ICONBANK: MINING HARD SAMPLES VIA VISUAL CONCEPTS FOR DATA-EFFICIENT GUI GROUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

The core capability of a Graphical User Interface (GUI) agent based on a Multimodal Large Language Model (MLLM) relies on accurate GUI grounding, which precisely locates actionable elements in screenshots according to instructions. The core challenges in traditional fine-tuning are **low data efficiency and small object grounding**. Supervised Fine-Tuning (SFT), as a mainstream approach, requires massive datasets. While rule-based Reinforcement Fine-Tuning (RFT) offers improvements, it still fails to accurately filter useful data from overwhelming redundancy. Most of the samples are easy to learn, and the performance of the model is barely improved. Inspired by the human learning mechanism of "**Problem-Type-Specific Retraining**", this paper constructs a decoupled visual concept library to acquire high-value retraining resources. Based on this library, we propose **IconBank**, a hard sample mining framework. Through this framework, our key finding is that only a minimal number of carefully selected difficult samples can achieve performance comparable to, or even better than, training with massive data. Specifically, we first extract operable elements from multiple open-source GUI datasets to build a unified decoupled visual concept library (IconBank), where "Icon" is redefined as pure visual atomic concepts stripped of context, background, and layout. Next, we search for similar elements through the decoupled visual concept library and finally select targeted practice samples to form a minimal refined training set. Experimental results show that a 3B model trained on only 2K samples achieves a score of 51.7% on the ScreenSpot-Pro benchmark, surpassing most 7B models.This significant effectiveness verifies the assumption of massive redundancy in GUI data and reveals that data quality (diversity and challenge) is far superior to quantity.

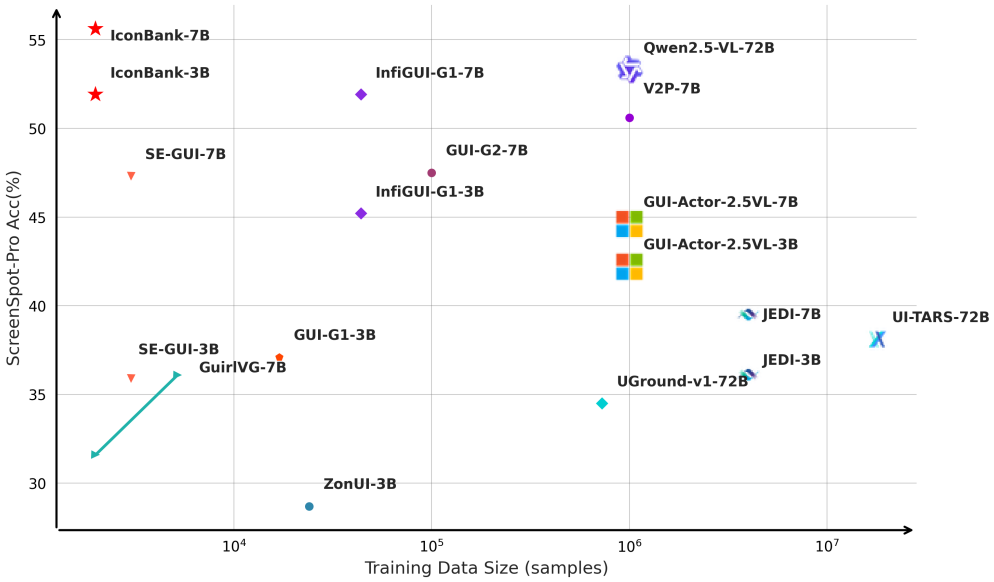

Figure 1: GUI Agent Performance vs Training Data Scale on Grounding Benchmark.

1 INTRODUCTION

With the development of Multi-modal large language models (MLLMs) in the field of human-computer interaction, Graphical User Interface (GUI) agents for office automation and unattended operation face core challenges.. The core ability of GUI agents depends on GUI grounding. It can accurately locate interactive elements (Lu et al., 2024) (such as buttons, input fields, etc.) in GUI by natural language instructions. The agent needs to recognize very small actionable elements from high-resolution images and complex backgrounds, which requires the ability of the MLLM to understand language instructions in the domain, as well as fine visual localization capabilities (Qin et al., 2025).

The current mainstream GUI grounding methods are mainly Supervised Fine-Tuning (SFT) and Reinforcement Fine-Tuning (RFT). Most of the current Supervised Fine-Tuning methods rely on large-scale labeled data. This pure vision-based method faces **two core challenges**: one is low data efficiency and high computational cost, and the other is the difficulty in grounding small objects. Specifically, **at the data level**, manually annotating millions of samples costs a lot of time and manpower, and it is difficult to ensure the quality of the annotations generated by the model. Moreover, large-scale data training typically requires days or even weeks,, which is difficult to meet the needs of low-cost deployment and fast iteration (e.g., small and medium enterprise development, edge device deployment) (Li et al., 2025a). **At the perception level**, compared with text grounding, Multimodal Large Language Models (MLLMS) find it more challenging to accurately grounding small targets composed of visual elements such as ICONS, as these targets often lack sufficient contextual semantic information. These challenges make it difficult to apply GUI agents in real-world scenarios. In contrast, rule-based reinforcement learning or reinforcement fine-tuning can effectively optimize the model with only thousands of samples(Yuan et al., 2025). These methods improve performance while reducing samples through heuristic exploration, but still face the problems of difficulty in grounding small objects.

In response to the above challenges, this paper asks the core question: Is it necessary to use massive data(up to millions of samples)? And does every sample contribute significantly to the performance after fine-tuning? Our inspiration comes from the human learning mechanism of "**problem-type-specific retraining**". In the learning process, humans will summarize the types of questions and consolidate the training of "error-prone" questions rather than solving problems indiscriminately without distinction. This kind of targeted training makes learning efficient. Also in the GUI grounding task, we can achieve great results by locating "error-prone", fine-grained visual concepts (e.g., ICONS of a specific style, elements of a specific function) and mining similar difficult examples around these concepts.

To achieve this goal, we propose **IconBank**, a framework for the extraction of difficult samples based on a library of decoupled visual concepts. The core idea is to first remove the contextual interference of GUI elements (e.g., background, layout), build a unified "visual atomic concept library", and then find similar elements (i.e., failed visual concepts) based on the library to form a refined training set. Experimental results show the effectiveness of IconBank: MLLM trained with only **1k hard samples and 1k random samples** achieves comparable accuracy on ScreenSpot Cheng et al. (2024) tasks to models trained with 1M original samples, with a 14% improvement on ScreenSpot-Pro Li et al. (2025b). This confirms the hypothesis that there is a large amount of redundancy in the GUI data and further confirms the effectiveness of our selection of difficult samples through comprehensive ablation experiments.

The main contributions of this paper can be summarized in the following four points.

- We propose IconBank, a difficult sample mining framework, which obtains difficult samples of the same type by decoupled visual concepts from precise matching, and provides a new paradigm for few-shot training by focusing on "error-prone" questions with more practice.
- We provide a new perspective that difficult samples do not only come from wrong questions, but also find the corresponding question type of the wrong question and repeatedly train the whole question type. This method not only improves the generalization ability of the model, but also enhances the semantic understanding of specific visual elements.
- Experiments show IconBank achieves 89.6% on ScreenSpot, 91.7% on ScreenSpot-V2 and 55.7% on ScreenSpot-Pro using only 2k samples, These results reveal substantial optimiza-

tion potential at the data level in contemporary GUI research. Our work thereby establishes a new direction for developing efficient and lightweight GUI agents through data-centric optimization strategies.

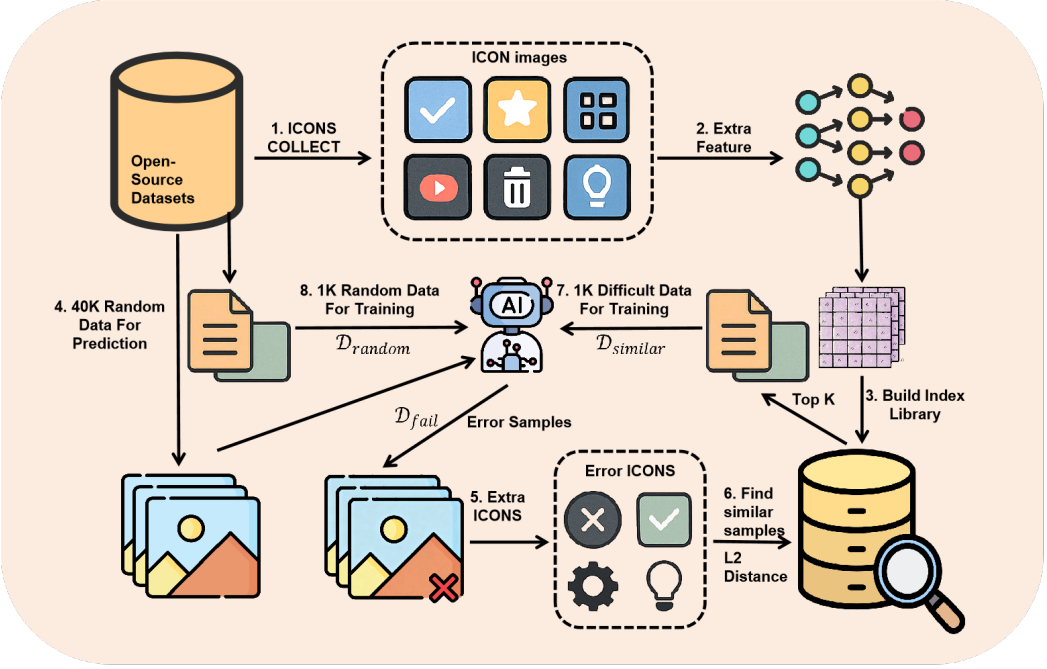

Figure 2: **Overview of IconBank framework pipeline**. The IconBank framework operates in two main stages. First, a Decoupled Visual Concept Library (IconBank) is constructed by cropping actionable elements (e.g., buttons, icons) from diverse GUI dataset. In the second stage, the Hard Sample Mining Pipeline is executed: (1) a base model predicts on a large candidate dataset, and the failed predictions ($\mathcal{D}_{\text{fail}}$) are collected; (2) for each failed element, IconBank retrieves the top-5 most visually similar concepts to form a set of challenging analogues ($\mathcal{D}_{\text{similar}}$); (3) these hard samples are merged with a small number of random samples ($\mathcal{D}_{\text{random}}$) to form a refined training set ($\mathcal{D}_{\text{train}}$).

2 Related Works

2.1 GUI Grounding Method

In recent years, GUI agents based on multi-modal large models have made significant progress (Agashe et al., 2024; 2025). Earlier studies such as CogAgent Hong et al. (2024) and Ferret-UI You et al. (2024) improved the model's understanding of mobile UI by fine-tuning visual commands. AppAgent Zhang et al. (2025a) and AppAgentX Jiang et al. (2025) further explore the practical application of multi-modal agents in smartphone operation. Aguvis Xu et al. (2024) explores the path of purely visual autonomous GUI interaction. In terms of visual localization, Spotlight Li & Li (2023) proposes a focusing mechanism to enhance the model's attention to UI elements. Aria-UI Yang et al. (2024) and GUI-G1 Zhou et al. (2025) attempt to optimize the visual localization policy through reinforcement learning. InfiGUI-G1 Liu et al. (2025) introduces adaptive exploration strategy optimization, which shows strong positioning ability in complex interfaces. In addition, GUI-G² Tang et al. (2025) proposes a Gaussian reward modeling method, and GUI-Actor Wu et al. (2025) proposes a coordinate-free visual localization method, which further improves the localization accuracy. Phi-Ground Zhang et al. (2025b) advances the field from the direction of improving the perception ability of the model.

2.2 Data-efficient Training and Reinforcement Learning

Traditional supervised fine-tuning methods rely on large-scale labeled data, which has the problems of high labeling cost and low training efficiency Pan et al. (2024). In order to improve the efficiency

of data utilization, researchers have widely explored the fine-tuning methods based on reinforcement learning. UI-R1 Lu et al. (2025b) optimize agent behavior through multiple rounds of interactive training. ARPO Lu et al. (2025a) introduces an experience replay mechanism to further improve sample efficiency. DigiRL Bai et al. (2024) and E-Ant Wang et al. (2024a) focus on training devices to control agents in real environments. Efficient Agent Training (He et al., 2025) and Enhancing Visual Grounding (Yuan et al., 2025) also optimize the training efficiency from different perspectives. GuirlVG Kang et al. (2025) and Gui-R1 Luo et al. (2025) explore the motivation method of GUI visual positioning based on R1 style.

2.3 Data Selection

Recently, some work has begun to focus on data filtering and difficult sample mining. Less is More Chen et al. (2025a) proposes context-aware interface simplification to reduce redundant information. InfiGUI-G1 Liu et al. (2025) selects data by eliminating easily identifiable samples. ZonUI-3B Hsieh et al. (2025), as a lightweight SFT model representative, tries to select effective data through platform diversity and resolution diversity. These methods all reflect the idea of "data quality is better than quantity", which is consistent with the starting point of this paper, but do not explicitly build a visual atomic concept library to achieve accurate hard sample matching.

2.4 GUI Grounding Dataset

Rich datasets have driven the development of the GUI field. RICO Deka et al. (2017) is an early dataset of mobile applications. OmniAct Kapoor et al. (2024) provides data for a multimodal general agent that supports both desktop and web pages. WebArena Zhou et al. (2024) and WebCanvas Pan et al. (2024) offer a real web environment for building and evaluating agents. Datasets such as OS-Atlas Wu et al. (2024), ShowUI Lin et al. (2025) and UGround Gou et al. (2024) have contributed abundant GUI screenshot and annotation resources, laying the foundation for model training in the GUI Grounding task.

In conclusion, existing research has made significant progress in aspects such as supervised fine-tuning training paradigms based on large-scale data, efficient reinforcement learning training paradigms, and the construction of large-scale datasets. Some work has also begun to focus on enhancing learning efficiency through data screening. However, these methods have yet to make breakthroughs in the core challenge of data efficiency. Most methods fail to deeply explore the redundancy within the data and systematically extract samples that are truly difficult for the model to learn.

3 Methodology

As illustrated in Figure 2, this paper constructs a new framework, IconBank, for mining difficult samples by decoupling the visual concept library to achieve efficient training of GUI Grounding tasks. We will introduce this framework in two stages. First, we elaborate on how to build IconBank in (Section 3.1). Then in (Section 3.2), we will use IconBank to design a pipeline for mining difficult samples.

3.1 Build the Decoupled Visual Concept Library

GUI screenshots contain rich actionable visual elements such as ICONS, components, and text, which usually distract the model's attention during the visual positioning process. The current mainstream MLLMS have a stronger ability to understand text than other visual elements. Therefore, to enhance the semantic understanding ability of non-text elements, IconBank is composed of the smallest operable elements such as ICONS and components, with their surrounding layout, background and text content removed. Each "icon" in IconBank is an interactive element (such as buttons, checkboxes, sliders, etc.). We collect GUI element images from multiple public datasets (Ugorund, showUI, OS-Atlas-data, OmniAct, AMEX), which come from various platforms such as desktop systems, web pages, and mobile applications, ensuring the diversity of the data. Each element is cropped from the original screenshot using bounding box annotations.

We use the Pre-trained ResNet-50 He et al. (2016) backbone to extract feature vectors for each element, with a feature dimension of 2048. Index each feature using the IndexFlatL2 method of

the Faiss Douze et al. (2024) vector database. The index id is mapped to the original screenshot, and the original data can be located through the index id. In addition, we have attempted to extract features from the original images of public datasets and establish an index library. The similar images retrieved through this index are usually those with similar layouts or interfaces, making it difficult to precisely identify the erroneous elements. Searching through IconBank is a more precise search, avoiding the situation where the found data is irrelevant to incorrect samples.

3.2 Hard Sample Mining Pipeline

This stage aims to mine samples that are challenging and underlearned by the current basic model, thereby forming a targeted training set. For this purpose, we will implement it in three phases.

Step 1: Obtain the failed samples. We consider the collected open-source dataset as the candidate set $\mathcal{D}_{\text{candidate}}$, which contains over one million samples. For each sample (I, T, B_{gt}), it includes screenshot I, text instruction T, and real bounding box B_{gt}. We randomly select N samples ($N = 40,000$) from the candidate set as the test set $\mathcal{D}_{\text{predict}}$ and use M_{base} (Qwen2.5-VL-3B) Bai et al. (2025) as the basic model to predict the coordinates P_{pred}.:

$$\mathcal{D}_{\text{predict}} \subset \mathcal{D}_{\text{candidate}}, \quad |\mathcal{D}_{\text{predict}}| = N$$

$$P_{\text{pred}} = M_{\text{base}}(I, T)$$

A sample is assigned to the failure set $\mathcal{D}_{\text{fail}}$ if the predicted point P_{pred} falls outside the ground-truth bounding box B_{gt}:

$$\mathcal{D}_{\text{fail}} = \{(I, T, B_{\text{gt}}) \in \mathcal{D}_{\text{predict}} \mid P_{\text{pred}} \notin B_{\text{gt}}\}$$

where $P_{\text{pred}} \notin B_{\text{gt}}$ indicates that the predicted point is not in the ground truth bounding box.

Step 2: Mine samples of the same type. For each failed sample $(I, T, B_{\text{gt}}) \in \mathcal{D}_{\text{fail}}$, we obtain the element E_{gt} by crop I with B_{gt}. Then, use E_{gt} to query IconBank and calculate the similarity using $\mathcal{L}_2$ distance. Retain the top five most similar visual concepts C_{top_5} as the similarity set $\mathcal{D}_{\text{similar}}$. The $\mathcal{L}_2$ distance between two vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is calculated as:

$$\mathcal{L}_2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1}$$

Where n is the dimensionality of the vectors.

Step 3: Refine the construction of the training set. We predicted incorrect samples from the original dataset and obtained $\mathcal{D}_{\text{fail}}$. We queried similar visual concepts from IconBank and obtained $\mathcal{D}_{\text{similar}}$. Then, we combined the above two sets to form the difficult sample set $\mathcal{D}_{\text{hard}}$.

$$\mathcal{D}_{\text{hard}} = \mathcal{D}_{\text{fail}} \cup \mathcal{D}_{\text{similar}} \tag{2}$$

However, training solely on difficult samples may compromise data diversity. To prevent the model from suffering catastrophic forgetting due to single data, we add a small number of random samples, $\mathcal{D}_{\text{random}}$ (for example, 1k samples from $\mathcal{D}_{\text{candidate}}$), to balance the data diversity. Finally, we randomly select N samples from $\mathcal{D}_{\text{hard}}$ (for example, $N = 1000$) and combine them with $\mathcal{D}_{\text{random}}$ to form the final training set. As a refined training set:

$$\mathcal{D}_{\text{train}} = \text{Sampled}_N(\mathcal{D}_{\text{hard}}) \cup \mathcal{D}_{\text{random}} \tag{3}$$

where $\text{Sampled}_N(\cdot)$ denotes the operation of randomly selecting N samples from a set.

4 Experiments

To comprehensively evaluate the GUI grounding capability of IconBank, we conduct extensive experiments. This section is structured as follows: Section 4.1 introduces the implementation details and evaluation metrics. Section 4.2 presents the main results and comparative analysis against state-of-the-art methods. Finally, Section 4.3 provides in-depth ablation studies and discussions to validate the contribution of each component with our framework.

4.1 EXPERIMENT SETUP

Implementation Details. We adopted Reinforcement Fine-Tuning(RFT) learning for training, and the training framework is from GUI-G^2 Tang et al. (2025). All models are trained on 8 NVIDIA A6000 GPU. The training parameters are consistent with GUI-G^2, and the key training parameters include the learning rate 1e-6, global batch size 32, sampling 8 responses per instruction, and total work training for 1 epoch.2 We using Qwen2.5-VL-3B-Instruct and Qwen2.5-VL-7B-Instruct as the base model for RFT training.

Evaluation Benchmarks. We designed a systematic experimental protocol. The experiments contain three publicly available GUI Grounding benchmarks, ScreenSpot, ScreenSpot-v2, and ScreenSpot-Pro. ScreenSpot contains 1272 natural language guided GUI Grouding tasks, involving three platforms: mobile application, desktop system, and web page. Screenspot-v2 fixes mislabeled content and clarifies obfuscation instructions on the ScreenSpot benchmark. ScreenSpot-Pro is a benchmark for high-resolution interface design. It contains screenshots from professional tools such as VSCode, AutoCAD, Photoshop, with smaller UI elements and more complex visual scenes. These characteristics make it provide a more realistic and rigorous evaluation criterion for GUI Grounding tasks. When the prediction center falls within the ground truth bounding box, it is determined to be correct.

Compare with State-of-the-Art Methods. We compare IconBank against a comprehensive set of state-of-the-art methods to ensure a rigorous evaluation. These include: (1) data-intensive models trained on large-scale datasets (e.g., UI-TARS-7B, UIPro-7B); (2) efficient supervised fine-tuning (SFT) models that leverage smaller, curated datasets (e.g., ZonUI-3B, ShowUI-2B); and (3) reinforcement learning (RL) based models designed for high sample efficiency (e.g., UI-R1, InfiGUI-G1, GUI-G²). We also include general-purpose vision-language models (e.g., Qwen2.5-VL) and proprietary models (e.g., GPT-4o) as baseline references.

4.2 MAIN RESULTS

Table 1 shows the performance of IconBank on the ScreenSpot and ScreenSpot-v2 benchmarks. IconBank-7B reaches 89.6% on ScreenSpot, which is 1.82% higher than GuirlVG with the same amount of training data. It reaches 91.7% on ScreenSpot-v2, which is nearly 1% higher than GuirlVG Kang et al. (2025). Although this result shows a slight advantage, the ability of IconBank lies in learning from difficult samples with small target elements, and the ScreenSpot benchmark is not enough to reflect the advantage of IconBank.

IconBank performs particularly well on the more challenging ScreenSpot-Pro benchmark. As shown in Table 2, IconBank-3B achieves 51.7% accuracy, which is 14.3% higher than the previous state-of-the-art 3B model InfiGUI-G1 Liu et al. (2025), and significantly better than all models with parameters below 4B. It even achieves performance comparable to that of the state-of-the-art 7B model. IconBank-7B has an accuracy of 55.7%, which is 7.1% higher than the previous SOTA 7B model InfiGUI-G1. ScreenSpot-Pro contains a large number of small target elements, which imposes extremely high requirements on the GUI Grounding ability of the model. IconBank performs well on this benchmark, owing to its accurate mining of challenging and underlearned small target elements.

Further analyzing the performance of ScreenSpot-Pro benchmark on different software scenarios, we find that IconBank is more accurate in locating ICONS in all software scenarios, which proves that our difficult samples can indeed enable the model to learn to locate small target elements. This result has significant implications, as high-resolution interfaces usually contain more complex layouts and smaller interaction elements, which is a difficult point in the GUI Grounding task.

To verify the effect of each component of IconBank, we performed ablation experiments on ScreenSpot-Pro. The high resolution and small target characteristics of this benchmark fit well with our requirements for investigating the IconBank Grounding capability. First, we evaluate the effectiveness of the difficult sample mining strategy. As shown in Table 3, when using 2k randomly sampled training samples, the accuracy of the model on ScreenSpot-Pro is only 49.6%, and when 1k difficult samples are combined with 1k random samples, the performance is improved by 24%. This comparison fully demonstrates that the difficult sample mining strategy can effectively mine valuable training samples for the GUI Grounding task.

Table 1: Performance Comparison on ScreenSpot v1 and v2 Benchmarks under Comparable Data Scales. Bold highlights the best results. "-" indicates results not mentioned in the original paper, unpublished model checkpoints.

Model	Data	ScreenSpot v1 Accuracy (%)						SSv1 Avg.	SSv2 Avg.
		Mobile		Desktop		Web			
		Text	Icon	Text	Icon	Text	Icon		
Data-Intensive Models									
Qwen2-VL-7B	1M	61.3	39.3	52.0	45.0	33.0	21.8	42.9	-
SeeClick-9.6B	1M	78.0	52.0	72.2	30.0	55.7	32.5	53.4	55.1
UGround-7B	773K	82.8	60.3	82.5	63.6	80.4	70.4	73.3	76.3
OS-Atlas-7B	2.3M	93.0	72.9	91.8	62.9	90.9	74.3	82.5	-
Aguvis-7B	1M	95.6	77.7	93.8	67.1	88.3	75.2	84.4	80.5
Qwen2.5-VL-3B	-	-	-	-	-	-	-	55.5	80.9
UIPro-7B	20M	-	-	-	-	-	-	82.5	86.9
Qwen2.5-VL-7B	-	-	-	-	-	-	-	84.7	88.8
UI-TARS-7B	18M	94.5	85.2	95.9	85.7	90.0	83.5	89.5	91.6
Supervised Fine-Tuning Models									
ShowUI-2B	22K	92.3	75.5	76.3	61.1	81.7	63.6	75.1	77.3
ZonUI-3B	24K	-	-	-	-	-	-	84.9	86.4
Reinforcement Learning Models									
UI-R1-3B	136	95.6	84.7	90.2	59.3	85.2	73.3	83.3	85.4
UI-R1-E-3B	3K	**97.1**	83.0	95.4	77.9	**91.7**	85.0	89.2	89.5
GUI-R1-3B	3K	-	-	93.8	64.8	89.6	72.1	-	-
GUI-R1-7B	3K	-	-	91.8	73.6	91.3	75.7	-	-
SE-GUI-7B	3K	-	-	-	-	-	-	88.2	90.3
GuirlVG-7B	2K	-	-	-	-	-	-	88.7	90.9
Ours									
IconBank-7B	**2K**	96.3	**86.4**	**95.9**	**92.2**	82.1	**86.9**	**89.6**	**91.7**

4.3 ABLATION STUDIES

Ablation on Combined Strategy. We further analyze the role of 1k random samples in the training samples. The experiments compare two strategies, using only difficult samples and using combined samples. As shown in Table 3, the accuracy rate of using only difficult samples is 53.1%, and that of using only random samples is 52.1%, which is lower than 55.7% using the combined samples. This shows that combining samples can provide the model with more comprehensive learning signals while ensuring targeted training, avoiding catastrophic forgetting, and improving its generalizability.

Ablation on Training Paradigm. We also analyze the effect of Supervised Fine-Tuning(SFT) training with only 2k samples. ZonUI-3B was used as a training framework with consistent parameters. As shown in Table 4, using only 1k hard samples extracted by IconBank and 1k random samples for training, ScreenSpot-Pro still maintains strong performance. The accuracy of ScreenSpot-Pro is 34.2%, much higher than that of ZonUI-3B (28.7%), and its performance is comparable to that of the 3B model trained by reinforcement learning. Such as UI-R1-E-3B (33.5%), InfiGUI-R1-3B Liu et al. (2025) (35.7%), and SE-GUI-3B (35.9%), GUI-G1-3B: 37.1%). This demonstrates that our method remains effective even within the SFT paradigm. It is worth emphasizing that the conventional SFT depends on large-scale training data is not completely true. Thousands of samples with critical information value are sufficient to train a model with superior performance.

Ablation on Data Scale Saturation. We similarly analyze the redundancy of the training data. We compare the effect of different sizes of training sets (1k, 2k, 3k, 5k samples), all of which are combined samples. Due to constraints on training resources, which limited our ability to conduct extensive RFT training, the following experiments were performed using supervised fine-tuning (SFT) with the ZonUI-3B framework. As shown in Figure 3, when 500 samples are used, the model accuracy is 31.3%, which is significantly lower than the accuracy of 35.0% trained with 2k samples.

Table 2: Performance Comparison on ScreenSpot-Pro Benchmarks. Bold highlights the best results. "-" indicates results not mentioned in the original paper, unpublished model checkpoints.

	CAD		Dev		Creative		Scientific		Office		OS		Avg.		**Overall**
Model	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	**Avg.**
Proprietary Models															
GPT-4o	2.0	0.0	1.3	0.0	1.0	0.0	2.1	0.0	1.1	0.0	0.0	0.0	1.3	0.0	0.8
General Vision-Language Models															
Qwen2.5-VL-3B	9.1	7.3	22.1	1.4	26.8	2.1	38.2	7.3	33.9	15.1	10.3	1.1	23.6	3.8	16.1
Qwen2.5-VL-7B	16.8	1.6	46.8	4.1	35.9	7.7	49.3	7.3	52.5	20.8	37.4	6.7	38.9	7.1	26.8
Supervised Fine-Tuning Models															
ShowUI-2B	2.5	0.0	16.9	1.4	9.1	0.0	13.2	7.3	15.3	7.5	10.3	2.2	10.8	2.6	7.7
UGround-7B	14.2	1.6	26.6	2.1	27.3	2.8	31.9	2.7	31.6	11.3	17.8	0.0	25.0	2.8	16.5
OS-Atlas-7B	12.2	4.7	33.1	1.4	28.8	2.8	37.5	7.3	33.9	5.7	27.1	4.5	28.1	4.0	18.9
UI-TARS-2B	17.8	4.7	47.4	4.1	42.9	6.3	56.9	17.3	50.3	17.0	21.5	5.6	39.6	8.4	27.7
ZonUI-3B	31.9	15.6	24.6	6.2	40.9	7.6	54.8	18.1	57.0	26.4	19.6	7.8	39.2	11.7	28.7
UGround-V1-7B	15.8	1.2	51.9	2.8	47.5	9.7	57.6	14.5	60.5	13.2	38.3	7.9	45.2	8.1	31.1
UI-TARS-7B	20.8	9.4	58.4	12.4	50.0	9.1	63.9	31.8	63.3	20.8	30.8	16.9	47.8	16.2	35.7
JEDI-3B	27.4	9.4	61.0	13.8	53.5	8.4	54.2	18.2	64.4	32.1	38.3	9.0	49.8	13.7	36.1
UI-TARS-72B	18.8	12.5	62.9	17.2	57.1	15.4	64.6	20.9	63.3	26.4	42.1	15.7	50.9	17.6	38.1
JEDI-7B	38.0	14.1	42.9	11.0	50.0	11.9	72.9	25.5	75.1	47.2	33.6	16.9	52.6	18.2	39.5
GUI-Actor-7B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	44.6
Reinforcement Learning Models															
UI-R1-3B	11.2	6.3	22.7	4.1	27.3	3.5	42.4	11.8	32.2	11.3	13.1	4.5	24.9	6.4	17.8
UI-R1-E-3B	37.1	12.5	46.1	6.9	41.9	4.2	56.9	21.8	65.0	26.4	32.7	10.1	-	-	33.5
GUI-R1-3B	26.4	7.8	33.8	4.8	40.9	5.6	61.8	17.3	53.6	17.0	28.1	5.6	-	-	-
GUI-R1-7B	23.9	6.3	49.4	4.8	38.9	8.4	55.6	11.8	58.7	26.4	42.1	16.9	-	-	-
InfiGUI-R1-3B	33.0	14.1	51.3	12.4	44.9	7.0	58.3	20.0	65.5	28.3	43.9	12.4	49.1	14.1	35.7
SE-GUI-3B	38.1	12.5	55.8	7.6	47.0	4.9	61.8	16.4	59.9	24.5	40.2	12.4	50.4	11.8	35.9
GUI-G1-3B	39.6	9.4	50.7	10.3	36.6	11.9	61.8	30.0	67.2	32.1	23.5	10.6	49.5	16.8	37.1
ReGUIDE-3B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	44.3
ReGUIDE-7B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	44.4
InfiGUI-G1-3B	50.8	25.0	64.9	20.0	51.5	16.8	68.8	32.7	70.6	32.1	49.5	15.7	-	-	45.2
SE-GUI-7B	51.3	**42.2**	68.2	19.3	57.6	9.1	75.0	28.2	78.5	43.4	49.5	25.8	63.5	21.0	47.3
GUI-G^2-7B	55.8	12.5	68.8	17.2	57.1	15.4	77.1	24.5	74.0	32.7	57.9	21.3	64.7	19.6	47.5
V2P-7B	58.3	12.5	67.5	24.8	62.6	16.0	73.6	33.6	75.7	43.4	**56.1**	32.6	65.8	25.8	50.5
InfiGUI-G1-7B	57.4	23.4	**74.7**	24.1	64.6	15.4	**80.6**	31.8	75.7	39.6	57.0	29.2	-	-	51.9
Ours															
IconBank-3B	55.8	32.8	63.6	**31.0**	65.2	31.5	75.0	35.5	72.9	37.7	46.7	25.8	63.9	32.0	51.7
IconBank-7B	**66.5**	29.7	66.2	27.6	**65.7**	**33.6**	76.4	**38.2**	**82.5**	**56.6**	48.6	**33.7**	**68.7**	**34.6**	**55.7**

However, when 3k and 5k samples are used, the accuracy rates are 34.4% and 35.7%, respectively, which does not show significant advantages. This result shows that if the source of training data is unchanged, it is difficult to blindly increase the amount of data to improve performance. This finding has important implications for reducing the training cost of GUI models.

5 Conclusion

In this paper, we propose IconBank, a novel framework for mining hard samples via a decoupled visual concept library to achieve data-efficient GUI grounding. Through extensive experiments on multiple benchmarks, we validate that data quality surpasses quantity in GUI grounding tasks. On the challenging ScreenSpot-Pro benchmark, IconBank-3B achieves 51.7% accuracy, outperforming most 7B models. We show that combining diffcult samples with a small number of random samples helps maintain model generalization and avoid catastrophic forgetting, leading to more robust

Table 3: Performance Comparison of Different 2K Training Data Strategies on the ScreenSpot-Pro Benchmark.

	CAD		Dev		Creative		Scientific		Office		OS		Avg.		**Overall**
Model	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	**Avg.**
7B Models															
IconBank-7B	**66.5**	**29.7**	**66.2**	**27.6**	65.7	**33.6**	76.4	**38.2**	82.5	**56.6**	48.6	**33.7**	68.7	**34.6**	**55.7**
w/o difficult samples	54.8	26.6	64.3	25.5	63.2	30.1	75.7	35.5	82.5	49.1	53.3	**37.1**	66.0	32.3	53.1
w/o random samples	56.8	21.8	64.9	26.9	**69.6**	28.7	**84.7**	35.4	**84.7**	47.1	**56.0**	34.8	**69.7**	31.2	55.1

Table 4: Performance Comparison of SFT with IconBank-Selected 2K Samples on ScreenSpot-Pro Benchmark.

	CAD		Dev		Creative		Scientific		Office		OS		Avg.		**Overall**
Model	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	**Avg.**
3B Models															
ZonUI-3B	**31.9**	15.6	24.6	6.2	40.9	7.6	54.8	18.1	**57.0**	**26.4**	19.6	7.8	39.2	11.7	28.7
w/ IconBank-data	23.4	**17.2**	**43.5**	**24.1**	**41.3**	**24.5**	**60.4**	**24.5**	48.6	22.6	**39.3**	**27.0**	**42.7**	**23.3**	**35.0**

performance. In the future, we plan to expand IconBank to support the decoupled visual concept library of associated semantic information. By searching through natural language instructions, the desired samples can be obtained more accurately. We believe that a data-centric approach to building efficient and lightweight GUI agents is a direction worthy of research.

Ethical Statements and Reproducibility

This work uses only publicly available datasets for academic purposes. We acknowledge that this technology could be misused for unauthorized automation, which is an important ethical consideration. To ensure reproducibility, the code and model checkpoints will be released after the double-blind review process. All experimental hyperparameters are detailed in Appendix A.

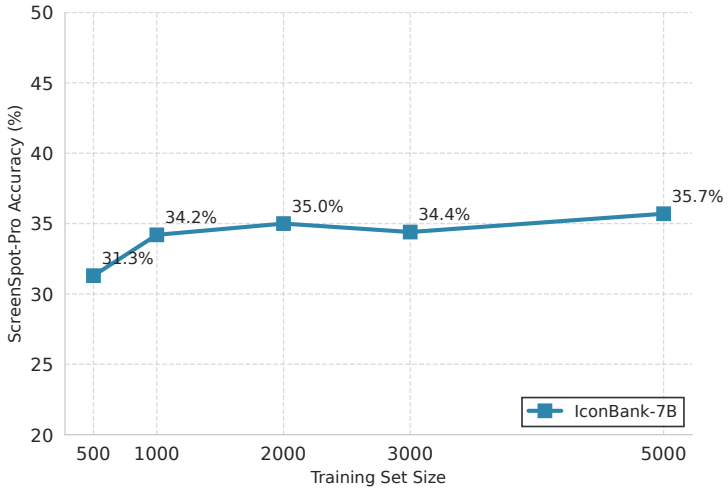

Figure 3: Accuracy on ScreenSpot-Pro with varying training data sizes using SFT.

REFERENCES

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Eric Wang. Agent s: An open agentic framework that uses computers like a human. *ArXiv preprint*, abs/2410.08164, 2024.

Saaket Agashe, Kyle Wong, Vincent Tu, Jiachen Yang, Ang Li, and Xin Eric Wang. Agent s2: A compositional generalist-specialist framework for computer use agents. *ArXiv preprint*, abs/2504.00906, 2025.

Hao Bai, Yifei Zhou, Jiayi Pan, Mert Cemri, Alane Suhr, Sergey Levine, and Aviral Kumar. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *ArXiv preprint*, abs/2502.13923, 2025.

Gongwei Chen, Xurui Zhou, Rui Shao, Yibo Lyu, Kaiwen Zhou, Shuai Wang, Wentao Li, Yinchuan Li, Zhongang Qi, and Liqiang Nie. Less is more: Empowering gui agent with context-aware simplification. *ArXiv preprint*, abs/2507.03730, 2025a.

Jikai Chen, Long Chen, Dong Wang, Leilei Gan, Chenyi Zhuang, and Jinjie Gu. V2p: From background suppression to center peaking for robust gui grounding task. *ArXiv preprint*, abs/2508.13634, 2025b.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. *ArXiv preprint*, abs/2401.10935, 2024.

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*, 2017.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *ArXiv preprint*, abs/2401.08281, 2024.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for gui agents. *ArXiv preprint*, abs/2410.05243, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016.

Yanheng He, Jiahe Jin, and Pengfei Liu. Efficient agent training for computer use. *ArXiv preprint*, abs/2505.13909, 2025.

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*, 2024.

ZongHan Hsieh, Tzer-Jen Wei, and ShengJing Yang. Zonui-3b: A lightweight vision-language model for cross-resolution gui grounding. *arXiv e-prints*, 2025.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *ArXiv preprint*, abs/2410.21276, 2024.

Wenjia Jiang, Yangyang Zhuang, Chenxi Song, Xu Yang, Joey Tianyi Zhou, and Chi Zhang. Appagentx: Evolving gui agents as proficient smartphone users. *ArXiv preprint*, abs/2503.02268, 2025.

Weitai Kang, Bin Lei, Gaowen Liu, Caiwen Ding, and Yan Yan. Guirlvg: Incentivize gui visual grounding via empirical exploration on reinforcement learning. *ArXiv preprint*, abs/2508.04389, 2025.

Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem AlShikh, and Ruslan Salakhutdinov. Omniact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In *European Conference on Computer Vision*. Springer, 2024.

Hyunseok Lee, Jeonghoon Kim, Beomjun Kim, Jihoon Tack, Chansong Jo, Jaehong Lee, Cheonbok Park, Sookyo In, Jinwoo Shin, and Kang Min Yoo. Reguide: Data efficient gui grounding via spatial reasoning and search. *ArXiv preprint*, abs/2505.15259, 2025.

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*, 2023.

Hongxin Li, Jingran Su, Jingfan Chen, Zheng Ju, Yuntao Chen, Qing Li, and Zhaoxiang Zhang. Uipro: Unleashing superior interaction capability for gui agents, 2025a.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use. *ArXiv preprint*, abs/2504.07981, 2025b.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

Yuhang Liu, Zeyu Liu, Shuanghe Zhu, Pengxiang Li, Congkai Xie, Jiasheng Wang, Xueyu Hu, Xiaotian Han, Jianbo Yuan, Xinyao Wang, et al. Infigui-g1: Advancing gui grounding with adaptive exploration policy optimization. *ArXiv preprint*, abs/2508.05731, 2025.

Fanbin Lu, Zhisheng Zhong, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Arpo: End-to-end policy optimization for gui agents with experience replay. *ArXiv preprint*, abs/2505.16282, 2025a.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. Omniparser for pure vision based gui agent. *ArXiv preprint*, abs/2408.00203, 2024.

Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning. *ArXiv preprint*, abs/2503.21620, 2025b.

Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. Gui-r1: A generalist r1-style vision-language action model for gui agents. *ArXiv preprint*, abs/2504.10458, 2025.

Yichen Pan, Dehan Kong, Sida Zhou, Cheng Cui, Yifei Leng, Bing Jiang, Hangyu Liu, Yanyi Shang, Shuyan Zhou, Tongshuang Wu, et al. Webcanvas: Benchmarking web agents in online environments. volume abs/2406.12373, 2024.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. Ui-tars: Pioneering automated gui interaction with native agents. *ArXiv preprint*, abs/2501.12326, 2025.

Fei Tang, Zhangxuan Gu, Zhengxi Lu, Xuyang Liu, Shuheng Shen, Changhua Meng, Wen Wang, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. Gui-g^2: Gaussian reward modeling for gui grounding, 2025.

Ke Wang, Tianyu Xia, Zhangxuan Gu, Yi Zhao, Shuheng Shen, Changhua Meng, Weiqiang Wang, and Ke Xu. E-ant: A large-scale dataset for efficient automatic gui navigation. *ArXiv preprint*, abs/2406.14250, 2024a.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *ArXiv preprint*, abs/2409.12191, 2024b.

Qianhui Wu, Kanzhi Cheng, Rui Yang, Chaoyun Zhang, Jianwei Yang, Huiqiang Jiang, Jian Mu, Baolin Peng, Bo Qiao, Reuben Tan, et al. Gui-actor: Coordinate-free visual grounding for gui agents. *ArXiv preprint*, abs/2506.03143, 2025.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: A foundation action model for generalist gui agents. *ArXiv preprint*, abs/2410.23218, 2024.

Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, et al. Scaling computer-use grounding via user interface decomposition and synthesis. *ArXiv preprint*, abs/2505.13227, 2025.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. Aguvis: Unified pure vision agents for autonomous gui interaction. *ArXiv preprint*, abs/2412.04454, 2024.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. Aria-ui: Visual grounding for gui instructions. *ArXiv preprint*, abs/2412.16256, 2024.

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*. Springer, 2024.

Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin Hou, Jinwei Chen, Peng-Tao Jiang, et al. Enhancing visual grounding for gui agents via self-evolutionary reinforcement learning. *ArXiv preprint*, abs/2505.12370, 2025.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 2025a.

Miaosen Zhang, Ziqiang Xu, Jialiang Zhu, Qi Dai, Kai Qiu, Yifan Yang, Chong Luo, Tianyi Chen, Justin Wagle, Tim Franklin, et al. Phi-ground tech report: Advancing perception in gui grounding. *ArXiv preprint*, abs/2507.23779, 2025b.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

Yuqi Zhou, Sunhao Dai, Shuai Wang, Kaiwen Zhou, Qinglin Jia, and Jun Xu. Gui-g1: Understanding r1-zero-like training for visual grounding in gui agents, 2025. *ArXiv preprint*, abs/2505.15810, 2025.

A APPENDIX

Table 5: Hyperparameter settings for IconBank training with different frameworks.

Hyperparameter	**GUI-G² RFT**	**ZonUI SFT**
Base Model	Qwen2.5-VL-3B/7B-Instruct	Qwen2.5-VL-3B-Instruct
Training Framework	GUI-G² RFT	ZonUI SFT
Deepspeed Config	zero3	zero2
Max Prompt Length	12048	-
Number of Generations per Instruction	8	-
Per Device Train Batch Size	1	1
Gradient Accumulation Steps	4	2
Global Batch Size	32	8
Learning Rate	1×10^{-6}	1×10^{-4}
Number of Training Epochs	1	12
Optimizer	AdamW	-
Precision	bfloat16	bf16
Gradient Checkpointing	true	true
Attention Implementation	flash_attention_2	sdpa
Beta (RL parameter)	0.04	-
Max Pixels	12845056	-
Data Seed	42	-
Steps per Epoch	-	100
Warmup Steps	-	122
Model Max Length	-	4096
LoRA Rank	-	8
LoRA Alpha	-	16
Min Visual Tokens	-	256
Max Visual Tokens	-	1280