Attention-Driven GUI Grounding: Leveraging Pretrained Multimodal Large Language Models Without Fine-Tuning

Hai-Ming Xu[1*], **Qi Chen**[1*], **Lei Wang**[2], **Lingqiao Liu**[1†]

[1]Australian Institute for Machine Learning, The University of Adelaide
[2]University of Wollongong
{hai-ming.xu, qi.chen04, lingqiao.liu}@adelaide.edu.au, lei_wang@uow.edu.au

Abstract

Recent advancements in Multimodal Large Language Models (MLLMs) have generated significant interest in their ability to autonomously interact with and interpret Graphical User Interfaces (GUIs). A major challenge in these systems is grounding—accurately identifying critical GUI components such as text or icons based on a GUI image and a corresponding text query. Traditionally, this task has relied on fine-tuning MLLMs with specialized training data to predict component locations directly. However, in this paper, we propose a novel Tuning-free Attention-driven Grounding (TAG) method that leverages the inherent attention patterns in pretrained MLLMs to accomplish this task without the need for additional fine-tuning. Our method involves identifying and aggregating attention maps from specific tokens within a carefully constructed query prompt. Applied to MiniCPM-Llama3-V 2.5, a state-of-the-art MLLM, our tuning-free approach achieves performance comparable to tuning-based methods, with notable success in text localization. Additionally, we demonstrate that our attention map-based grounding technique significantly outperforms direct localization predictions from MiniCPM-Llama3-V 2.5, highlighting the potential of using attention maps from pretrained MLLMs and paving the way for future innovations in this domain.

Code — https://github.com/HeimingX/TAG.git

1 Introduction

The integration of artificial intelligence with Graphical User Interfaces (GUIs) holds tremendous potential to transform how humans interact with software systems. Leading this innovation are Multimodal Large Language Models (MLLMs) (OpenAI 2023; Reid et al. 2024; Anthropic 2024), which have shown exceptional capabilities in interpreting GUIs across various applications. A crucial task in AI for GUIs is GUI grounding—accurately identifying and localizing key components such as text and icons—since this is fundamental to enabling the automated operation of GUIs. While MLLMs excel at understanding GUI images, the precise grounding of GUI elements remains challenging.

Current state-of-the-art solutions often improve the GUI grounding capabilities of MLLMs through fine-tuning on specialized datasets, as demonstrated in works like (Hong et al. 2023; Cheng et al. 2024). In these methods, the MLLM directly predicts the location of GUI elements. In contrast, our approach takes a different path by leveraging the inherent attention patterns of a pretrained MLLM, utilizing its existing spatial awareness and attention mechanisms to achieve accurate GUI grounding without the need for additional fine-tuning.

We propose a novel Tuning-free Attention-driven Grounding (TAG) approach that carefully selects and aggregates attention patterns from MiniCPM-Llama3-V 2.5, a state-of-the-art MLLM, to perform GUI element grounding. Our method begins by identifying specific tokens from either the user input query or the model-generated response and then propagates the corresponding attention values back to the image plane. To further enhance performance, we implement a selective mechanism to filter out irrelevant attention heads, ensuring that only the most relevant attention is utilized for accurate grounding.

We compare our approach with existing tuning-based methods, and our results demonstrate that utilizing attention patterns from a pretrained model can achieve accurate GUI element grounding. Additionally, our approach significantly improves text localization. These findings highlight the untapped potential of leveraging inherent model capabilities and open the door to more robust, scalable, and efficient applications of MLLMs in GUI automation.

2 Related Works

Multimodal Large Language Models for GUI Agents The use of MLLMs (Liu et al. 2024a; Bai et al. 2023; Yao et al. 2024; Liu et al. 2024b; Zhu et al. 2023; Lu et al. 2024; Wang et al. 2023) as GUI agents marks significant progress in AI's ability to interact with GUI. These models understand user queries and images, enabling them to perform tasks across various platforms, from desktops to mobiles. Recent works in this domain have explored various applications, from automating routine tasks on desktop interfaces (Hong et al. 2023; Wu et al. 2024; Kil et al. 2024; He et al. 2024; Kapoor et al. 2024; Xie et al. 2024) to providing interactive assistance on mobile platforms (Ma, Zhang, and Zhao 2024; Nong et al. 2024; Wang et al. 2024b,c; You et al.

[*]These authors contributed equally.
[†]Corresponding Author.

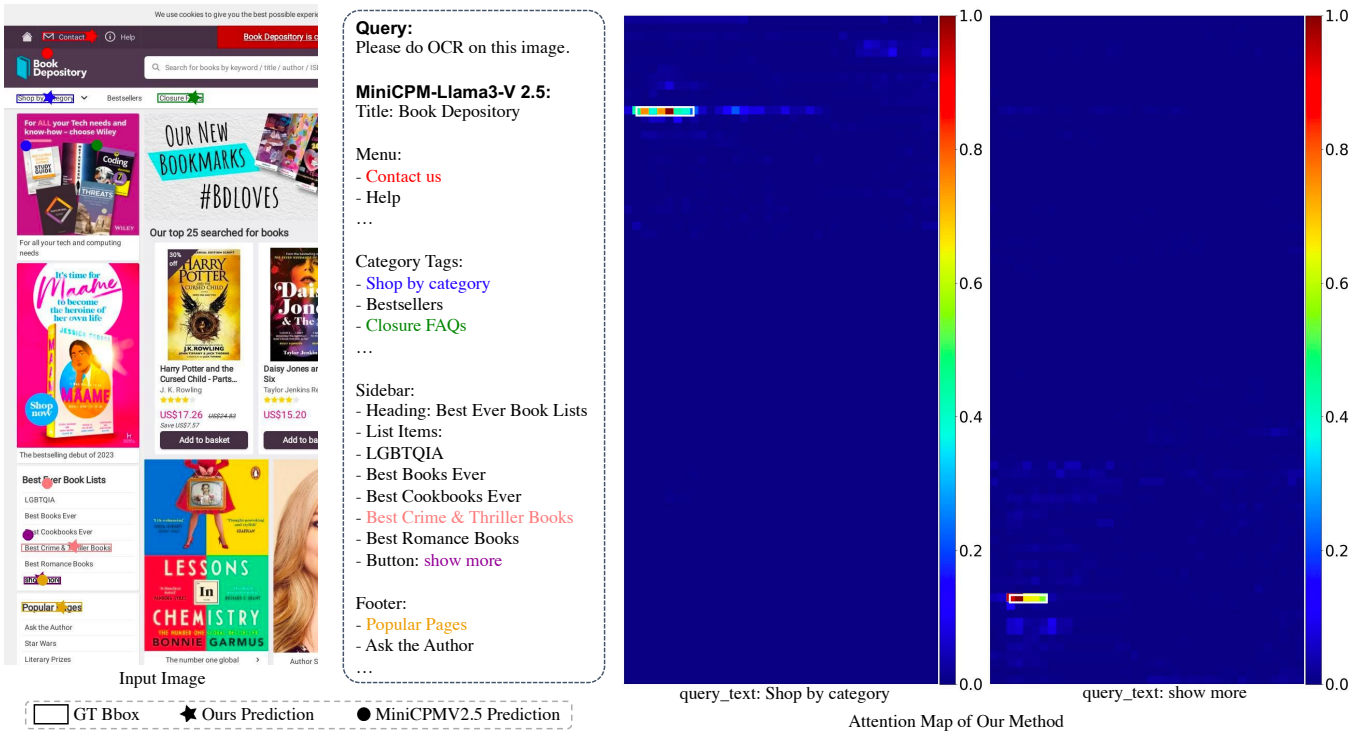

Figure 1: Illustration of MiniCPMV2.5's strong GUI image understanding but poor element localization. Our attention-driven GUI grounding leverages its inherent attention to enhance localization accuracy without fine-tuning, as shown on the right.

2024). These applications highlight the potential of MLLMs to act as autonomous agents that can understand and execute user commands across different platforms. However, the challenge often lies in effectively training these models to handle the intricacies and variability of GUIs without extensive domain-specific tuning.

Grounding in GUI Agents Grounding in GUI agents (Cheng et al. 2024; Li, Mitchell, and Myers 2020; Liu et al. 2024c; Gao et al. 2024; Wang et al. 2024a; Li and Li 2023; Wang, Li, and Li 2023; Li et al. 2020) involves the model's ability to locate and identify interface elements accurately, which is essential for effective interaction. Traditional methods (Cheng et al. 2024; Chen et al. 2024; Hong et al. 2023; Fan et al. 2024) typically require fine-tuning on detailed, annotated datasets. Recent research has explored both supervised and unsupervised techniques to enhance grounding accuracy, such as the SeeClick model (Cheng et al. 2024) fine-tunes on GUI-specific datasets. However, these methods can suffer from scalability issues and overfitting. Our work contributes to this field by proposing a tuning-free approach that leverages pre-trained MLLMs' inherent attention mechanisms to associate text queries with visual elements, offering a scalable and adaptable grounding solution.

3 Our Method

3.1 Preliminary

GUI Grounding GUI grounding is a crucial task for agents that interpret and interact with graphical user interfaces (GUIs). It demands that systems comprehend users' text queries, such as "I want to book a dental appointment on Tuesday", analyze GUI screenshots, and accurately pinpoint the relevant components. While recent advancements in MLLMs have shown potential in understanding both textual queries and visual GUI layouts, they often encounter challenges in precisely localizing elements without the aid of additional tools like OCR (Wang et al. 2024b) or Set-of-Mark (Yang et al. 2023; Wang et al. 2024b) techniques. Thus, SOTA methods typically rely on fine-tuning MLLMs with specialized training data to directly achieve accurate element localization (Hong et al. 2023; Cheng et al. 2024).

MiniCPMV2.5 and Its Attention Map MiniCPMV2.5 (i.e., MiniCPM-Llama3-V 2.5) is a SOTA MLLM that integrates a vision encoder, a token compression module, and the Llama3 language model. It supports high-resolution images up to 1344×1344 pixels with any aspect ratio, making it well-suited for precise GUI grounding tasks. To manage the large number of visual tokens generated from high-resolution inputs, the model uses cross-attention to compress thousands of vision patch embeddings into a fixed-size (Q) set of visual query tokens. These visual query tokens are then processed alongside text tokens by Llama3, which fuses the two modalities through multi-layer transformers utilizing multi-head self-attention. For more details, refer to the report by (Yao et al. 2024).

Empirically, as shown in Figure 1, when presented with a GUI image, MiniCPMV2.5 demonstrates a strong ability to comprehend the UI layout and accurately recognize optical

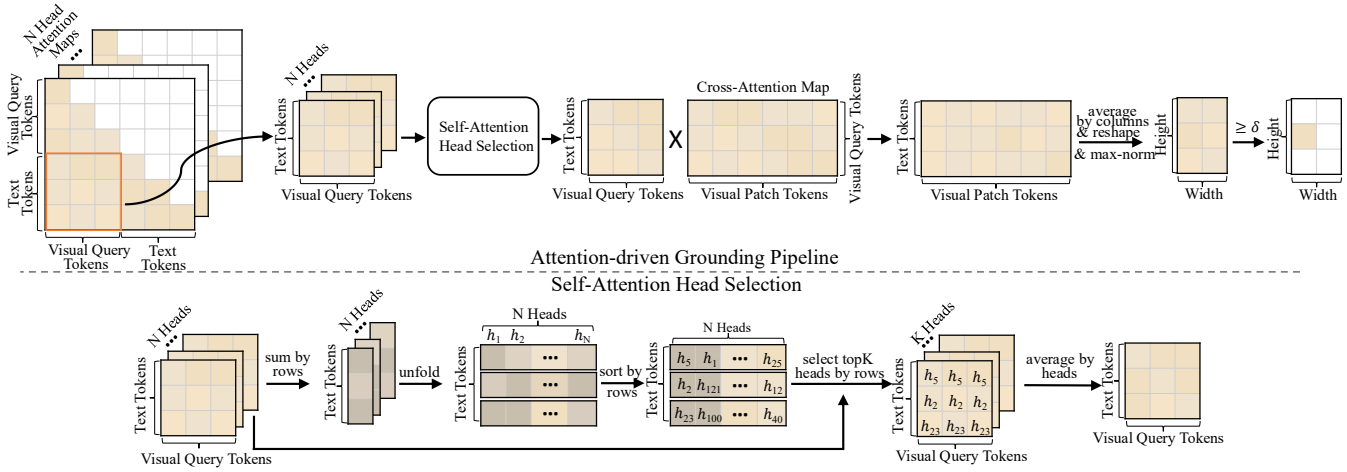

Figure 2: Overall pipeline of our TAG approach in Sec. 3.4 (top) and the self-attention selection module in Sec. 3.5 (bottom).

characters within the image. Additionally, due to its training on object-detection-related tasks, MiniCPMV2.5 is capable of localizing objects by predicting the bounding box of the object of interest.

Our method aims to further enhance localization performance by leveraging the attention maps within MiniCPMV2.5. Specifically, MiniCPMV2.5 consists of two major components: the token compression module and the Llama3 LLM, from which attention weights can be extracted. For the token compression module, attention values can be obtained from the cross-attention layer. By averaging these attention weights across all heads, we obtain an attention map $A_{\text{cross}} \in [0,1]^{Q\times H\cdot W}$, where Q is the number of visual query tokens, and H and W are the height and width of the patchified image, respectively. The self-attention weights in Llama3 LLM can be represented as $A_{llm} \in [0,1]^{N\times M\times M}$, where N is the total number of multi-head self-attention (MHA) layers multiplied by the number of attention heads per MHA, and M is the number of tokens input to the LLM, including both visual tokens and text tokens.

3.2 Overview of Our Method

Our method focuses on selecting and aggregating attention weights from MiniCPMV2.5 to achieve accurate localization of GUI elements. The key insight of our approach is that a well-crafted selection and aggregation strategy is essential for success. Specifically, our method comprises the following three components:

1. **Adaptive Text Token Selection:** A_{llm} contains self-attention values for all token pairs, but not all of them contribute to effective grounding. This component focuses on identifying the attention between the most relevant tokens to ensure accurate localization.
2. **Attention-driven GUI Grounding:** This component aggregates both A_{llm} and A_{cross} to identify the element localization.
3. **Self-Attention Head Selection:** This component improves grounding accuracy by selecting high-quality attention heads among 1024 attention heads in Llama3.

3.3 Adaptive Text Token Selection

GUI grounding tasks aim to locate elements relevant to a user's query. However, user queries often contain numerous tokens, not all of which pertain to the target GUI element. Some queries explicitly identify the element of interest, like "go to the next page" implying a click on the 'next page' button, while others only imply it, such as "take a photo as input" indirectly referring to the 'Camera' button in the GUI. Figure 4 illustrates how complex, multi-step queries can struggle to align with dynamic UI changes, leading to inaccurate grounding. Therefore, it's essential to develop a mechanism that selects key tokens and leverages the relevant self-attention weights for accurate GUI grounding.

Leveraging MiniCPMV2.5's remarkable ability to comprehend GUI images, in this paper we propose a simple yet effective strategy: *constructing the query prompt to prompt the model to first explicitly generate a description of the content or elements relevant to the query. We then use the attention between these descriptive tokens* $\{\mathcal{T}_j\}_{j=1}^{T}$ *and visual tokens to achieve localization.* This approach significantly improves GUI grounding performance by bridging the gap between user queries and UI elements.

3.4 Attention-driven GUI Grounding

As discussed in Section 3.1, the image tokens are not directly fed into the LLM. Instead, they are first compressed into visual query tokens before being passed to the LLM. This means that the selected text tokens, which correspond to the content or description of the target GUI element, may not directly attend to the image region. To address this issue, we propose a method to propagate attention from the selected text tokens to the image grid. Specifically as illustrated in Figure 2, we leverage the selected text tokens $\{\mathcal{T}_j\}_{j=1}^{\mathrm{T}}$ from Sec.3.3 to generate head-wise attention maps $A'_{llm} \in [0,1]^{\mathrm{N}\times\mathrm{T}\times\mathrm{Q}}$, which represent the attention between these text tokens and visual query tokens across all layers' multi-head self-attentions in Llama3. Here, N, T, and Q de-

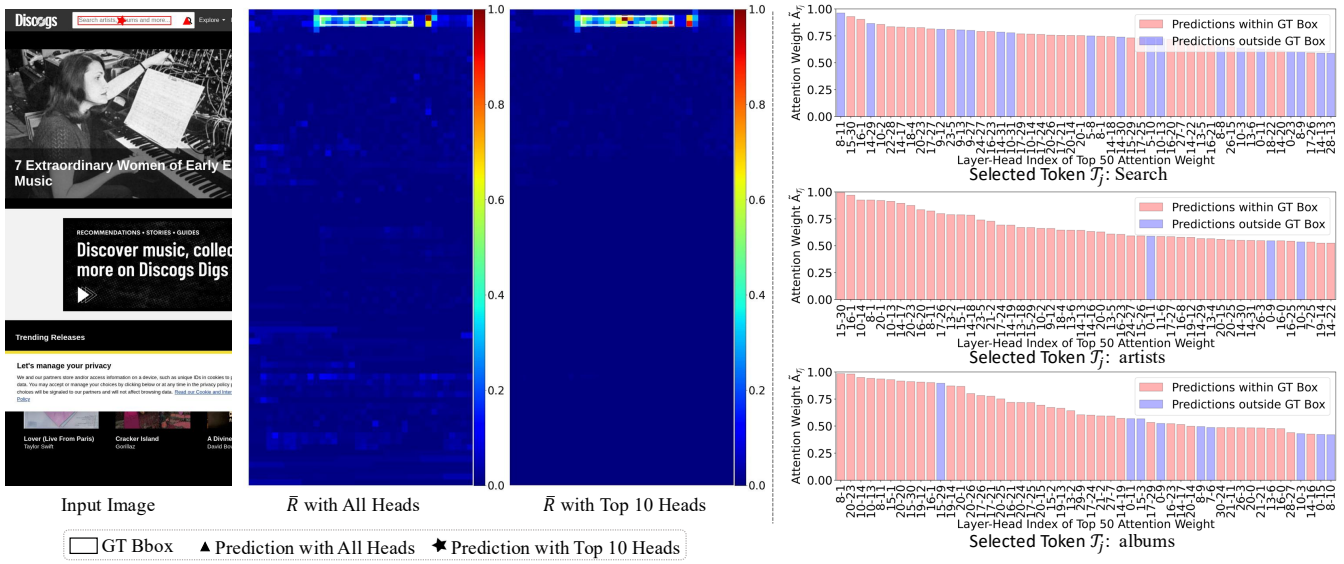

Figure 3: Demonstrating how choosing top self-attention heads improves text-to-image token mapping (see Sec. 3.5 for details).

note the number of heads in Llama3, the number of selected text tokens, and the number of visual query tokens, respectively. To obtain an overall relationship between each selected text token and the visual query tokens, we aggregate the attention from different heads by weighted summation:

$$\bar{A}_{llm}(\mathcal{T}_j) = \frac{1}{N}\sum_{k=1}^{N} \alpha_{k,j} A'_{llm}[k, j, :] \in [0,1]^{\mathrm{Q}}, \quad (1)$$

where k is the head index and $\alpha_{k,j}$ is the aggregation weight for k-th head and j-th selected text token, respectively. The strategy of how to set $\alpha_{k,j}$ will be discussed in Section 3.5. After obtaining $\bar{A}_{llm}(\mathcal{T}_j)$, which represents the attention of each selected text token to each visual query token, we propagate the attention from each visual query token to the corresponding image patch token using $A_{cross} \in [0,1]^{\mathrm{Q}\times(\mathrm{H}\cdot\mathrm{W})}$. This is accomplished through a simple matrix multiplication:

$$R_j = \bar{A}_{llm}(\mathcal{T}_j) \times A_{cross} \in [0,1]^{\mathrm{H}\cdot\mathrm{W}}. \quad (2)$$

Intuitively, this operation distributes the attention received by each visual query token to the corresponding image patch tokens, proportional to the attention values between the visual query token and each image patch token. Finally, to obtain an overall relationship between the query text and image patches, we average across different selected text tokens:

$$\bar{R} = \frac{1}{\mathrm{T}} \sum_{j\in\{1,2,\cdots,\mathrm{T}\}} R_j \in [0,1]^{\mathrm{H}\cdot\mathrm{W}}. \quad (3)$$

$\bar{R}$ represents the relevance of image patches to the query. To achieve pixel-level localization, we first map the relevance score from the patch to the pixel level by assigning the same value to all pixels within a patch (e.g., an 14×14 pixel grid). Next, we apply a threshold δ to binarize the image and identify connected regions. The region with the highest average relevance score is selected, and its center is used as the predicted location.

3.5 Self-Attention Head Selection

Empirically, we find that not all self-attention heads in the LLM part of MiniCPMV2.5 are equally useful for aligning the text tokens to the image patches. As the investigation presented in Figure 3, to ground the text "`Search artists, albums and more`" in the input field, the method with naive averaging attention maps of all heads falsely ground to the search icon. To find the reason, we further use each head's self-attention to map every text token to the image space separately. As figures shown on the right side of Figure 3, there are always some attention heads (which are colored in blue) that map the text token outside the ground truth bounding box for every text token, which means not all attention heads corresponding to each text token is equally effective in accurately mapping the token to its expected region. To determine the quality of attention heads, we find that magnitude of the average attention between a selected text token $\mathcal{T}_j$ and visual query tokens can be a good indicator, namely,

$$\tilde{A}^k_{\mathcal{T}_j} = \sum_{q\in\{1,2,\cdots,\mathrm{Q}\}} A'_{llm}[k, j, q]. \quad (4)$$

This is demonstrated by the observation that when $\tilde{A}^k_{\mathcal{T}_j}$ is larger, the head's attention is more likely to map the text token to the intended region. As illustrated in Figure 3, heads with high $\tilde{A}_{\mathcal{T}_j}$ tend to make predictions within the ground-truth bounding box (which are colored in red). This insight leads us to retain only the attentions of heads corresponding to the top-K values of $\tilde{A}_{\mathcal{T}_j}$. Additionally, we observe that the head rankings based on $\tilde{A}_{\mathcal{T}_j}$ vary across different text tokens. Therefore, we select the top heads for each token individually. This strategy effectively sets $\alpha_{k,j}$ to '1' for the selected heads while assigning '0' to the others.

MLLMs	w/o SFT	Aspect Ratio of Input Image (width:height)										Average
		1:4	9:21	9:19	1:2	9:16	4:3	16:9	2:1	21:9	4:1	
Qwen-VL-Chat	✓	7.3%	3.2%	3.1%	2.8%	2.2%	2.7%	2.9%	3.8%	4.5%	9.7%	4.2%
MiniCPMV2.5	✓	17.2%	13.6%	15.9%	21.4%	31.0%	80.2%	84.6%	81.1%	77.2%	59.2%	48.1%
TAG (Ours)	✓	**86.1%**	**80.3%**	**80.2%**	**84.8%**	**84.7%**	**82.6%**	**86.6%**	**87.9%**	**83.9%**	**88.0%**	**84.5%**
SeeClick	✗	52.7%	57.5%	56.6%	56.3%	57.5%	56.6%	63.6%	66.1%	65.9%	69.1%	60.2%

Table 1: Method comparison on the proposed OCG dataset. Our method significantly outperforms other tuning-free and tuning-based methods across all aspect ratios.

MLLMs	Model Size	w/o SFT	Mobile		Desktop		Web		Average
			Text	Icon/Widget	Text	Icon/Widget	Text	Icon/Widget	
MiniGPT-v2	7B	✓	8.4%	6.6%	6.2%	2.9%	6.5%	3.4%	5.7%
Qwen-VL-Chat	9.6B	✓	9.5%	4.8%	5.7%	5.0%	3.5%	2.4%	5.2%
GPT-4V	-	✓	22.6%	24.5%	20.2%	11.8%	9.2%	8.8%	16.2%
MiniCPMV2.5	8.5B	✓	40.3%	14.0%	62.4%	12.1%	67.4%	19.9%	36.0%
TAG (Ours)	8.5B	✓	**88.3%**	**29.3%**	**82.5%**	**28.6%**	**70.9%**	**29.1%**	**54.8%**
CogAgent	18B	✗	67.0%	24.0%	**74.2%**	20.0%	**70.4%**	28.6%	47.4%
SeeClick	9.6B	✗	**78.0%**	**52.0%**	72.2%	**30.0%**	55.7%	**32.5%**	**53.4%**

Table 2: Method comparison on Screenspot. The highest value in each column is bolded, considering both the upper section of tuning-free approaches and the lower section of tuning-based approaches.

4 Experiments

In this section, we compare our method to the SOTA ones on three benchmarks, each designed to test our method from different perspectives. Besides, we conduct several ablation studies to further analyze the effectiveness of our method. We use the greedy generation strategy in our method for a reproducible result and all experiments can be conducted on one NVIDIA RTX 4090 GPU.

4.1 Task1: Optical Character Grounding

Our method primarily achieves grounding by mapping text tokens to the image space. To directly validate our approach, we developed an optical character grounding benchmark using the Mind2Web (Deng et al. 2024) dataset. While Mind2Web was originally designed for text-based (HTML) GUI agent evaluation in website environments, it also includes corresponding screenshots, which we leveraged to create our novel dataset, *OCG*.

OCG Dataset First, we collect homepage screenshots from 104 websites in the Mind2Web test set. We then use the Azure Vision API tool[1] to obtain OCR information for each screenshot. This API can identify all text in the screenshot, including non-element text within images, allowing us to evaluate the MLLM's ability to locate general text. Besides, to assess model performance across various image aspect ratios, we crop sub-images from the homepage screenshots corresponding to different aspect ratios. We retain only the OCR bounding boxes that fell entirely within these sub-images for evaluation. Based on common screen resolutions[2], we construct 10 different aspect ratios (width:height): 1:4, 9:21, 9:19, 1:2, 9:16, 4:3, 16:9, 2:1, 21:9, and 4:1. This diverse set of aspect ratios allows us to comprehensively assess our model's robustness to varying image dimensions, which is crucial for real-world applications where screen sizes and orientations can vary significantly.

Baseline Methods We benchmark our approach against three notable models: MiniCPMV2.5 (Yao et al. 2024), a recently open-sourced SOTA MLLM that serves as the foundation for our method; SeeClick (Cheng et al. 2024), the current SOTA GUI grounding method; and Qwen-VL-Chat (Bai et al. 2023), the foundation model for SeeClick. For each model, we use specific prompts tailored to their respective functionalities. For Qwen-VL-Chat, we use "`Generate the bounding box of {query_text}`". SeeClick's prompt is "`In this UI screenshot, what is the position of the element "{query_text}" (with point)?`". MiniCPMV2.5 utilizes the prompt "`What is the bounding box of "{query_text}" in the image? The bounding box output format is: <box>xmin ymin xmax ymax</box>. Please directly output the bounding box.`"[3]. For our method, we employ "`What is the bounding box of "{query_text}"`". Since the query text is extracted by OCR which is well aligned with the corresponding text in the image, we thus directly use the query text for grounding to verify our method.

[1] https://azure.microsoft.com/products/ai-services/ai-vision

[2] https://gs.statcounter.com/screen-resolution-stats

[3] MiniCPMV2.5 sometimes fails to generate the box without the last prompt sentence.

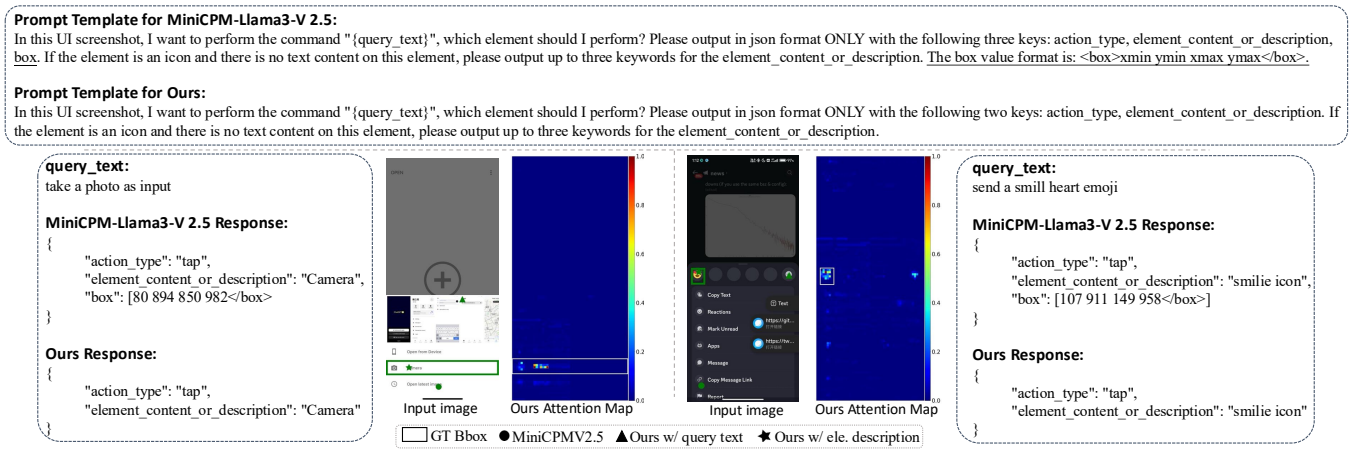

Figure 4: Demonstration of the comparing methods on two cases of ScreenSpot. Our attention-driven grounding with element description success in localizing the text and icon elements respectively. Please zoom in for a better view.

Results As shown in Tab. 1, the foundation MLLM Qwen-VL-Chat, while capable of detecting general objects, struggles to localize query text in the OCG task. In contrast, the more recent MiniCPMV2.5 demonstrates improved text grounding ability. However, MiniCPMV2.5's performance varies considerably across different aspect ratios, achieving 80.2% accuracy on the 4:3 aspect ratio but only 13.6% on 9:21. we speculate that although the model can support inputs of any aspect ratios, its pre-training data may make it impossible to include images of any aspect ratios, and the grounding ability may be difficult to generalize well to unseen aspect ratios. After being fine-tuned on the GUI-specific datasets, SeeClick improves the OCG task a lot compared to the Qwen-VL-Chat and surprisingly, it also excels at the more advanced MiniCPMV2.5. Notably, our approach, without additional SFT, substantially enhances MiniCPMV2.5's grounding ability. It achieves 84.5% average accuracy across 10 different aspect ratio settings, outperforming MiniCPMV2.5 by 36.4% and SeeClick by 24.3%.

4.2 Task2: GUI Element Grounding

Next, we evaluate our method on the ScreenSpot dataset which is a GUI element grounding benchmark.

ScreenSpot Dataset It is a realistic grounding evaluation dataset proposed by (Cheng et al. 2024), which contains over 600 GUI screenshots across three platforms, i.e., mobile, desktop and web. Each screenshot contains multiple command instructions and corresponding actionable elements, which include both text and icon/widget type elements.

Baseline Methods Following (Cheng et al. 2024), we compare our method to multiple popular foundation MLLMs: MiniGPT-v2 (Chen et al. 2023), Qwen-VL-Chat (Bai et al. 2023), the latest GPT-4V (OpenAI 2023) and MiniCPMV2.5 (Yao et al. 2024). Meanwhile, we also compare to CogAgent (Hong et al. 2023) and SeeClick (Cheng et al. 2024) which are SOTA GUI element grounding models supervised fine-tuned on a large amount of GUI-specific grounding tasks. To have a fair comparison, we directly use the evaluation setup in SeeClick and compare to the numbers reported in SeeClick paper. The prompt templates used for MiniCPMV2.5 and our method are presented in Fig. 4.

Results As Table 2 illustrates, foundation MLLMs generally perform poorly on GUI element grounding. MiniGPT-v2 and Qwen-VL-Chat average below 6% accuracy across platforms, while GPT-4V reaches only 16.2%. MiniCPMV2.5 performs better at 36.0%, likely due to OCR-related pretraining. GUI-specific fine-tuned models like CogAgent and SeeClick outperform these. Our approach, built on MiniCPMV2.5 without additional fine-tuning, achieves the highest average accuracy of 54.8%, surpassing even GUI-specific SFT models. It excels in text grounding, with accuracies of 88.3%, 82.5%, and 70.9% for mobile, desktop, and web platforms respectively. The cases demonstrated in Fig. 4 suggest that adaptively selecting text tokens from generated element descriptions can be more effective for GUI grounding than using query text directly.

4.3 Task3: GUI Agent Evaluation

We further evaluate our method on GUI agent benchmark.

Mind2Web Dataset (Deng et al. 2024) introduced the Mind2Web dataset to evaluate GUI agents in web environments using text-based HTML content. Each sample in the dataset typically consists of an open-ended, high-level goal instruction and a human action trajectory sequence, including clicking, selecting, and typing actions. While the released dataset also includes GUI screenshots corresponding to each sample, we follow (Cheng et al. 2024) and evaluate our method using only the GUI images. Since this work mainly focuses on the GUI grounding task, we evaluate compared methods on the Element accuracy metric. A prediction is correct if the predicted coordinate is within the target element's bounding box for vision-based methods.

Baseline Methods We compare our method to two vision-based GUI agents Qwen-VL and SeeClick (Cheng et al. 2024), which are both fine-tuned on the Mind2Web training set. Additionally, we include the foundation MLLM

MLLMs	w/o SFT	Cross-Task	Cross-Website	Cross-Domain	Average
MiniCPMV2.5	✓	15.0%	13.8%	18.2%	15.7%
TAG (Ours)	✓	25.4%	20.6%	**26.8%**	**24.3%**
Qwen-VL*	✗	15.9%	13.2%	14.1%	14.4%
SeeClick*	✗	**28.3%**	**21.4%**	23.2%	**24.3%**

Table 3: Element accuracy on Mind2Web dataset. The highest value in each column is bolded. Qwen-VL* and SeeClick* refer to the fine-tuning of Qwen-VL-Chat and SeeClick models, respectively, on Mind2Web training set.

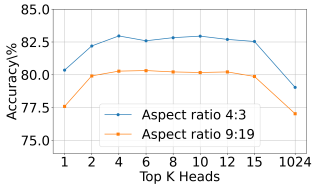

Figure 5: Ablation on Top K.

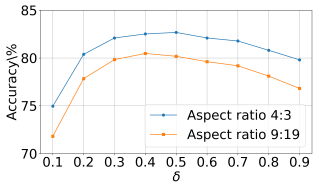

Figure 6: Ablation on δ.

MiniCPMV2.5 for comparison. Due to space constraints, the prompt templates for our method and MiniCPMV2.5 are provided in Figure 2 in the supplementary materials[4].

Results Results in Table 3 demonstrate that the proposed attention-driven grounding method improves MiniCPMV2.5's element accuracy across all settings, achieving comparable average accuracy to the best tuning-based approach. Figure 3 in the supplementary material showcases one example that our method grounds precisely at each step and successfully achieves the overall goal.

4.4 Ablation Study

We investigate the impact of each component in our TAG method. In Table 4, adding attention-driven grounding significantly improves performance, with Mobile Text accuracy increasing from 40.3% to 71.8%. Introducing adaptive text token selection further enhances results, particularly for Mobile Text (86.4%) and Icon/Widget (28.4%). The full model, incorporating self-attention selection, achieves the best performance across all metrics, with notable improvements in Mobile Text (88.3%) and Desktop Icon/Widget (28.6%).

4.5 More Discussions

Impact of Top K K is used for filtering self-attention weights and keeping top-ranked attentions for text-to-image mapping. Figure 5 shows that reducing K initially improves performance, with optimal results at $K = 10$ for both aspect ratios. However, extreme values ($K = 1$ or $K = 1024$, i.e., not reduced) lead to decreased accuracy. This demonstrates the benefit of filtering noisy attention heads while retaining sufficient information for text-to-image mapping. Based on these results, we use $K = 10$ in all experiments.

[4]The prompt template demonstrates our attention-driven grounding method for GUI agents, with the potential for further performance improvements through refined prompting.

Attn-d. ground	Token Select	self-attn filtering	Mobile		Desktop	
			Text	Icon/W.	Text	Icon/W.
✗	✗	✗	40.3%	14.0%	62.4%	12.1%
✓	✗	✗	71.8%	27.1%	73.2%	20.0%
✓	✓	✗	86.4%	28.4%	77.3%	24.3%
✓	✗	✓	80.9%	27.5%	78.8%	25.0%
✓	✓	✓	**88.3%**	**29.3%**	**82.5%**	**28.6%**

Table 4: Ablation on each component of our method.

Impact of Threshold δ δ is defined to determine the highlight region for final grounding prediction. Figure 6 shows that with a lower threshold $\delta \leq 0.3$, the model's performance is suboptimal due to including too many fairly attended regions. As δ increases, the model's performance reaches its peak at $\delta = 0.5$, but diminishes if δ is increased further. Thus $\delta = 0.5$ is used across all datasets.

Generalization Ability We applied our attention-driven grounding to another foundation MLLM, Qwen-VL-Chat (Bai et al. 2023), to demonstrate its generalization. Despite Qwen-VL-Chat's initial poor performance in GUI grounding, our method improved its accuracy from 2.7% to 10.2% on the 4:3 aspect ratio on our Mind2Web-OCG dataset. This showcases the broad applicability of our proposed mechanism across different foundation MLLMs.

5 Conclusion

In this paper, we introduce a Tuning-free Attention-driven Grounding (TAG) method, which uses the inherent attention mechanisms of pretrained MLLMs to accurately ground GUI elements without additional fine-tuning. Applied to MiniCPM-Llama3-V 2.5 model, TAG demonstrates that leveraging built-in model capabilities can effectively match or exceed the performance of traditional methods, particularly in text localization tasks. These suggest that MLLMs can be used more efficiently, reducing the need for resource-intensive fine-tuning while avoiding the risk of overfitting. TAG has the potential to be applied across various models and multimodal scenarios, offering a promising way to enhance AI's adaptability in interacting with user interfaces.

Limitations TAG relies heavily on the capabilities and the quality of the pretrained models it uses. If these models have inherent biases or have not been trained on diverse enough data, TAG's effectiveness could be limited, potentially affecting its accuracy and generalization ability. To alleviate this, we can expand the training datasets used for pretraining the MLLMs, which, while promising, is beyond the scope of this paper. We regard it as our future work.

Acknowledgements

This work was supported by the Centre for Augmented Reasoning, an initiative by the Department of Education, Australian Government.

References

Anthropic. 2024. Introducing the next generation of Claude.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities. *arXiv preprint arXiv:2308.12966*.

Chen, J.; Zhu, D.; Shen, X.; Li, X.; Liu, Z.; Zhang, P.; Krishnamoorthi, R.; Chandra, V.; Xiong, Y.; and Elhoseiny, M. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Chen, W.; Cui, J.; Hu, J.; Qin, Y.; Fang, J.; Zhao, Y.; Wang, C.; Liu, J.; Chen, G.; Huo, Y.; et al. 2024. GUICourse: From General Vision Language Models to Versatile GUI Agents. *arXiv preprint arXiv:2406.11317*.

Cheng, K.; Sun, Q.; Chu, Y.; Xu, F.; Li, Y.; Zhang, J.; and Wu, Z. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*.

Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; and Su, Y. 2024. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Fan, Y.; Ding, L.; Kuo, C.-C.; Jiang, S.; Zhao, Y.; Guan, X.; Yang, J.; Zhang, Y.; and Wang, X. E. 2024. Read Anywhere Pointed: Layout-aware GUI Screen Reading with Tree-of-Lens Grounding. *arXiv preprint arXiv:2406.19263*.

Gao, D.; Ji, L.; Bai, Z.; Ouyang, M.; Li, P.; Mao, D.; Wu, Q.; Zhang, W.; Wang, P.; Guo, X.; et al. 2024. AssistGUI: Task-Oriented PC Graphical User Interface Automation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13289–13298.

He, H.; Yao, W.; Ma, K.; Yu, W.; Dai, Y.; Zhang, H.; Lan, Z.; and Yu, D. 2024. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. *arXiv preprint arXiv:2401.13919*.

Hong, W.; Wang, W.; Lv, Q.; Xu, J.; Yu, W.; Ji, J.; Wang, Y.; Wang, Z.; Dong, Y.; Ding, M.; et al. 2023. CogAgent: A Visual Language Model for GUI Agents. *arXiv preprint arXiv:2312.08914*.

Kapoor, R.; Butala, Y. P.; Russak, M.; Koh, J. Y.; Kamble, K.; Alshikh, W.; and Salakhutdinov, R. 2024. OmniACT: A Dataset and Benchmark for Enabling Multimodal Generalist Autonomous Agents for Desktop and Web. *arXiv preprint arXiv:2402.17553*.

Kil, J.; Song, C. H.; Zheng, B.; Deng, X.; Su, Y.; and Chao, W.-L. 2024. Dual-View Visual Contextualization for Web Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14445–14454.

Li, G.; and Li, Y. 2023. Spotlight: Mobile UI Understanding using Vision-Language Models with a Focus. In *The Eleventh International Conference on Learning Representations*.

Li, T. J.-J.; Mitchell, T.; and Myers, B. 2020. Interactive task learning from GUI-grounded natural language instructions and demonstrations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 215–223.

Li, Y.; He, J.; Zhou, X.; Zhang, Y.; and Baldridge, J. 2020. Mapping Natural Language Instructions to Mobile UI Action Sequences. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8198–8210. Online: Association for Computational Linguistics.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024a. Visual Instruction Tuning. *NeurIPS*, 36.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Liu, J.; Song, Y.; Lin, B. Y.; Lam, W.; Neubig, G.; Li, Y.; and Yue, X. 2024c. VisualWebBench: How Far Have Multimodal LLMs Evolved in Web Page Understanding and Grounding? *arXiv preprint arXiv:2404.05955*.

Lu, H.; Liu, W.; Zhang, B.; Wang, B.; Dong, K.; Liu, B.; Sun, J.; Ren, T.; Li, Z.; Sun, Y.; et al. 2024. DeepSeek-VL: Towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*.

Ma, X.; Zhang, Z.; and Zhao, H. 2024. CoCo-Agent: A Comprehensive Cognitive MLLM Agent for Smartphone GUI Automation. In *Findings of the Association for Computational Linguistics ACL 2024*, 9097–9110.

Nong, S.; Zhu, J.; Wu, R.; Jin, J.; Shan, S.; Huang, X.; and Xu, W. 2024. MobileFlow: A Multimodal LLM For Mobile GUI Agent. *arXiv preprint arXiv:2407.04346*.

OpenAI. 2023. GPT-4V(ision) System Card.

Reid, M.; Savinov, N.; Teplyashin, D.; Lepikhin, D.; Lillicrap, T.; Alayrac, J.-b.; Soricut, R.; Lazaridou, A.; Firat, O.; Schrittwieser, J.; et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Wang, B.; Li, G.; and Li, Y. 2023. Enabling Conversational Interaction with Mobile UI using Large Language Models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. New York, NY, USA: Association for Computing Machinery. ISBN 9781450394215.

Wang, H.; Li, T.; Deng, Z.; Roth, D.; and Li, Y. 2024a. Devil's Advocate: Anticipatory Reflection for LLM Agents. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 966–978. Miami, Florida, USA: Association for Computational Linguistics.

Wang, J.; Xu, H.; Ye, J.; Yan, M.; Shen, W.; Zhang, J.; Huang, F.; and Sang, J. 2024b. Mobile-Agent: Autonomous Multi-Modal Mobile Device Agent with Visual Perception. *arXiv preprint arXiv:2401.16158*.

Wang, L.; Deng, Y.; Zha, Y.; Mao, G.; Wang, Q.; Min, T.; Chen, W.; and Chen, S. 2024c. MobileAgentBench: An Efficient and User-Friendly Benchmark for Mobile LLM Agents. *arXiv preprint arXiv:2406.08184*.

Wang, W.; Lv, Q.; Yu, W.; Hong, W.; Qi, J.; Wang, Y.; Ji, J.; Yang, Z.; Zhao, L.; Song, X.; et al. 2023. CogVLM: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*.

Wu, Z.; Han, C.; Ding, Z.; Weng, Z.; Liu, Z.; Yao, S.; Yu, T.; and Kong, L. 2024. Os-copilot: Towards generalist computer agents with self-improvement. *arXiv preprint arXiv:2402.07456*.

Xie, T.; Zhang, D.; Chen, J.; Li, X.; Zhao, S.; Cao, R.; Hua, T. J.; Cheng, Z.; Shin, D.; Lei, F.; et al. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972*.

Yang, J.; Zhang, H.; Li, F.; Zou, X.; Li, C.; and Gao, J. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Yao, Y.; Yu, T.; Zhang, A.; Wang, C.; Cui, J.; Zhu, H.; Cai, T.; Li, H.; Zhao, W.; He, Z.; Chen, Q.; Zhou, H.; Zou, Z.; Zhang, H.; Hu, S.; Zheng, Z.; Zhou, J.; Cai, J.; Han, X.; Zeng, G.; Li, D.; Liu, Z.; and Sun, M. 2024. MiniCPM-V: A GPT-4V Level MLLM on Your Phone. *arXiv preprint 2408.01800*.

You, K.; Zhang, H.; Schoop, E.; Weers, F.; Swearngin, A.; Nichols, J.; Yang, Y.; and Gan, Z. 2024. Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs. *arXiv preprint arXiv:2404.05719*.

Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.