MP-GUI: Modality Perception with MLLMs for GUI Understanding

Ziwei Wang[1†], Weizhi Chen[1†], Leyang Yang[1], Sheng Zhou[2*], Shengchu Zhao[3], Hanbei Zhan[1],
Jiongchao Jin[3], Liangcheng Li[1], Zirui Shao[1], Jiajun Bu[1]
[1]College of Computer Science and Technology, Zhejiang University, China
[2]Zhejiang Key Laboratory of Accessible Perception and Intelligent Systems, Zhejiang University, China
[3]Ant Group
{wangziwei98, zhousheng_zju, chenweizhi, yangleyang, beiii7533}@zju.edu.cn
{shaozirui,liangcheng_li, bjj}@zju.edu.cn {shengchu.sc,jinjiongchao.jjc}@antgroup.com

Abstract

Graphical user interface (GUI) has become integral to modern society, making it crucial to be understood for human-centric systems. However, unlike natural images or documents, GUIs comprise artificially designed graphical elements arranged to convey specific semantic meanings. Current multi-modal large language models (MLLMs) already proficient in processing graphical and textual components suffer from hurdles in GUI understanding due to the lack of explicit spatial structure modeling. Moreover, obtaining high-quality spatial structure data is challenging due to privacy issues and noisy environments. To address these challenges, we present MP-GUI, a specially designed MLLM for GUI understanding. MP-GUI features three precisely specialized perceivers to extract graphical, textual, and spatial modalities from the screen as GUI-tailored visual clues, with spatial structure refinement strategy and adaptively combined via a fusion gate to meet the specific preferences of different GUI understanding tasks. To cope with the scarcity of training data, we also introduce a pipeline for automatically data collecting. Extensive experiments demonstrate that MP-GUI achieves impressive results on various GUI understanding tasks with limited data. Our codes and datasets are publicly available at https://github.com/BigTaige/MP-GUI.

1. Introduction

Graphical user interface (GUI) is an important medium of human-computer interaction. Understanding GUI is critical in many human-centric interactions such as accessibility [21], automated agent systems [15, 19, 38, 45] and app testing systems [39]. The early research on GUI understanding mainly contributes to simple deep models [6, 29, 33, 64] to accomplish specific tasks. Recently, the rapid development of MLLMs [3, 14, 55, 60] has achieved tremendous success in various visual understanding scenarios [24, 32, 46, 53, 61, 63] (*e.g.*, natural image and documents) and shown great potential to GUI recently [15, 62].

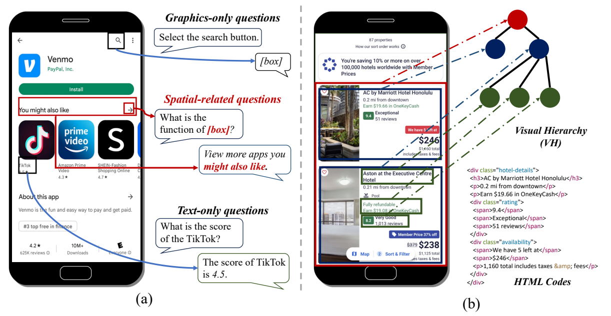

Figure 1. (a) Graphical, textual and spatial oriented tasks on GUI understanding. (b) Some classic GUI spatial structure forms.

Although a few efforts have been made to utilize MLLMs for learning GUI knowledge[6, 47, 62], they typically treat the GUI screens as a natural images problem. Unlike objects in unstructured natural images that lack a pre-defined functional or semantic organization, GUI elements (*e.g.*, icons/widgets and texts) are artificially designed and spatially arranged to convey specific semantic meanings, which are crucial for learning GUI knowledge. For example, in Fig. 1-(a), answering the question `"What is the function of the icon in the box?"` highly depends on the spatial relationship between the `"you might also like"` text and `"right arrow"` button that are semantically related. Otherwise, the `"right arrow"` button will only convey the fundamental meaning of `"going right"` and result in a

*Corresponding author. †Equal Contribution.

wrong answer. However, existing MLLMs for GUI understanding [15, 30, 62] only rely on the original vision backbone[20, 44] to provide global screen visual clues and combine instruction fine-tuning to learn GUI knowledge. As a result, despite their advanced capabilities in graphical and textual recognition [14, 60], existing works still suffer from inaccurate feature representation after fusion and misunderstanding the GUI structure.

Meanwhile, accessing the screen spatial structure data is quite challenging in practice. Some previous deep models have utilized manual and programmatic exploration [17, 43] to obtain spatial structure information such as View Hierarchy (VH) and HTML [48] (Fig. 1-(b)). However, they are often noisy [51] and inconsistent with screenshots [34], failing to meet the demands of purely visual understanding.

To tackle the aforementioned challenges, we present MP-GUI, a dual visual-clues MLLM for visual GUI understanding. MP-GUI extracts graphical, textual, and spatial modality signals from the screen using three perceivers. This design reinforces the model's capability to grasp both visual and textual modalities by integrating spatial information, improving overall screen perception. Notably, each type of GUI understanding task may have distinct screen content preferences [7, 23, 35]. Thus, MP-GUI includes a Fusion Gate that adaptively combines screen signals of different modalities, enabling task-oriented feature fusion. Unlike current end-to-end implicit training[15, 38], we design a training recipe including a Multi-stage Training Strategy and matching training objectives, especially a novel Spatial Relationship Prediction task and a data synthesis pipeline, which explicitly guides the model to learn GUI knowledge. Extensive experiments on various GUI-related benchmarks showcase the robust performance of MP-GUI. Our main contributions can be summarized as follows:

(i) We propose MP-GUI, which provides GUI-tailored visual clues for LLM via three perceivers and a semantically guided Fusion Gate, endowing the MLLM with effective GUI perception and understanding capability.

(ii) We introduce a training recipe with multi-stage strategies and task-specific objectives, which include a novel Spatial Relationship Prediction task and a data synthesis pipeline, to enable stage-wise explicit training of each perceiver in GUI knowledge learning.

(iii) We collect various GUI-related downstream tasks and the experimental results demonstrate the effectiveness of our special designs in the GUI scenarios.

2. Related Work

2.1. Multi-modal Large Language Models

Recently, Large Language Models (LLMs) [1, 2, 10, 58] achieve remarkable results in various text tasks. On this basis, for text and vision modalities, most Multi-modal Large Language Models (MLLMs) [3, 8, 14, 22, 55, 60], incorporate a pre-trained vision backbone, such as CNNs [27, 28] or ViT [8], and utilize a bridge module, like MLP [14, 41] or an attention-based resampler [31], to connect the vision backbone with the LLM for providing visual clues from images. Thanks to the powerful generic capabilities, instruction-tuning MLLMs on domain-specific data can quickly learns domain knowledge [42]. ChemVLM [32] utilizes chemically relevant instructional data to facilitate the acquisition of multimodal chemical knowledge by foundational models. Med-MLLM [40] and MLeVLM [56] harness medical multimodal data to enhance the foundational model's performance within medical contexts. SeeClick [15] collects GUI-related data to instruction-tune the foundational MLLM [8] for improving the grounding capability. Furthermore, ShowUI [38] introduces a Visual Token Selection approach to reduce computational costs. It also integrates various GUI tasks in a systematic manner, aiming to effectively guide the MLLM [55] in learning domain knowledge. Generally, for most research on solving domain-specific multi-modal tasks, the dense image features generated by the vision backbone are the sole vision clues provided to the LLM. Given the distinctiveness of GUI scenarios, we introduce a dual-visual-clues framework, which provides the LLM with additional GUI-tailored visual clues, thus enhancing the screen perception capabilities of the MLLM.

2.2. GUI Understanding

GUI understanding involves a full range of downstream tasks, such as screen grounding [6, 15], widget captioning [35], screen question answering [5, 12], navigation [18, 50], screen summarization [54], etc. Considering the huge difference between GUI and general visual scenarios, many GUI-specialized models have been proposed[5, 6, 29, 30, 62]. Pix2Struct [29] uses a dynamic patching strategy to adaptively divide the UI screen based on its resolution, along with a pre-training task of HTML reconstruction to allow the model to learn the UI knowledge. ScreenAI [5] uses a LLM [4] to synthesize challenging screen schema data and integrates a large amount of UI-related pre-training data to improve model's UI and Infographics understanding. Ferret-UI [62] uses image segmentation strategies to ensure that the original aspect ratio of mobile screenshots are not destroyed, and collects extensive GUI-related training data to improve the understanding of the foundational MLLM [61] in GUI scenarios. Existing methods blend GUI-related data and implicitly learn GUI knowledge through unified end-to-end training. In contrast, our approach uses three GUI-tailored perceivers integrated with a multi-stage training strategy. This enables training each perceiver with specific data at different stages, clarifying GUI knowledge learning via explicit training.

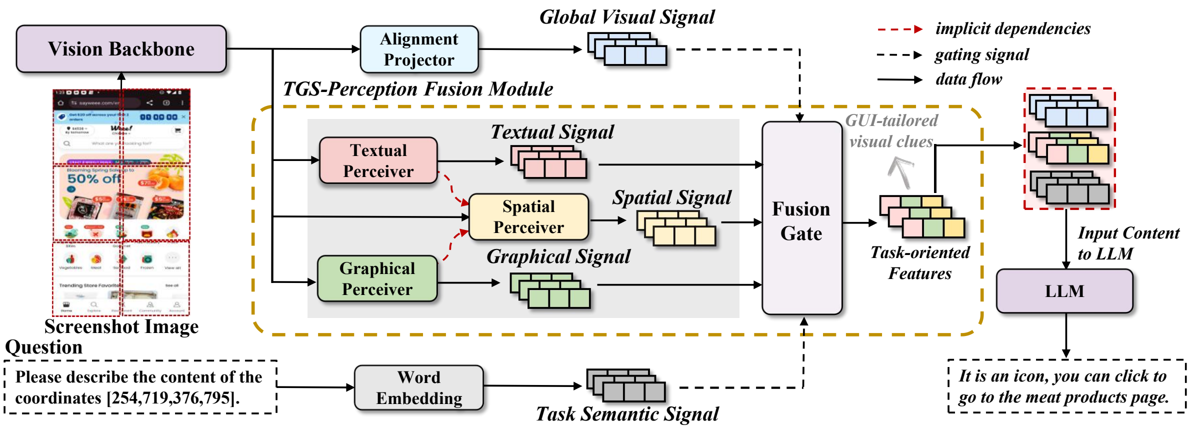

Figure 2. Overview of our MP-GUI. MP-GUI consists of three parts: (1) a vision backbone providing visual clues of the screenshot; (2) a TGS-Perception Fusion Module including three GUI-tailored perceivers for extracting specific GUI modality signals and a Fusion Gate for dynamically fusing these signals based on task semantics to produce GUI-tailored visual clues; and (3) an LLM generating results relying on screen visual clues, GUI-tailored visual clues, and task semantic signal.

3. Methodology

3.1. Model Architecture.

As shown in Fig. 2, MP-GUI is composed of three components: (1) a vision backbone; (2) a TGS-Perception Fusion Module (TGS-PFM) which includes three GUI-tailored perceivers and a Fusion Gate (FG) module; and (3) a LLM.

Given a screenshot image, we first utilize the Dynamic High-Resolution method [14] to divide the high-resolution image into several sub-images of 448×448 pixels with optimal proportions. Let $\mathcal{I} \in \mathbb{R}^{N \times \mathcal{C}}$ denote the visual clues produced by vision backbone, where N means the number of image tokens and C is logits dimension. $\mathcal{Q} \in \mathbb{R}^{M \times \mathcal{D}}$ means the word embeddings of the question, where M denotes the token length and D is the dimension. Besides, visual clues $\mathcal{I}$ will be sent to an pre-trained Alignment Projector to align the dimention of LLM and get $\overline{\mathcal{I}} \in \mathbb{R}^{N \times \mathcal{D}}$.

GUI-tailored Perceivers. Using the dense embeddings produced by vision backbone as image representation is a convenient paradigm to obtain visual clues [14, 22, 60]. However, the GUI information combined with visual clues is complex for several reasons: ***(i)* screen images incorporate numerous multimodal elements and intricate details, which increase the difficulty of grounding for MLLMs; *(ii)* there are high-level semantic associations between elements that need to be clearly identified, posing a challenge for MLLMs in understanding and being aware of the spatial relationships between GUI elements on the screen.**

This complexity inhibits MLLMs from effectively handling GUI understanding tasks that rely solely on global visual clues. Thus, we propose the TGS-PFM, where three GUI-tailored perceivers (MLPs) are proposed for each type of signals on the screen, *i.e.*, textual, graphical, and spatial relationship between them, named Textual Perceiver (TxP) $\Phi_{\mathbb{T}}$, Graphical Perceiver (GaP) $\Phi_{\mathbb{G}}$ and Spatial Perceiver (SaP) $\Phi_{\mathbb{S}}$, respectively. We design a specific training recipe to guide each perceiver in extracting modality-specific signals from visual clues $\mathcal{I}$, which will be discussed in Sec. 3.2. Thus, we can extract the textual signal $X_t = \Phi_{\mathbb{T}}(\mathcal{I})$, the graphical signal $X_g = \Phi_{\mathbb{G}}(\mathcal{I})$, and the spatial signal $X_s = \Phi_{\mathbb{S}}(\mathcal{I})$ from $\mathcal{I}$, where $X_t, X_g, X_s \in \mathbb{R}^{N \times \mathcal{D}}$.

Fusion Gate. Different GUI-related downstream tasks exhibit varying preferences for screen content. For example, in widget captioning [35], the model should pay more attention to graphical signals (icon/widget) related to the question; in screen summarization [54], both textual and graphical signals across the screen should be emphasized; and in screen question answering [5, 23] and navigation [18, 50], all three signals should be taken into consideration. Thus, the GUI signals extracted by three perceivers should be dynamically fused based on semantics to align with the requirements of tasks. Inspired by MoE [16, 37, 65] technology, we introduce FG to achieve the above goals.

Specifically, we first concatenate three perceiver signals X_t, X_g and X_s as a union of modality information, $X_f = [X_t; X_g; X_s]$, where $X_f \in \mathbb{R}^{3N \times \mathcal{D}}$ and ';' means concatenation operation. Subsequent to a self-attention operation, interacting $\mathcal{Q}$ with $\overline{\mathcal{I}}$ to align dimensions and enhance the semantics of $\mathcal{Q}$ results in the gating signal $\mathcal{G} \in \mathbb{R}^{N \times \mathcal{D}}$. Next, $\mathcal{G}$ is used to fuse features with X_f, incorporating semantic awareness. We can formulate this as follows:

$$\mathcal{G} = \text{softmax}(\frac{(\overline{\mathcal{I}}W_g^Q)(\mathcal{Q}W_g^K)^T}{\sqrt{\mathcal{D}}})(\mathcal{Q}W_g^V), \quad (1)$$

$$\mathcal{X}_t = \text{softmax}(\frac{(\mathcal{G}W_t^Q)(X_fW_t^K)^T}{\sqrt{\mathcal{D}}})(X_fW_t^V), \quad (2)$$

where $\mathcal{X}_t \in \mathbb{R}^{N \times \mathcal{D}}$ denotes task-oriented features, which could serve as GUI-tailored visual clues, and $W_g^Q, W_g^K, W_g^V, W_t^Q, W_t^K, W_t^V \in \mathbb{R}^{\mathcal{D} \times \mathcal{D}}$ are learnable parameter matrices.

With FG, MP-GUI interprets the task semantics of the question, interacts with the extracted GUI signals of three perceivers, and dynamically fuses them to produce task-oriented features. Finally, $\overline{\mathcal{I}}$, $\mathcal{X}_t$ and $\mathcal{Q}$ are concatenated along the sequence dimension as LLM input.

3.2. Multi-stage Training Strategy

Our TGS-PFM consists of two components: three perceivers (TxP, GaP and SaP) and FG. The perceivers focus on modality-specific signals, while FG dynamically fuses these signals based on task semantics to produce GUI-tailored visual clues. The order of training these components is crucial, hence we introduce a Multi-stage Training Strategy (MTS). Tab. 1 presents the overview of MTS.

Perceivers Training. We train each perceiver (*Step 1–3*) using specialized data to guide TGS-PFM to extract modality-specific signals on the screen. We first train TxP and GaP, aiming to warm up MP-GUI in adapting GUI-related scenarios. Afterwards, we train the SaP, which implicitly relies on TxP and GaP to model spatial relationships between screen elements.

FG Training. After training perceivers, each of them has the ability to be aware of modality-specific signals (textual, graphical, and spatial content) on the screen. To equip FG with the ability to interpret task preferences from the question semantics and dynamically fuse each type of signal, we mix the training data with multiple task preferences in the *Step 4* training stage.

After training, the TGS-PFM is able to perceive and extract distinct GUI signals from the visual clues, and dynamically complete feature fusion based on task semantics.

4. Data Preparation

4.1. Post-Processing Based on Existing Data

In TGS-PFM, TxP extracts text signals from the screen, GaP captures graphical signals, and SaP identifies the spatial relationships between screen elements. To ensure the functional distinctiveness of each perceiver, we constructed specific training data for them.

***Text Aware Data* (TAD).** As *Step 1* in Tab. 1, we create TAD to guide the TxP to focus on textual signals within screens. We filter grounding task samples from the AMEX [11] dataset by using OCR tools to identify bounding boxes that contain only text. These purely textual targets are retained as *text2bbox* pairs[1]. Further, we exchange the function description and coordinates as *bbox2text* data.

[1]All boxes related are defined by $[x_{left}, y_{top}, x_{right}, y_{bottom}]$, scaled to $[0, 1000]$.

Training	Dataset	Task	Samples	Unlocked Params
Step 1	TAD	text2bbox bbox2text	160K	TxP LLM
Step 2	GAD	text2bbox bbox2text	187K	GaP LLM
Step 3	SAD	SRP	200K	SaP LLM
Step 4	TAD GAD SAD	text2bbox bbox2text SRP	*35K*	TGS-PFM LLM vision backbone
	SynD	SPE-QA MPE-QA	*48K*	
		Total	680K	

Table 1. Details of our Multi-stage Training Strategy (MTS). SynD refers to Synthetic Data (Sec. 4.2), while SRP, SPE-QA, and MPE-QA denote Spatial Relationship Prediction (Sec. 4.1), Single Perceiver Enhanced QA, and Multi-Perceiver Enhanced QA (Sec. 4.2), respectively. During training, we employ LoRA [25] to fine-tune the LLM or the vision backbone.

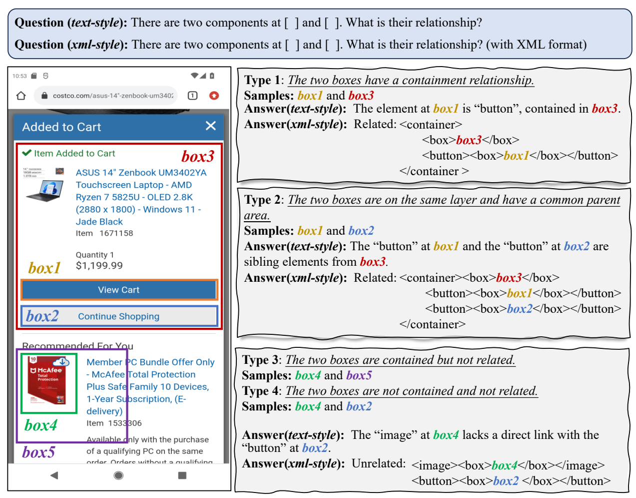

Figure 3. Examples of Spatial Relationship Prediction (SRP) task.

***Graphics Aware Data* (GAD).** In *Step 2*, to focus GaP on graphical information, *e.g.*, icons and widgets, we apply the same method of TAD, selecting samples without text in bounding box. Additionally, **we find that MLLMs struggle to perceive fine-grained visual information on screens.** To alleviate this weakness, we created a grounding dataset for small objects derived from AITW [50]. Specifically, we treat episodes as independent QA pairs and calculate the screen proportion of the target element within each pair using the formula $r = \frac{w \times h}{W \times H} \times 100\%$, where w and h represent the width and height of the box, and W and H represent the resolution of the screen image. Ultimately, we retain only those targets for which $r <= 0.3\%$.

***Spatial Aware Data* (SAD).** Previous research has focused on acquiring structured layout knowledge on the screen via explicit provision of code-like clues to the model [29, 34, 36] or via autoregressive reconstruction of the screen code to implicitly guide model learning [5]. **We introduce a novel task called Spatial Relationship Prediction (SRP), which explicitly models the spatial relationships between elements on the screen to guide MP-GUI in clearly perceiving the spatial context of GUI elements.**

As depicted in Fig. 3, the SRP task comprises four types. *Type 1*: containment relationships (one element is the parent node of another in the VH tree). *Type 2*: two elements share the same parent node and depth. **To prevent the model from making predictions based solely on the box coordinates in question**, we introduce two types of negative samples. *Type 3*[2]: two boxes have a containment relationship but no actual association with UI design. *Type 4*[3]: two boxes without either containment or association.

These tasks allow the SaP to perceive the visual, semantic and spatial relationships of screen elements more effectively. The SRP dataset is constructed using the original screen code files of images from the Semantic UI dataset [43]. For *Type 1* and *Type 2*, the root node is excluded, and only 1-hop connections within the VH tree are retained to ensure strong semantic and spatial correlations among boxes. More details of SRP are in Suppl.Mater.

4.2. Data Synthesis via MLLM

Unlike the vanilla MoE [16, 26, 37], which uses a router to control the information flow to experts, we introduce a task-oriented FG to dynamically fuse the output GUI-related signals according to task semantics (Sec. 3.1). To support FG training, different tasks with clear semantic preferences should be mixed together; thus, we propose a data synthesis pipeline with an advanced MLLM (Qwen2-VL-72B [55]). Specifically, we prompt the MLLM to generate two types of synthetic data, namely, ***Single Perceiver Enhanced Question Answering*** **(SPE-QA)** and ***Multi-Perceiver Enhanced Question Answering*** **(MPE-QA)**.

For **SPE-QA**, the data should have a strong semantic preference for one modality of content on the screen, such as OCR-related QA pairs (for TxP) and graphics-related captioning (for GaP) tasks. Since each GUI perceiver is pretrained on specific type of data (Sec. 4.1) and are therefore able to coarsely distinguish tasks with different semantics, we are hopeful that **SPE-QA** data can guide FG to selectively enhance the fusion of specific GUI signals.

The role of **MPE-QA** data is to further improve FG's semantic understanding and weight assignment capabilities by introducing semantic integration tasks that require FG to have the ability to collaborate with different perceivers. For this type of task, we prompt the MLLM to provide a fine-grained description of the screen content[4]. The generated output includes textual content, graphical elements, and coarse-grained positional information of elements. In order to synthesize multi-granularity screen perception data, we further introduce a task known as ***Local Description***. This task focuses on capturing the contextual information surrounding graphics to enhance local cognition. Specifically, we utilize an enhanced version of YOLO[5] to detect graphics on the screen, draw their bounding boxes on the image (similar to the Set-of-Marks [59] method), and subsequently input the image into the MLLM. We then prompt the MLLM to consider the surrounding content of the target area as much as possible when generating descriptions, thereby facilitating the inclusion of contextual associations in the output. See Suppl.Mater for more details.

5. Experiments

5.1. Experimental Setting

Basic GUI Understanding Benchmark. As described in Tab. 2, we extensively collect multiple public GUI-related datasets, including screen summarization [54], widget captioning [35], clickable prediction [52], grounding [7], question answering [5, 12, 23], aiming to comprehensively evaluate the basic GUI understanding ability of MLLMs. Details of the above datasets are in Suppl.Mater.

Screen Navigation Benchmark. Compared to basic GUI understanding tasks, screen navigation (*i.e.*, autonomous agent [15, 38]) is more challenging as it requires MLLM to have solid GUI understanding, decompose the user's goal into a series of subtasks to be completed as a trajectory, and continuously interact with different screenshots via specific operations like clicking, sliding, and typing. We select AITW [50] (mobile) and Mind2Web [18] (website) as benchmark, and treat screen navigation as a purely visual problem, using prompt, action spaces and data splitting consistent with previous literature [15].

Screen Grounding Benchmark. To evaluate the grounding ability of MLLMs on devices with different resolutions, in addition to RefExp [7], we also use ScreenSpot [15], which contains grounding tasks for text objects and icon/widget objects in three scenarios: mobile, desktop, and website.

Training Details. We initialize the Alignment Projector, vision backbone (InternViT-300M), and Word Embedding Layer&LLM (InternLM2.5-7B-chat) from InternVL2-

[2]We randomly select the elements and expand their boxes to generate negative samples. We limit the expanded box to have IoU ≤ 0.1 with the original box and IoU ≤ 0.3 with the box of its parent node in the VH tree.

[3]These pairs are randomly chosen from the VH tree file except for those included in *Type 1* and *Type 2*.

[4]We found that Qwen2-VL-72B [55] effectively performs OCR and has coarse but useful graphics perception. While its quality isn't on par with human supervision, it is still worth considering for training FG.

[5]We fine-tuned YOLOv8 using about 9k manually labeled in-house data, which mainly includes medical appointment registration, catering, funds, logistics and other scenes in Alipay.

Benchmark	Metric
Screen Analysis	
Widget Captioning(WC) [35]	CIDEr
Taperception(TP) [52]	F1
Screen Question-Answering	
ScreenQA(QA) [23]	ROUGE-L
ScreenQA Short(QAS) [5]	SQuAD F1
Complex ScreenQA(CQA) [5]	SQuAD F1
WebSRC(WS) [12]	SQuAD F1
Screen Grounding	
RefExp(RE) [7]	Acc@IoU=0.1
Screen Summarization	
Screen2Words(S2W) [54]	CIDEr

Table 2. Details of basic GUI understanding benchmark and their metric. The abbreviations of these datasets are given in brackets.

Method	Size	WC	S2W	RE	TP	WS	QA	QAS	CQA
Qwen-VL [8]	9.6B	84.1	100.2	36.6	83.5	57.3	78.9	69.1	54.9
MiniCPM-V 2.6 [60]	8B	110.5	107.3	48.5	80.4	85.2	76.3	77.3	71.5
Qwen2-VL [55]	7B	136.6	98.9	47.6	88.0	82.8	87.0	87.8	65.3
Llama 3.2-V [3]	11B	113.6	108.8	51.3	83.4	87.0	<u>88.4</u>	**91.6**	74.6
CogAgent [22]	18B	136.2	115.0	<u>73.3</u>	**88.4**	63.1	85.3	74.6	65.1
InternVL2 [14]	8B	140.6	<u>115.2</u>	71.7	86.7	**89.7**	84.2	89.2	<u>82.4</u>
InternVL2-P	8B	<u>142.8</u>	115.1	72.4	87.8	<u>89.6</u>	87.2	88.9	81.9
MP-GUI	8B	**151.0** +5.7%	**118.4** +2.8%	**83.0** +13.2%	<u>88.2</u> −0.2%	89.2 −0.6%	**88.6** +0.2%	<u>90.5</u> −1.2%	**84.3** +2.3%

Table 3. Comparison of MP-GUI with other advanced MLLMs. **Bold** represents the best results, <u>underlined</u> represents the second best results, and the improvement of each task is compared with the second best method. We conduct multi-task fine-tuning for all MLLMs individually on the basic GUI understanding benchmark.

Method	ScreenAI [5]	Spotlight [30]	Ferret-UI [62]	Pix2Struct [29]	MP-GUI
#Samples	383.5M	2.69M	0.84M	80M	0.68M
WC	<u>156.4</u>	141.8	142.0	136.7	**156.5**
S2W	<u>120.8</u>	106.7	115.6	109.4	**121.4**
RE	**86.3**	-	-	-	<u>84.7</u>
TP	-	<u>88.4</u>	78.4	-	**88.7**
WS	<u>87.2</u>	-	-	-	**90.1**
QA	**91.9**	-	-	-	<u>88.7</u>
QAS	**94.6**	-	-	-	<u>92.7</u>
CQA	-	-	-	-	**87.7**

Table 4. Comparison of MP-GUI with GUI-specific methods. The results we show are from single-task fine-tuning and ***#Samples*** means the count of GUI-related training data used in each method.

8B [14], while training other modules (TGS-PFM) from scratch. Adhering to our MTS procedure (Sec. 3.2), we set the learning rates at 1e-5 for *Step 1–3*, 5e-6 for *Step 4*, and 5e-4 for benchmark fine-tuning. LoRA [25] was applied to both the LLM and vision backbone, with rank and alpha values of 8 and 16 for *Steps 1–3*, and 64 and 128 for *Steps 4* and benchmark fine-tuning. Each stage involved training for 1 epoch using the AdamW optimizer, with baseline MLLMs following their official fine-tuning protocols, also for 1 epoch of multi-task fine-tuning utilizing LoRA [25][6].

For screen navigation, we apply LoRA [25] with rank and alpha value set at 128 and 256, a learning rate of 2e-5, and perform 3 epochs on AITW [50] and Mind2Web [18]. All experiments were performed on 8 A100 GPUs with a global batch size of 64.

5.2. Main Results

Gains of Instruction-tuning for Domain-specific Data. Instruction-tuning a MLLM using domain-specific or task-related data can enable the model to quickly learn domain knowledge [15]. We instruction-tune the vanilla InternVL2 [14] using the GUI-related training data in Tab. 1 to obtain InternVL2-P, which can be considered as a GUI knowledge enhancement in-domain model. Next, we keep InternVL2-P with the same fine-tuning setting as MP-GUI on each benchmark (Sec. 5.1).

It can be observed that for both screen grounding (Fig. 4 and Tab. 5) and screen navigation (Tab. 6 and Tab. 7), InternVL2-P has achieved considerable gains compared to InternVL2 [14]. However, in Tab. 3, for tasks like CQA and QAS that rely on the foundational model's reasoning abilities, InternVL2-P is inferior to InternVL2 [14]. Indicating that instruction-tuning may put MLLMs at risk of weakened generic abilities. Notably, MP-GUI shows favorable performance over InternVL2-P in all the above benchmarks.

According to the above results, we argue that **relying solely on domain-specific data for instruction-tuning enables the MLLM to perceive only superficial GUI information**. Our special designs, which provide additional GUI-tailored visual clues, effectively enhance MLLM in solving GUI-related tasks.

Basic GUI Understanding. In Tab. 3, MP-GUI outperforms baselines on most tasks and ranks second on TP, WS, and QAS. Notably, our MP-GUI exceeds the second-best methods by 5.7% on WC and 13.2% on RE, which demonstrates the excellent spatial location awareness of elements on the screen. Compared to Llama 3.2-V(11B) [3] and CogAgent(18B) [22], MP-GUI remains competitive with less parameters, exhibiting only a decrease of 0.2% and 1.2% on TP and QAS tasks. In Tab. 4, we list current advanced GUI-specific methods [5, 29, 30, 62]. Compared with these methods that require a large amount of GUI-related pre-training data (*e.g.*, close-sourced ScreenAI [5] with 383.5M pre-training samples), our method leveraging only 0.68M GUI-specific training samples outperforms them on most

[6]Our multi-task fine-tuning on the basic GUI understanding benchmark reveals that: (*i*) baseline methods achieve optimal performance after 1 epoch, while additional training generally don't show further performance gains; (*ii*) fine-tuning using LoRA outperforms full parameter fine-tuning.

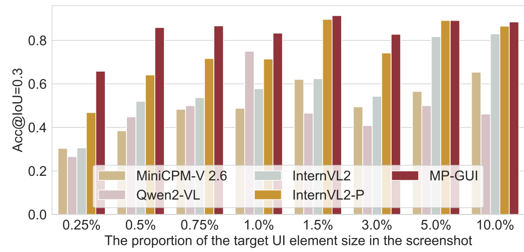

Figure 4. Comparison of the grounding results of various methods on UI elements of different sizes under Ref-Exp [7]. The *proportion* of $k\%$ indicates that $\frac{w \times h}{W \times H} \leq k\%$, where w and h represent the width and height of UI elements, and W and H represent the screen resolution.

Method	Size	Mobile		Desktop		Web		Avg.
		Text	Icon/Widget	Text	Icon/Widget	Text	Icon/Widget	
Llama 3.2-V [3]	11B	14.7%	5.7%	9.3%	4.3%	4.3%	4.4%	7.1%
GPT-4V[49]†	-	22.6%	24.5%	20.2%	11.8%	9.2%	8.8%	16.2%
Fuyu [9]†	8B	41.0%	1.3%	33.0%	3.6%	33.9%	4.4%	19.5%
InternVL2 [14]	8B	74.0 %	25.8%	54.6%	27.1%	38.3%	31.6%	41.9%
CogAgent [22]†	18B	67.0%	24.0%	**74.2%**	20.0%	**70.4%**	28.6%	47.4%
SeeClick [15]†	9.6B	78.0%	52.0%	72.2%	30.0%	55.7%	32.5%	53.4%
InternVL2-P	8B	83.2%	52.0%	63.4%	43.6%	47.0%	41.3%	55.1%
MP-GUI	8B	**86.8%**	**65.9%**	70.8%	**56.4%**	58.3%	**46.6%**	**64.1%**

Table 5. Zero-shot grounding performance on ScreenSpot [15]. The best results in each column are **bold**. † means the results come from SeeClick [15] and pink color indicates the method we compare.

tasks and remains competitive in others.

These results show that MP-GUI has advanced GUI understanding capabilities, especially in tasks with graphical content (*e.g.*, WC, RE, and CQA). Our special model architecture and training recipe (*i.e.,* MTS) can guide MLLM to learn GUI knowledge effectively with limited data.

Screen Grounding. The significance of MLLMs' grounding for GUI tasks was explored in prior literature [15]. Here, we comprehensively evaluate the grounding ability of MP-GUI from: ***(i)* fine-grained perception of small-sized objects**; ***(ii)* perception of text and graphical objects (icon/widget) on different-resolution devices**.

In Fig. 4, it is evident that most MLLMs struggle to accurately ground UI elements of smaller size. Thanks to the training on GAD data (which includes *text2bbox* and *bbox2text* tasks involving custom small-size icons/widgets) and our dual-visual-clues framework (Sec. 4.1), MP-GUI achieves the best performance especially for the small-sized objects (with a *proportion* $\leq 1\%$). Notably, InternVL2-P has made significant gains compared to InternVL2 [14], verifying the effectiveness of the domain-specific data we constructed. However, **on small-size objects, InternVL2-P is still weaker than MP-GUI, demonstrating the effectiveness of our model aspect optimization.**

In Tab. 5, our MP-GUI consistently attains competitive results in zero-shot grounding across different device scenarios [15]. Notably, it shows leading performance in grounding graphical objects (icon/widget), achieving significant gains compared to InternVL2 [14]. Moreover, our training data only contains mobile resolution images (Sec. 4), and do not use website and desktop resolution images to provide priors [15, 38], showing that **there are generic GUI patterns across different device scenarios our MP-GUI can perceive**.

Screen Navigation. The generic abilities of foundational models, such as visual perception and reasoning, are crucial for navigation tasks. In the AITW [50] benchmark (Tab. 6), compared to SeeClick [15], which is based on Qwen-VL [8]

Method	General	Install	G.Apps	Single	WebShop	Overall
GPT-4V [57]	41.7	42.6	49.8	72.8	45.7	50.5
Qwen-VL [8]	49.5	59.9	46.9	64.7	50.7	54.3
OmniParser [45]	48.3	57.8	51.6	**77.4**	52.9	57.7
SeeClick [15]	54.0	66.4	54.9	63.5	57.6	59.3
InternVL2 [14]	58.1	65.3	56.8	68.7	61.1	62.0
ShowUI [38]	63.5	72.3	**66.0**	72.3	65.8	68.3
InternVL2-P	61.2	70.3	61.6	74.6	65.1	66.6
MP-GUI	**63.7**	**74.3**	65.3	75.4	**67.2**	**69.2**

Table 6. Performance of Screen Navigation on AITW [50]. The pink color indicates the method we compare.

Method	Cross-Task			Cross-Website			Cross-Domain		
	Ele.Acc	Op.F1	Step.SR	Ele.Acc	Op.F1	Step.SR	Ele.Acc	Op.F1	Step.SR
InternVL2 [14]	18.8	87.4	16.7	17.6	85.8	14.5	13.9	87.0	12.0
CogAgent [22]	22.4	53.0	17.6	18.4	42.4	13.4	20.6	42.0	15.5
SeeClick [15]	28.3	87.0	25.5	21.4	80.6	16.4	23.2	84.8	20.8
GPT-4 [2]	41.6	60.6	36.2	35.8	51.1	30.1	37.1	46.5	26.4
ShowUI [38]	39.7	88.0	36.9	**41.0**	83.6	**34.2**	**38.9**	85.3	**34.1**
InternVL2-P	27.4	87.8	24.0	27.4	86.1	23.1	24.3	87.1	21.1
MP-GUI	**42.1**	**89.0**	**38.1**	39.4	**87.1**	32.9	37.6	**87.4**	33.7

Table 7. Performance of Screen Navigation on Mind2Web [18]. The pink color indicates the method we compare.

and achieved an 8.3% gain, and ShowUI [38], which is based on Qwen2-VL [55] and provides a 1.2% improvement, our MP-GUI achieves a 10.4% gain compared to the foundational model InternVL2 [14]. For ShowUI [38], we choose the version without visual history. In the Mind2Web [18] benchmark (Tab. 7), MP-GUI achieves leading results on action prediction (Op.F1) and competitive results on both click-element location (Ele.Acc) and step success rate (Step.SR). Compared with InternVL2 [14], our MP-GUI still achieved significant gains.

5.3. Ablation Study

In Tab. 8, we perform ablation studies evaluating (1) the impact of the FG, (2) the effects of different perceivers and (3) the gains of MTS.

Method	WC	S2W	RE	TP	WS	QA	QAS	CQA
w/o FG†	142.1 (-6.3%)	117.8 (-0.5%)	76.8 (-8.1%)	87.9 (-0.3%)	87.2 (-2.3%)	87.3 (-1.5%)	89.1 (-1.6%)	80.7 (-4.5%)
w/o FG‡	143.4 (-5.3%)	116.7 (-1.5%)	77.5 (-7.1%)	88.1 (-0.1%)	89.3 (+0.1%)	88.0 (-0.7%)	89.3 (-1.3%)	82.6 (-2.1%)
w/o TxP	142.6 (-5.9%)	115.2 (-2.8%)	78.8 (-5.3%)	88.3 (+0.1%)	89.1 (-0.1%)	87.5 (-1.3%)	89.4 (-1.2%)	80.8 (-4.3%)
w/o GaP	143.1 (-5.5%)	116.0 (-2.1%)	79.5 (-4.4%)	88.1 (-0.1%)	89.3 (+0.1%)	87.7 (-1.0%)	89.5 (-1.1%)	80.3 (-5.0%)
w/o SaP	141.9 (-6.4%)	116.2 (-1.9%)	78.4 (-5.9%)	88.2 (0.0%)	89.0 (-0.2%)	87.6 (-1.1%)	89.4 (-1.2%)	80.3 (-5.0%)
w/o MTS	148.3 (-1.8%)	117.0 (-1.2%)	82.4 (-0.7%)	87.2 (-1.1%)	86.9 (-2.6%)	87.4 (-1.4%)	88.3 (-2.4%)	83.5 (-1.0%)
MP-GUI	151.0	118.4	83.0	88.2	89.2	88.6	90.5	84.3

Table 8. Ablation study results. *w/o* FG† indicates that we don't pre-train FG (without implementing *Step 4* in the MTS). *w/o* FG‡ indicates that we remove FG and directly use the mean of outputs from the three perceivers for feature fusion. *w/o* MTS means keeping the same settings of MP-GUI except that we don't use step-by-step MTS (Sec. 3.2) but collect all training data for end-to-end training TGS-PFM.

Fusion Gate. In Sec. 3.1, we introduce FG module that dynamically assign weights to different perceivers according to task semantics and global visual signal. In this study, we examine the effect of FG on performance by modifying both the training recipe and the architecture. For *w/o* FG†, the results show that it is necessary to utilize task-oriented data to achieve task-specific semantic awareness of FG. Meanwhile, for *w/o* FG‡, the removal of FG weakens the performance of MP-GUI, especially in WC, RE, and CQA tasks, confirming FG's effectiveness.

GUI-tailored Perceivers. In Tab. 8, for S2W, QA, and QAS that prefer to focus on the text information from screens, the gains decrease by 2.8%/1.3%/1.2% respectively without TxP. For WC and RE tasks, MP-GUI needs to clarify the spatial context among elements. The gains on WC and RE decrease by 6.4% and 5.9% respectively when without SaP. For challenging CQA tasks, it is necessary to comprehensively localize the target element and perceive the spatial contextual information around it to assist MLLM in reasoning. Thus, both GaP and SaP are crucial, and removing either of them will reduce performance by 5.0%.

MTS. We conduct pre-training on TGS-PFM (Sec. 3.1) by using the same data (Tab. 1) as that of MP-GUI. Instead of adopting our step-by-step MTS recipe (Sec. 3.2), we opt for the end-to-end training mode. As shown in Tab. 8, this end-to-end method results in performance degradation when compared to MTS, validating the gains of MTS.

5.4. Qualitative Analysis

In Fig. 5, we display several examples on basic GUI understanding tasks (Sec. 5.1). The results demonstrate that our MP-GUI effectively understands the implicit knowledge on the screen, including graphics, text, and their spatial relationships (case **a**). For queries that necessitate deeper reasoning (case **b**), MP-GUI can effectively locates and associates details from different areas to generate accurate answers. Furthermore, MP-GUI demonstrates a comprehensive ability to capture essential modality details of the screen (case **c**). Notably, MP-GUI's advanced GUI perception and understanding capabilities can alleviate the hallucination problem often encountered in MLLMs (case **d**).

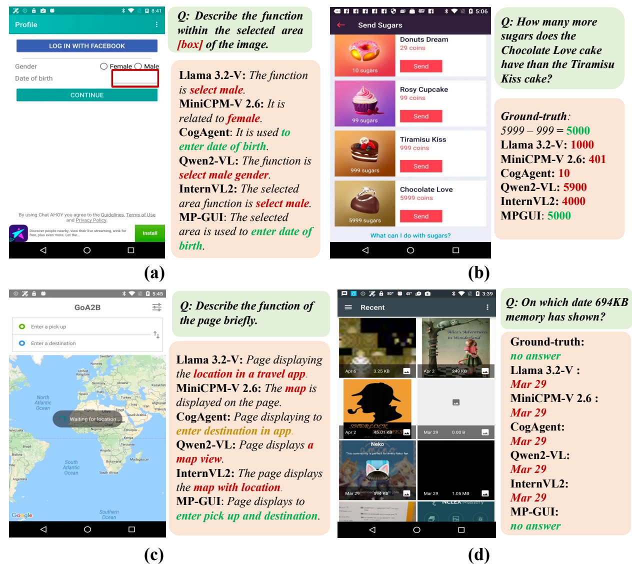

Figure 5. Case studies on basic GUI understanding benchmark (Sec. 5.1). Accurately described answer is marked in **green**, while inaccurately and incompletely described ones in **red** and **orange**.

More results and detailed discussion are in Suppl.Mater.

6. Conclusion

In this paper, we center on enhancing MLLM in GUI scenarios and present MP-GUI, a dual-visual-clues model. It uses three GUI-specific perceivers to extract modality signals from the screen and a FG module to fuse them based on task semantics, generating additional GUI-tailored visual clues to enhance MLLM's GUI visual perception. We also introduce a novel SRP task for explicitly modeling GUI elements' spatial relationships and an automated synthetic data pipeline for FG training. Extensive experiments confirm our designs significantly enhance MLLM's GUI understanding, facilitating the improvement of various downstream tasks.

Acknowledgments This work was supported by the National Natural Science Foundation of China (Grant No.6237 2408). This work was also supported by Ant Group.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2, 7

[3] Meta AI. Llama 3. *CoRR*, 2024. Accessed: 2024-11-12. 1, 2, 6, 7, 3, 5

[4] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. 2

[5] Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Carbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. Screenai: A vision-language model for UI and infographics understanding. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 3058–3068. ijcai.org, 2024. 2, 3, 5, 6, 7, 8

[6] 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 *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 1705–1712, 2021. 1, 2

[7] 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 *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 1705–1712. ijcai.org, 2021. 2, 5, 6, 7, 3

[8] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. 2, 6, 7, 3

[9] Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. Fuyu-8b: A multimodal architecture for ai agents. *CoRR*, 2023. Accessed: 2024-11-12. 7

[10] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024. 2

[11] Yuxiang Chai, Siyuan Huang, Yazhe Niu, Han Xiao, Liang Liu, Dingyu Zhang, Peng Gao, Shuai Ren, and Hongsheng Li. Amex: Android multi-annotation expo dataset for mobile gui agents. *arXiv preprint arXiv:2407.17490*, 2024. 4

[12] Xingyu Chen, Zihan Zhao, Lu Chen, JiaBao Ji, Danyang Zhang, Ao Luo, Yuxuan Xiong, and Kai Yu. WebSRC: A dataset for web-based structural reading comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4173–4185, 2021. 2, 5, 6, 3

[13] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023. 2

[14] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 1, 2, 3, 6, 7, 5

[15] 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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9313–9332. Association for Computational Linguistics, 2024. 1, 2, 5, 6, 7

[16] Damai Dai, Chengqi Deng, Chenggang Zhao, R.x. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y.k. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1280–1297, 2024. 3, 5

[17] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pages 845–854, 2017. 2

[18] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samual Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 2, 3, 5, 6, 7

[19] Christian Derksen, Cherif Branki, and Rainer Unland. Agent. gui: A multi-agent based simulation framework. In *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 623–630. IEEE, 2011. 1

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[21] W Keith Edwards, Elizabeth D Mynatt, and Kathryn Stockton. Access to graphical interfaces for blind users. *Interactions*, 2(1):54–67, 1995. 1

[22] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for GUI agents. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 14281–14290. IEEE, 2024. 2, 3, 6, 7

[23] Yu-Chung Hsiao, Fedir Zubach, Gilles Baechler, Victor Carbune, Jason Lin, Maria Wang, Srinivas Sunkara, Yun Zhu, and Jindong Chen. Screenqa: Large-scale question-answer pairs over mobile app screenshots. *arXiv preprint arXiv:2209.08199*, 2022. 2, 3, 5, 6, 1

[24] Anwen Hu, Yaya Shi, Haiyang Xu, Jiabo Ye, Qinghao Ye, Ming Yan, Chenliang Li, Qi Qian, Ji Zhang, and Fei Huang. mplug-paperowl: Scientific diagram analysis with the multimodal large language model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6929–6938, 2024. 1

[25] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 4, 6

[26] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *https://arxiv.org/abs/2401.04088*, 2024. 5

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2

[28] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 2

[29] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, pages 18893–18912. PMLR, 2023. 1, 2, 5, 6

[30] Gang Li and Yang Li. Spotlight: Mobile UI understanding using vision-language models with a focus. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 2, 6

[31] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 2

[32] Junxian Li, Di Zhang, Xunzhi Wang, Zeying Hao, Jingdi Lei, Qian Tan, Cai Zhou, Wei Liu, Yaotian Yang, et al. Chemvlm: Exploring the power of multimodal large language models in chemistry area. *arXiv preprint arXiv:2408.07246*, 2024. 1, 2

[33] Toby Jia-Jun Li, Lindsay Popowski, Tom Mitchell, and Brad A Myers. Screen2vec: Semantic embedding of gui screens and gui components. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021. 1

[34] Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. Mapping natural language instructions to mobile UI action sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8198–8210, Online, 2020. Association for Computational Linguistics. 2, 5

[35] Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. Widget captioning: Generating natural language description for mobile user interface elements. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5495–5510, 2020. 2, 3, 5, 6, 1

[36] Yang Li, Gang Li, Xin Zhou, Mostafa Dehghani, and Alexey Gritsenko. Vut: Versatile ui transformer for multimodal multi-task user interface modeling. *arXiv preprint arXiv:2112.05692*, 2021. 5

[37] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Jinfa Huang, Junwu Zhang, Yatian Pang, Munan Ning, et al. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024. 3, 5

[38] Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent. *arXiv preprint arXiv:2411.17465*, 2024. 1, 2, 5, 7

[39] Mario Linares-Vásquez, Kevin Moran, and Denys Poshyvanyk. Continuous, evolutionary and large-scale: A new perspective for automated mobile app testing. In *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 399–410. IEEE, 2017. 1

[40] Fenglin Liu, Tingting Zhu, Xian Wu, Bang Yang, Chenyu You, Chenyang Wang, Lei Lu, Zhangdaihong Liu, Yefeng Zheng, Xu Sun, et al. A medical multimodal large language model for future pandemics. *NPJ Digital Medicine*, 6(1): 226, 2023. 2

[41] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 2

[42] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee.

Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 2

[43] Thomas F. Liu, Mark Craft, Jason Situ, Ersin Yumer, Radomír Mech, and Ranjitha Kumar. Learning design semantics for mobile apps. In *The 31st Annual ACM Symposium on User Interface Software and Technology, UIST 2018, Berlin, Germany, October 14-17, 2018*, pages 569–579. ACM, 2018. 2, 5, 3

[44] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[45] Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. Omniparser for pure vision based gui agent. *arXiv preprint arXiv:2408.00203*, 2024. 1, 7

[46] Chuwei Luo, Yufan Shen, Zhaoqing Zhu, Qi Zheng, Zhi Yu, and Cong Yao. Layoutllm: Layout instruction tuning with large language models for document understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15630–15640, 2024. 1

[47] Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. Coco-agent: A comprehensive cognitive mllm agent for smartphone gui automation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9097–9110, 2024. 1

[48] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. 2

[49] OpenAI. Gpt-4v(ision) system card. *CoRR*, 2023. 7

[50] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36:59708–59728, 2023. 2, 3, 4, 5, 6, 7

[51] Anne Spencer Ross, Xiaoyi Zhang, James Fogarty, and Jacob O Wobbrock. Examining image-based button labeling for accessibility in android apps through large-scale analysis. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility*, 2018. 2

[52] Eldon Schoop, Xin Zhou, Gang Li, Zhourong Chen, Bjoern Hartmann, and Yang Li. Predicting and explaining mobile ui tappability with vision modeling and saliency analysis. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2022. 5, 6, 1

[53] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024. 1

[54] Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. Screen2words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 498–510, 2021. 2, 3, 5, 6

[55] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 1, 2, 5, 6, 7, 3

[56] Dexuan Xu, Yanyuan Chen, Jieyi Wang, Yue Huang, Hanpin Wang, Zhi Jin, Hongxing Wang, Weihua Yue, Jing He, Hang Li, et al. Mlevlm: Improve multi-level progressive capabilities based on multimodal large language model for medical visual question answering. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4977–4997, 2024. 2

[57] An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, et al. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. *arXiv preprint arXiv:2311.07562*, 2023. 7

[58] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 2

[59] 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. 5

[60] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 1, 2, 3, 6

[61] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023. 1, 2

[62] 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*, pages 240–255. Springer, 2025. 1, 2, 6

[63] Xian Zhang, Haokun Wen, Jianlong Wu, Pengda Qin, Hui Xue', and Liqiang Nie. Differential-perceptive and retrieval-augmented mllm for change captioning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 4148–4157, 2024. 1

[64] Tianming Zhao, Chunyang Chen, Yuanning Liu, and Xiaodong Zhu. Guigan: Learning to generate gui designs using generative adversarial networks. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 748–760. IEEE, 2021. 1

[65] Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu. Mova: Adapting mixture of vision experts to multimodal context. *arXiv preprint arXiv:2404.13046*, 2024. 3