of any point P_i on the screen as input by a user, we use the input screenshot to detect all local elements and corresponding GROI candidates. We then identify the bounding box of the local element containing the selected point, and then the GROI encompassing this local element. Following the prompting approach of the ToL agent in [9], we curate two ‘‘lenses’’ or images to illustrate this hierarchy. The first lens consists of only the GROI region cropped from the original image, highlighting the local element with a labeled bounding box and marking the input point. The second lens shows the complete screenshot, highlighting the GROI with a labeled bounding box. Both lenses, along with the point coordinate P_i and input prompt, are sent to an LVLm, to generate the content description \hat{D}_c and the layout description \hat{D}_l .

3. Experiments

3.1. ScreenSpot and VisualWebBench

Dataset and Experiments- We evaluate the action grounding capability of TRISHUL agent on the ScreenSpot [19] dataset. ScreenSpot consists of 610 interface screenshots from mobile (iOS, Android), desktop (macOS, Windows), and web platforms, paired with 1,276 task instructions corresponding to actionable GUI elements. Traditional training-based methods, which are often trained on datasets like Screenspot, tend to perform poorly on out-of-distribution

Method	Mobile (ScreenSpot)		Desktop (ScreenSpot)		Web (ScreenSpot)		ScreenSpot	VisualWebbench
	Text	Icon/widget	Text	Icon/widget	Text	Icon/widget	Overall	Overall
Training Based								
SeeClick	78.0	52.2	72.2	30.0	55.7	32.5	53.4	31.0
CogAgent	67.0	24.0	74.2	20.0	70.4	28.6	47.4	59.0
OmniParser (GPT-4V)	90.1	54.1	88.6	60.0	73.4	27.1	66.9	58.3
OmniParser* (GPT-4V)	92.1	55.2	90.1	61.1	77.4	30.1	69.5	63.1
OmniParser (GPT-4o)	93.9	57.0	91.3	63.6	81.3	51.0	72.6	68.9
OmniParser* (GPT-4o)	94.8	66.3	95.4	64.2	80.8	32.0	73.7	69.9
Training Free								
GPT-4V	22.6	24.5	20.2	11.8	9.2	8.8	16.2	6.0
GPT-4o	20.2	24.9	21.1	23.6	12.2	7.8	18.2	6.7
TRISHUL [†] (GPT-4V)	75.8	38.4	66.3	25.4	69.5	31.2	53.4	56.3
TRISHUL* (GPT-4V)	88.6	37.9	82.9	23.5	72.6	29.1	59.0	58.1
TRISHUL* [†] (GPT-4V)	86.0	43.7	77.3	32.8	75.2	40.8	61.9	68.0
TRISHUL [†] (GPT-4o)	92.1	63.4	83.7	38.2	80.2	42.1	69.3	60.2
TRISHUL* (GPT-4o)	92.7	62.0	90.2	39.2	84.8	40.8	71.1	62.1
TRISHUL* [†] (GPT-4o)	93.8	64.6	85.6	45.7	83.5	44.7	72.2	68.0

Table 2. Performance across platforms and methods on ScreenSpot (Mobile, Desktop, Web) and VisualWebbench datasets. * denotes the usage of SEED module to improve the element functionality descriptors generated using OCR (for TRISHUL) / BLIPv2 (for OmniParser). [†] represents GROI-based action grounding instead of using the full image. *[†] represents our proposed end-to-end framework for action grounding that uses GROIs and SEED descriptors. Refer to Sec. 3.1 for detailed discussion.

samples such as those from VisualWebBench due to domain shift. Therefore, to assess the generalization capability of our approach, we also utilize the VisualWebBench [27] dataset’s action grounding subset, which consists of 103 pairs of images and their corresponding instruction.

Implementation Details: The formulation of the action grounding tasks for the datasets used in our experiments is discussed in detail in Section 2.3. The specific prompts employed for these tasks are provided in the supplementary.

Unfortunately, we were unable to replicate the results reported by OmniParser in their study on the ScreenSpot benchmark using the publicly available weights and codebase. In Table 2, we present the performance metrics for OmniParser as obtained from our own experiments on the ScreenSpot and VisualWebBench datasets. Due to the non-reproducibility of their results as observed above and limited resources, we were unable to verify their results on the AiTW and Mind2Web benchmarks hence we have chosen to exclude their results for these benchmarks from our analysis.

Evaluation and Results: As shown in Table 2, the TRISHUL agent, when paired with LVLMS (GPT-4V [29] and GPT-4o [30]), significantly outperforms the baseline GPT-4V and GPT-4o. Our approach also surpasses task-specific models such as SeeClick [7] and CogAgent [15], achieving an overall accuracy of 61.9% with GPT-4V and 72.2% with GPT-4o on the ScreenSpot benchmark. This performance exceeds SeeClick’s 53.4%, CogAgents 47.4% and closely rivals OmniParser’s 72.6%. On VisualWebBench [27], unlike SeeClick, which suffers a sharp drop in accuracy

on out-of-distribution data with 31% accuracy, TRISHUL maintains strong generalization, achieving a robust 68.0% accuracy with both GPT-4V and GPT-4o closely matching the performance of OmniParser which achieves 68.9%.

We further present ablations in Table 2 to assess the impact of the SEED module and GROI-based action grounding in TRISHUL. Removing SEED (TRISHUL[†]) results in a notable accuracy drop of 8.5% for GPT-4V and 2.9% for GPT-4o on ScreenSpot. Similarly, eliminating GROI-based action grounding (TRISHUL*) reduces accuracy by 2.9% for GPT-4V and 1.1% for GPT-4o. These results highlight the critical role of these components in TRISHUL’s performance.

Additionally, we demonstrate TRISHUL’s modularity by integrating its components into existing grounding pipelines. In Table 2, we show that augmenting OmniParser’s BLIPv2-derived icon descriptors—originally lacking local semantic context—with TRISHUL’s SEED module (OmniParser*) yields the best performance among training-based methods.

Our GROI-based action grounding proves particularly effective for web and desktop platforms, where hierarchical and content-dense GUIs benefit from structured decomposition. However, its impact is less pronounced in mobile interfaces, where regions have minimal semantic separation. Further details can be found in the supplementary. Lastly, we observe that GPT-4o outperforms GPT-4V significantly when paired with SEED, suggesting that improved reasoning capabilities in LVLMS enhance the accuracy of SEED-generated descriptions.

Method	General	Install	GoogleApps	Single	WebShopping	Overall
ChatGPT-CoT	5.9	4.4	10.5	9.4	8.4	7.7
Palm2-CoT	-	-	-	-	-	39.6
GPT-4V + Image	41.7	42.6	49.8	72.8	45.7	50.5
MM-Navigator (GPT-4V)	43	49.2	46.1	78.3	48.2	53.0
MM-Navigator (GPT-4o)	55.8	58.2	48.2	76.9	52.1	57.8
SeeClick (Qwen-VL)	54.0	66.4	54.9	63.5	57.6	59.3
TRISHUL (GPT-4V)	47.5	50.7	50.7	66.7	49.5	54.5
TRISHUL (GPT-4o)	52.9	60.7	55.0	78.2	52.6	60.0

Table 3. Results on the different categories on the AITW dataset. TRISHUL (GPT-4V) outperforms all prior GPT-4V baselines that use IconNet’s element detections. TRISHUL (GPT-4o) outperforms TRISHUL (GPT-4V) by 5.55% achieving State of the Art performance.

Methods	Modality	Cross-Website			Cross-Domain			Cross-Task		
		Ele.Acc	Op.F1	Step SR	Ele.Acc	Op.F1	Step SR	Ele.Acc	Op.F1	Step SR
MindAct (gen)	HTML	13.9	44.7	11.0	14.2	44.7	11.9	14.2	44.7	11.9
MindAct	HTML	42.0	65.2	38.9	42.1	66.5	39.6	42.1	66.5	39.6
GPT-3.5-Turbo	HTML	19.3	48.8	16.2	21.6	52.8	18.6	21.6	52.8	18.6
GPT-4	HTML	35.8	51.1	30.1	37.1	46.5	26.4	41.6	60.6	36.2
GPT-4V+Text	HTML, Image	38.0	67.8	32.4	42.4	69.3	36.8	46.4	73.4	40.2
GPT-4V+SOM	Image	-	-	32.7	-	-	23.7	-	-	20.3
CogAgent	Image	18.4	42.2	13.4	20.6	42.0	15.5	22.4	53.0	17.6
Qwen-VL	Image	13.2	83.5	9.2	14.1	84.3	12.0	14.1	84.3	12.0
SeeClick	Image	21.4	80.6	16.4	23.2	84.8	20.8	28.3	87.0	25.5
TRISHUL (GPT-4V)	Image	33.91	74.33	27.98	36.49	76.60	31.71	34.04	71.88	29.76
TRISHUL (GPT-4o)	Image	31.43	81.52	24.53	37.12	82.96	32	37.58	83.78	32.52

Table 4. Results for Cross-Website, Cross-Domain, and Cross-Task scenarios with Element Accuracy, Operational F1, and Step Success Rate metrics on the Mind2Web benchmark. TRISHUL (GPT-4o) consistently gives better Element Accuracy and Step Success Rate in all three scenarios on Image modality, its performance trails state-of-the-art HTML-based method like MindAct

LVL	Method	Desc. Acc.	Cont. Acc.	BERT	ROUGE
GPT-4V	Baseline	8	0.92	0.7130	0.1462
	ToL	31.84	14.24	0.7230	0.1527
	TRISHUL	32.64	17.07	0.7220	0.1534
Claude-3.5	Baseline	16.04	7.43	0.7274	0.1134
	ToL	60.56	43.02	0.7306	0.1462
	TRISHUL	60.91	49.74	0.7336	0.1495
GPT-4o	Baseline	18.82	5.64	0.6948	0.1843
	ToL	71.30	42.46	0.7147	0.1869
	TRISHUL	71.58	43.59	0.7151	0.1871

Table 5. Evaluation of description and content accuracy, BERT score, and ROUGE-L score across different methods on the Screen Point-and-Read benchmark. Desc. Acc. - Description Accuracy, Cont. Acc. - Content Accuracy

3.2. AITW

Dataset and Experiments To evaluate TRISHUL on the mobile navigation benchmark AITW[31], which consists of 30,000 instructions and 715,000 trajectories, we use the same train/test split as defined in [7]. This split retains only one trajectory per instruction, ensuring no overlap between the train and test sets.

Implementation details- We adopt a similar prompt format to that used in MM-Navigator [39], where we label the detected elements on the screen using SoM prompting and present the model with the annotated image and the clean image. However, we replace IconDet’s bounding boxes (as used in MM-Navigator) with local element boxes generated from our Hierarchal Screen Parsing method, and also provide our spatially enhanced element descriptions (Section 2.2) for all the local elements in our input prompt. The exact prompt is mentioned in the supplementary.

Evaluation and Results In Table 3, we report the baselines as presented in MM-Navigator[39]. The best perform-

ing baseline incorporates action history and uses only image modality for navigation. MM-Navigator presents baselines with GPT-4V only, we also run MM-navigator’s best configuration (Image+History) with GPT-4o to contrast it with TRISHUL’s GPT-4o performance. We observe that TRISHUL with GPT-4V outperforms all prior GPT-4V-based baselines, achieving an overall accuracy of 54.5%. With GPT-4o model, TRISHUL achieves an average accuracy of 60%, surpassing MM-Navigator’s GPT-4o baseline by over 2.2% to become the state of the art.

3.3. Mind2Web

Dataset and Experiments- To test on the web-navigation task we use the Mind2Web [8] dataset. The test set consists of three different categories - Cross Task, Cross Website, and Cross Domain having 252, 177, and 912 tasks respectively.

Implementation details - We use the pre-processed test set provided by [39]. During inference, we feed the detected local elements outputs from our Hierarchical Screen Parsing (HSP) module along with the clean image. Additionally, our input prompts are augmented with the descriptions of local elements from our SEED module. The prompt is mentioned in the supplementary.

Evaluation and Results - The results are presented in Table 4 where we compare multiple baselines across two modalities HTML and image. GPT-4V+SoM and GPT-4V+Text correspond to SeeAct [46] with image annotations and text choice grounding methods respectively. Without using any parsed HTML information, TRISHUL is able to outperform all the approaches relying on only GUI screenshots in almost every sub-category. Compared to other baselines we surpass them in Element accuracy and Step success rate, while remaining competitive in Operational F1. This indicates that the local elements detected by our HSP module and SEED descriptions provide highly valuable information for web navigation tasks. Although we provide better Operational F1 than HTML-based methods, we still falter when it comes to element accuracy and step success rate as predicting bounding boxes is a more complex task than selecting HTML elements.

3.4. Screen Point-and-Read

Dataset and experiments- We use the Screen Point and Read[9] benchmark to evaluate TRISHUL’s performance on the GUI referring task. It evaluates the accuracy of the generated content description \hat{D}_c and layout description \hat{D}_l for the region marked by the user over the interface. This benchmark comprises of 650 screenshots across three domains: web, mobile, and operating systems. To validate our method, we run experiments using GPT-4o [30], GPT-4V [29], and Claude-3.5-Sonnet [1], enabling us to examine performance across multiple LVLMS.

Evaluation and Results - To assess the quality of the

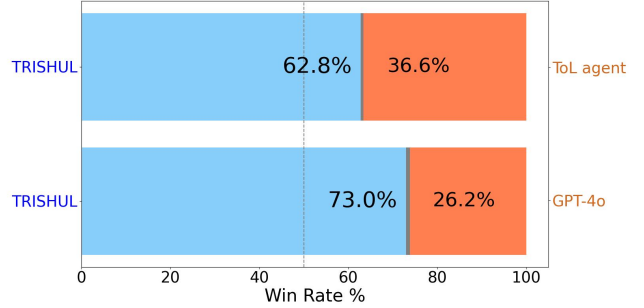


Figure 4. Human evaluation results on ScreenPR benchmark. TRISHUL is preferred by human annotators 63% of the time over ToL agent and 73% of the time over baseline GPT-4o

generated content description and layout description we employ the cycle consistency evaluation following the screen point-and-read [9] paper. The agent outputs (\hat{D}_c , \hat{D}_l) are fed into an auxiliary model, which is asked to complete a downstream task, with its performance indicating description quality. We benchmark our approach against baseline GPT-4o, Claude, and the ToL agent from Screen point-and-read, using GPT-4o, GPT-4V, and Claude-3.5-Sonnet as the primary models. We also compute language similarity metrics like BERT [44] score and ROUGE-L [25] to evaluate alignment with human-verified ground truth.

To further validate quality, we conduct two rounds of human evaluation: the first compares our approach against baseline GPT-4o, while the second compares our approach with the ToL agent, both using GPT-4o as the primary LLM. We employ 10 human annotators from [16] and ask them to choose between the description generated by our approach and the alternative approach. Each evaluator is presented with the labeled image and asked a single question “Given the image with the labeled point, which description do you prefer?”. The majority vote is used to select the preferred description. To ensure unbiased evaluation the annotators are unaware of which model generates which descriptions. The annotators are compensated at minimum wage.

TRISHUL consistently outperforms both the baseline and the ToL agent across all evaluation metrics for GPT-4V, Claude, and GPT-4o models (Table 5). Human evaluation results (Figure 4) further validate TRISHUL’s efficacy, with descriptions generated by TRISHUL being preferred by annotators 73% of the time over GPT-4o and 62.8% of the time over ToL. TRISHUL ties with GPT-4o 0.9% of the times and with ToL agent 0.6% of the times.

4. Discussion

4.1. Analysis on sampling multiple candidates

LVLM-based GUI agents that rely solely on visual perception aim to mirror human like interface interaction. Humans

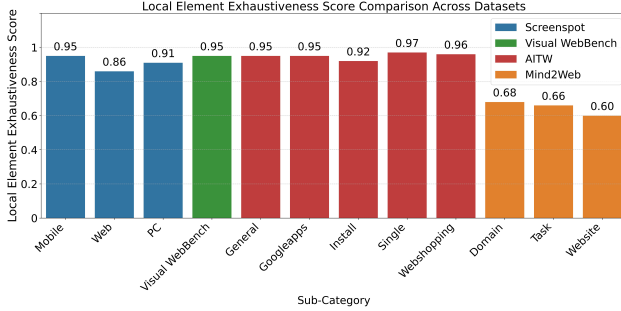


Figure 5. Local Element Exhaustiveness Score for ScreenSpot, Visual WebBench, AITW and Mind2Web

Benchmark	Model	Accuracy (%)		
		Pass@1	Pass@2	Pass@3
VisualWebBench	GPT-4o	68.0	81.6	83.5
	GPT-4V	56.3	69.9	71.8
	SeeClick	31.0	36.0	36.0
ScreenSpot	GPT-4o	72.2	77.8	80.0
	GPT-4V	59.0	67.2	70.6
	SeeClick	55.0	55.0	59.0

Table 6. Pass@1, Pass@2, and Pass@3 Accuracy (%) for VisualWebBench and ScreenSpot using GPT-4o, GPT-4V,(with the TRISHUL framework) and SeeClick models.

often explore multiple paths when interacting with novel/complicated GUIs. Traditional metrics like pass@1 (top@1), may not fully reflect an agent’s success in tasks that benefit from exploration. Recent research [21], shows that sampling and evaluating multiple potential action paths, then filtering them with a value model, improves success rates by reducing decision uncertainty.

The ToL agent has proven effective as a verification layer for mobile agents [9], accurately identifying correct and incorrect action paths. Leveraging this insight, we propose utilizing TRISHUL as a verification agent in a GUI agent system to enable multi-click grounding with enhanced accuracy. Our findings in Table 6 indicate that multi-sampling metrics like pass@2 and pass@3 improve grounding accuracy by over 10% across models on tasks in the ScreenSpot and VisualWebBench datasets. Here, pass@k highlights top K action-grounding candidates generated by TRISHUL.

4.2. Limitations

In Figure 5, we evaluate the Local Element Exhaustiveness (LEE) metric across various datasets and their splits. We evaluate LEE metric across various datasets and splits. The LEE score is binary: it is 1 if the midpoint of the ground truth (GT) bounding box falls within any detected local element bounding box from our Hierarchical Screen Parsing (HSP)

module; otherwise, it is 0. A low LEE score indicates a bottleneck in our pipeline after local element detection, with incomplete UI identification being the primary cause of failures in trajectory-level tasks such as Mind2Web and AITW. Our results show a strong correlation between LEE scores and TRISHUL’s performance, particularly in Mind2Web, where limited element detection significantly constrains effectiveness in web navigation. TRISHUL’s agentic formulation can also function as a standalone action-grounding framework for any base reasoning model in GUI navigation. Future work will focus on enhancing its reasoning and planning capabilities.

4.3. Potential Negative Impact

The deployment of TRISHUL for autonomous GUI navigation raises concerns about job displacement, digital inequality, and ethical misuse. Workers in data entry and software testing may need reskilling, and unequal access to AI could widen the digital divide. Additionally, risks like unauthorized data extraction or surveillance must be addressed. However, with responsible development—such as ethical AI guidelines, human-AI collaboration, and accessibility initiatives TRISHUL can enhance productivity while ensuring fairness, security, and inclusive technological progress.

5. Conclusion

In this paper, we introduced TRISHUL, a training-free agentic framework that enables LVLMs to achieve comprehensive GUI screen understanding using two key modules: HSP and SEED. The HSP module organizes GUI elements into a multi-granular hierarchical structure, distinguishing Global Regions of Interest (GROIs) from local elements, while the SEED module enhances spatial context-aware reasoning. Experiments on ScreenSpot, VisualWebBench, AITW, Mind2Web, and ScreenPR demonstrate that TRISHUL outperforms all training-free methods and rivals training-based approaches while maintaining superior cross-task and cross-platform generalizability.

References

- [1] Anthropic. Introducing claude 3.5, 2023. 7
- [2] Chongyang Bai, Xiaoxue Zang, Ying Xu, Srinivas Sunkara, Abhinav Rastogi, Jindong Chen, and Blaise Agüera y Arcas. Uibert: Learning generic multimodal representations for ui understanding. In *International Joint Conference on Artificial Intelligence*, 2021. 1
- [3] Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *ArXiv*, abs/2406.11896, 2024. 1, 2
- [4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan,

- Pranav Shyam, Girish Sastry, Amanda Askill, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Ma teusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. 3
- [5] Jieshan Chen, Chunyang Chen, Zhenchang Xing, Xiwei Xu, Liming Zhu, Guoqiang Li, and Jinshui Wang. Unblind your apps: Predicting natural-language labels for mobile gui components by deep learning. *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 322–334, 2020. 1
- [6] Jieshan Chen, Mulong Xie, Zhenchang Xing, Chunyang Chen, Xiwei Xu, Liming Zhu, and Guoqiang Li. Object detection for graphical user interface: old fashioned or deep learning or a combination? *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020. 1
- [7] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Annual Meeting of the Association for Computational Linguistics*, 2024. 1, 2, 5, 6
- [8] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *ArXiv*, abs/2306.06070, 2023. 1, 2, 7
- [9] Yue Fan, Lei Ding, Ching-Chen Kuo, Shan Jiang, Yang Zhao, Xinze Guan, Jie Yang, Yi Zhang, and Xin Eric Wang. Read anywhere pointed: Layout-aware gui screen reading with tree-of-lens grounding, 2024. 1, 4, 7, 8
- [10] Hiroki Furuta, Ofir Nachum, Kuang-Huei Lee, Yutaka Matsuo, Shixiang Shane Gu, and Izzeddin Gur. Multimodal web navigation with instruction-finetuned foundation models. *ArXiv*, abs/2305.11854, 2023. 1
- [11] Izzeddin Gur, Ulrich Rückert, Aleksandra Faust, and Dilek Z. Hakkani-Tür. Learning to navigate the web. *ArXiv*, abs/1812.09195, 2018. 1
- [12] Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. *ArXiv*, abs/2307.12856, 2023. 1
- [13] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. In *Annual Meeting of the Association for Computational Linguistics*, 2024. 1
- [14] Zecheng He, Srinivas Sunkara, Xiaoxue Zang, Ying Xu, Lijuan Liu, Nevan Wichers, Gabriel Schubiner, Ruby B. Lee, and Jindong Chen. Actionbert: Leveraging user actions for semantic understanding of user interfaces. In *AAAI Conference on Artificial Intelligence*, 2020. 1
- [15] Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juan-Zi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents. *ArXiv*, abs/2312.08914, 2023. 1, 5
- [16] <https://www.indikaai.com/>. *Indika.ai*. 7
- [17] JaidedAI. Easyocr: Ready-to-use ocr with 80+ supported languages and all popular writing scripts including latin, chinese, arabic, devanagari, cyrillic and etc. 2
- [18] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. 1
- [19] Marko Jurmu, Sebastian Boring, and Jukka Rieki. Screenshot: multidimensional resource discovery for distributed applications in smart spaces. In *International Conference on Mobile and Ubiquitous Systems: Networking and Services*, 2008. 3, 4
- [20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 2
- [21] Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. Tree search for language model agents, 2024. 8
- [22] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1
- [23] Yang Li, Jiacong He, Xiaoxia Zhou, Yuan Zhang, and Jason Baldridge. Mapping natural language instructions to mobile ui action sequences. *ArXiv*, abs/2005.03776, 2020. 1
- [24] Y. Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. Widget captioning: Generating natural language description for mobile user interface elements. In *Conference on Empirical Methods in Natural Language Processing*, 2020. 1
- [25] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. page 10, 2004. 7
- [26] Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement learning on web interfaces using workflow-guided exploration. *ArXiv*, abs/1802.08802, 2018. 1
- [27] Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. Visualwebbench: How far have multimodal llms evolved in web page understanding and grounding? *ArXiv*, abs/2404.05955, 2024. 1, 2, 4, 5
- [28] Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. Omniparser. *arXiv preprint arXiv:2408.00203*, 2024. 1
- [29] OpenAI. "gpt-4v(ision) system card", June, 2024. 1, 2, 5, 7
- [30] OpenAI. "hello gpt-4o.", June, 2024. 2, 5, 7
- [31] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Android in the wild: A large-scale dataset for android device control, 2023. 2, 6
- [32] Peter Shaw, Mandar Joshi, James Cohan, Jonathan Berant, Panupong Pasupat, Hexiang Hu, Urvashi Khandelwal, Kenton Lee, and Kristina Toutanova. From pixels to ui actions: Learning to follow instructions via graphical user interfaces. In *Advances in Neural Information Processing Systems*, 2023. 1

- [33] Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. World of bits: An open-domain platform for web-based agents. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3135–3144. PMLR, 2017. 1
- [34] Abishek Sridhar, Robert Lo, Frank F. Xu, Hao Zhu, and Shuyan Zhou. Hierarchical prompting assists large language model on web navigation. In *Conference on Empirical Methods in Natural Language Processing*, 2023. 1
- [35] Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. Screen2words: Automatic mobile ui summarization with multimodal learning. *The 34th Annual ACM Symposium on User Interface Software and Technology*, 2021. 1
- [36] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. "chain-of-thought prompting elicits reasoning in large language models", 2023. 3
- [37] Jason Wu, Xiaoyi Zhang, Jeffrey Nichols, and Jeffrey P. Bigham. Screen parsing: Towards reverse engineering of ui models from screenshots. *The 34th Annual ACM Symposium on User Interface Software and Technology*, 2021. 1
- [38] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *ArXiv*, abs/2404.07972, 2024. 1
- [39] An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Qinghong Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian J. McAuley, Jianfeng Gao, Zicheng Liu, and Lijuan Wang. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. *ArXiv*, abs/2311.07562, 2023. 1, 2, 6, 7
- [40] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023. 1, 4
- [41] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *ArXiv*, abs/2207.01206, 2022. 1
- [42] Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. Ferret-ui: Grounded mobile ui understanding with multimodal llms. In *European Conference on Computer Vision*, 2024. 1
- [43] China. Xiaoyan Zhang, Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Apagent: Multimodal agents as smartphone users. *ArXiv*, abs/2312.13771, 2023. 1
- [44] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675, 2019. 7
- [45] Xiaoyi Zhang, Lilian de Greef, Amanda Swearngin, Samuel White, Kyle I. Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, Aaron Everitt, and Jeffrey P. Bigham. Screen recognition: Creating accessibility metadata for mobile applications from pixels. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021. 1
- [46] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v(ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024. 1, 7
- [47] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023. 1