FERRET-UI ONE: MASTERING UNIVERSAL USER INTERFACE UNDERSTANDING ACROSS PLATFORMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Building a generalist model for user interface (UI) understanding is challenging due to various foundational issues, such as platform diversity, resolution variation, and data limitation. In this paper, we introduce **Ferret-UI One**, a multimodal large language model (MLLM) designed for universal UI understanding across a wide range of platforms, including iPhone, Android, iPad, Webpage, and AppleTV. Building on the foundation of Ferret-UI, Ferret-UI One introduces three key innovations: support for multiple platform types, high-resolution perception through adaptive scaling, and advanced task training data generation powered by GPT-4o with set-of-mark visual prompting. These advancements enable Ferret-UI One to perform complex, user-centered interactions, making it highly versatile and adaptable for the expanding diversity of platform ecosystems. Extensive empirical experiments on referring, grounding, user-centric advanced tasks (comprising 9 subtasks $\times$ 5 platforms), GUIDE next-action prediction dataset, and GUI-World multi-platform benchmark demonstrate that Ferret-UI One significantly outperforms Ferret-UI, and also shows strong cross-platform transfer capabilities.

1 INTRODUCTION

User interfaces (UIs) are central to human-computer interaction, shaping how users interact with digital systems. The complexity of UIs has evolved with the proliferation of platforms such as smartphones, tablets, web platforms, and smart TVs. Despite this increasing diversity, many current approaches to UI understanding and interaction (Hong et al., 2023; Wang et al., 2024b; Kapoor et al., 2024), particularly those in multi-platform ecosystems, face limitations.

One prominent effort in this space is Ferret-UI (You et al., 2024), which has advanced the field of referring and grounding UIs. However, though taking an any-resolution approach (Liu et al., 2024a), Ferret-UI is constrained by a fixed grounding resolution (*i.e.*, 336×672 and 672×336), and focuses on single-type platforms (*i.e.*, mobile devices including iPhone and Android), limiting its applicability in the context of today's highly diverse platform landscape. For example, as illustrated in Figure 1, one notable difference among these four exemplified platforms is resolution, the native resolution of iPhone differs from that of iPad, Web UI, and also AppleTV, directly applying Ferret-UI across these platforms presents significant challenges. Another major challenge is the lack of platform-specific, high-quality data, given different platforms. Though Ferret-UI's approach to training data generation can be extended to these platforms, it primarily relies on text-based GPT-4 prompting, where bounding boxes are represented in a purely textual format. This absence of visual input and spatial relationships between UI elements diminishes the quality of training data, which in turn limits the performance and effectiveness of the resulting model.

To address these limitations, we introduce **Ferret-UI One**, a multimodal large language model (MLLM) designed to understand diverse UI screens and respond to user intent through single-step interactions across multiple platforms. Building on the foundation of Ferret-UI (You et al., 2024), Ferret-UI One significantly enhances UI perception and user interaction capabilities via three key advancements: (i) multi-platform support, (ii) dynamic high-resolution image encoding, and (iii) high-quality multimodal training data generation.

First, Ferret-UI One extends its compatibility beyond mobile platforms (iPhone and Android), incorporating additional platforms like tablets, webpages, and smart TVs. Figure 1 illustrates visual ex-

Figure 1: Real examples of a single Ferret-UI One model interacting with four different platforms (iPhone, iPad, Webpage, and AppleTV) for UI understanding. Refer to Appendix F for more examples including multi-step interactions.

amples of Ferret-UI One interacting with users across four typical screen types. This multi-platform support enables broader applicability and allows the system to seamlessly scale across a variety of user environments.

Second, Ferret-UI One supports high-resolution image encoding via the any-resolution method (Liu et al., 2024a; Zhang et al., 2024c). However, going beyond that, we introduce an enhanced *adaptive gridding* approach to maintain perception capabilities at the original resolution of the UI screenshot, ensuring more accurate recognition of visual elements. By leveraging human-collected bounding box annotations, we enhance referring and grounding precision, which allows for a more detailed understanding of UI components and their relationships.

Third, Ferret-UI One leverages high-quality training data for both elementary and advanced tasks. For elementary tasks, we convert simple referring and grounding data into conversations, allowing the model to develop a fundamental understanding of diverse UI screens. For advanced tasks, which focus on user-centered, free-form conversations, we replace the text-based GPT-4 prompting (where bounding boxes are described only in text) with GPT-4o using set-of-mark visual prompting (Yang et al., 2023) for training data generation. This approach enhances spatial understanding of UI elements, resulting in higher-quality training data. Additionally, unlike previous methods that use straightforward instructions such as "click on [bbox location]", Ferret-UI One performs single-step user-centered interactions. For example, when given a command like "please confirm submission", the system understands and executes the intended action, rather than simply following mechanical click instructions. Overall, our contributions are summarized as follows.

- We present Ferret-UI One, a multimodal LLM that sets itself apart from previous efforts by supporting a broader range of platforms, including iPhone, Android, iPad, Webpage, and AppleTV. We upgrade Ferret-UI across multiple fronts, including better instruction-tuning data for model training, high-resolution image encoding for enhanced performance, and new referring and grounding benchmarks tailored for different platforms.
- We demonstrate that Ferret-UI One advances the UI referring and grounding performance on different platforms. On three categories of tasks (referring, grounding, and user-centric advanced tasks, comprising 9 subtasks $\times$ 5 platforms), Ferret-UI One outperforms Ferret-UI, and also shows competitive performance compared to GPT-4o. Besides, Ferret-UI One also exhibits strong transfer capabilities across platforms. Finally, Ferret-UI One achieved strong performance on recent benchmarks like GUIDE (Chawla et al., 2024) and GUI-World (Chen et al., 2024a).

2 Related Work

UI agents have garnered significant attention in recent research, particularly in multimodal models that seek to automate complex UI tasks across diverse platforms. Many works have advanced the field by tackling specific challenges related to single-platform and multi-platform UI understanding, interaction, and automation.

Single-Platform UI Agents. Single-platform UI agents focus on automating tasks on a specific device ecosystem, such as Android, iOS, desktop environments, or webpages. On the mobile side, DigiRL (Bai et al., 2024), AppAgent V2 (Li et al., 2024c), AutoDroid (Wen et al., 2024), and MobileFlow (Nong et al., 2024) proposed Android agents targeting on human-like interactions. For web-based agents, systems like WebShop (Yao et al., 2022), WebArena (Zhou et al., 2023), LASER (Ma et al., 2023), WebAgent (Gur et al., 2023), AutoWebGLM (Lai et al., 2024), WebVoyager (He et al., 2024) and Agent-E (Abuelsaad et al., 2024) explored agents that navigate and perform tasks within web environments, while MindSearch agent Chen et al. (2024b) focused on AI engine for web search. AssistGUI (Gao et al., 2023), OS-Copilot (Wu et al., 2024), SYNAPSE (Zheng et al., 2023) and UFO (Zhang et al., 2024a) explored the more complex computer OS interaction. These efforts have significantly improved task-specific automation, although their single-platform nature limits cross-platform flexibility.

Multi-Platform UI Agents. Multi-platform UI agents have emerged to address the growing complexity of device ecosystems, supporting a variety of devices and platforms, including mobile, web, and desktop environments. Zheng et al. (2024a); Cheng et al. (2024) demonstrated GPT-4V as a generalist agent when grounded. Recent works like CogAgent (Hong et al., 2023) support UI navigation on both PC webpages and Android devices. Mobile-Agent V2 (Wang et al., 2024a) features Harmony OS and Android OS for non-English and English scenarios. Ferret-UI (You et al., 2024)

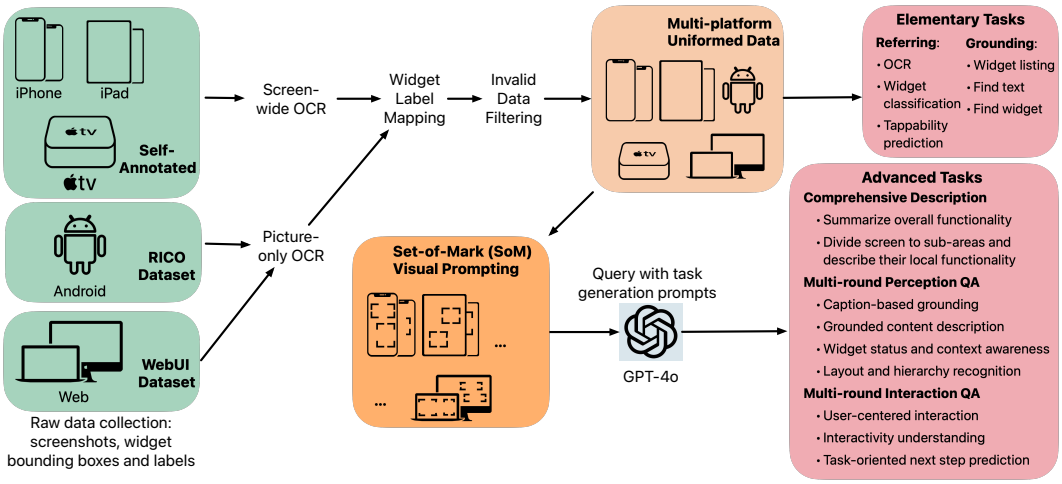


Figure 2: Illustration of the Core-set data generation pipeline.

focuses on mobile UI understanding for both Android and iPhone screenshots using multimodal LLMs (McKinzie et al., 2024; Zhang et al., 2024b), with a focus on referring and grounding capabilities. These agents aim to perform more complex, user-intent-based interactions across multiple device types, paving the way for truly generalist multimodal agents.

UI-Agent Benchmarks. The evaluation of UI agents requires specialized benchmarks to test various aspects of UI interaction, including task execution, navigation, and interaction understanding. Rico (Deka et al., 2017) remains a foundational dataset for mobile app interaction, while Mobile-Env (Zhang et al., 2023), AndroidEnv (Toyama et al., 2021), AndroidWorld (Rawles et al., 2024a), Android in-the-Wild (Rawles et al., 2024b), AndroidControl (Li et al., 2024b), and AMEX (Chai et al., 2024) provide benchmarks for mobile device control. Windows Agent Arena (Bonatti et al., 2024) introduces a benchmark focusing on PC Windows environment. OSWorld (Xie et al., 2024) takes a broader approach by providing benchmarks for agents in real computer environments, including Ubuntu, MacOS, and Windows. Web-based interaction benchmarks include WebSRC (Chen et al., 2021), Mind2Web (Deng et al., 2024), and WebCanvas (Pan et al., 2024) focusing on structural reading comprehension and task execution in web environments. OmniACT (Kapoor et al., 2024) supports both desktop and web interfaces. More recently, MobileAgentBench (Wang et al., 2024c) and VisualWebBench (Liu et al., 2024b) have introduced taxonomies designed to evaluate the performance of multimodal agents across both mobile and web interfaces. VisualAgentBench (Liu et al., 2024c) expands this with a focus on multimodal LLMs as visual foundation agents. GUI Odyssey (Lu et al., 2024) provides benchmarks for cross-app navigation. GUI-World (Chen et al., 2024a) pioneer in covering multi-platform benchmarking for GUI agents, while CRAB (Xu et al., 2024) further tests cross-environment tasks for GUI agents. These benchmarks contribute to a growing need for unified, multi-platform evaluations that can assess UI agents' adaptability, precision, and efficiency.

Compared to the aforementioned works, Ferret-UI One is the first to target universal UI understanding across diverse platforms, including smartphones, tablets, web interfaces, and smart TVs. It focuses on foundational capabilities like fine-grained referring, grounding, and reasoning, aiming to create a generalist agent for versatile UI navigation.

3 Ferret-UI One

In this section, we first describe how we curate our training datasets from the raw data annotations (Section 3.1) and then describe the model architecture (Section 3.2).

3.1 Dataset Construction

We construct our own dataset in order to train a strong multi-platform UI understanding model. A flow diagram of our complete dataset generation pipeline is shown in Figure 2.

Table 1: A summary of datasets across various platforms used to train Ferret-UI One. The screenshot resolution statistics are shown in Appendix G.

Training Data	Platform	#Images (k)	Task Types
Core-set	iPhone	112	Elementary Tasks (Referring, Grounding), Advanced Tasks (Comprehensive Description, Multi-Round Perception QA and Interaction QA)
	Android	61	
	iPad	19	
	Web	321	
	AppleTV	16	
GUIDE	Web	51	Next Action Prediction
GroundUI-18k	Web	18	Simple Interaction
Spotlight	Android	66	Screen2Word, Widget Caption, Taperception

Raw Annotation Collection. The primary dataset used for training Ferret-UI One is a combination of data from various platform types, including iPhone, Android, iPad, Webpage, and AppleTV. The data collection process varies depending on the platform type:

- **iPhone, iPad, and AppleTV:** We use human-collected iPhone, iPad, and AppleTV data under diverse usage scenarios and human-annotated widget bounding box coordinates and labels. To save annotation costs, we do not collect text annotations; instead, text bounding boxes are replaced by screen-wide OCR-detected text and bounding boxes using an OCR confidence threshold of 0.5.
- **Webpage:** The web data is derived from the WebUI dataset (Wu et al., 2023). Bounding boxes of all types of UI widgets and text annotations for non-picture widgets are directly parsed from the source HTML view hierarchy tree, providing high-quality annotations. For picture widgets, we further use OCR to detect texts contained in the pictures.
- **Android:** The Android data for screenshots, bounding boxes, and text annotations is transformed from the RICO dataset (Deka et al., 2017). Similar to the WebUI dataset, we also perform picture-only OCR to complete the missing text annotations in picture widgets.

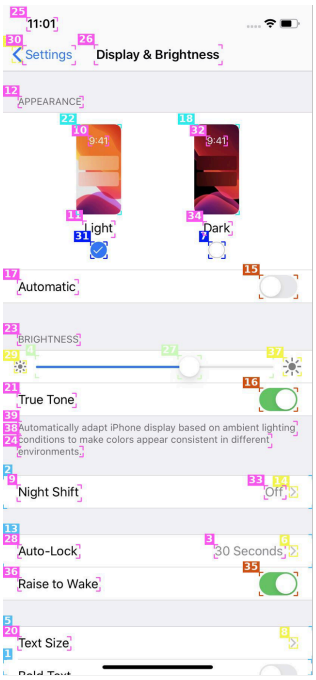
For all the collected data, we perform data filtering including: (i) filter out or narrow down out-of-bound bounding boxes and remove empty screenshots with no remaining bounding boxes after box filtering; (ii) since we do not intend to add multilingual support for the Ferret-UI One model, screenshots with more than 5% non-ASCII characters in the text annotations are also removed.

Despite the different types of label spaces from various sources, we filter out bounding boxes associated with less relevant labels (*e.g.*, UI types) and uniformly map the remaining labels to a common label space containing 13 classes: 'Checkbox', 'Button', 'Container', 'Dialog', 'Icon', 'PageControl', 'Picture', 'SegmentedControl', 'Slider', 'TabBar', 'Text', 'TextField', and 'Toggle', and obtain a multi-platform uniformed dataset with raw UI widget annotations. We provide the original label statistics and their converted labels in Appendix H.

We name the above-collected screenshot dataset the Core-set, which we will use to construct the elementary and advanced task data. Besides, we also employ third-party training datasets to enrich our data source and avoid overfitting our predefined tasks. A complete statistics of the training dataset of Ferret-UI One is summarized in Table 1, indicating that the dataset distribution is very unbalanced across different platforms. In particular, the number of iPad and AppleTV screenshots is significantly smaller than that of other platforms. To address this, we (i) assign different loss weights to different platforms during training, and (ii) generate all three types of advanced tasks for each example of iPad and AppleTV platforms and generate only 1 type of advanced task for each example of other platforms.

Compared to the training dataset of Ferret-UI, which relies on model-detected bounding boxes, Ferret-UI One's training dataset predominantly utilizes either human-collected annotations or bounding boxes directly parsed from the source HTML, resulting in a significant improvement in annotation quality, evidenced later in our quantitative evaluation in Section 4.2.

Task Data Generation. For task data generation, we follow the paradigm of Ferret-UI data construction, which includes both elementary tasks and advanced tasks.



Task type: Multi-round Interaction QA
Conversations:
User:
<image>
How can I switch to dark mode on this screen?
Model:
You can switch to dark mode by tapping the "Dark" option [<WIDGET_34>] and then selecting the circle [<WIDGET_7>] underneath it.
User:
What does the "True Tone" option do?
Model:
The "True Tone" option [<WIDGET_21>] automatically adapts the iPhone display based on ambient lighting conditions to make colors appear consistent [<WIDGET_39> <WIDGET_38>].
User:
How can I adjust the brightness?
Model:
You can adjust the brightness using the slider [<WIDGET_4>] under the "BRIGHTNESS" section [<WIDGET_23>].
User:
Is there a way to toggle the automatic brightness?
Model:
Yes, you can toggle automatic brightness by switching the toggle [<WIDGET_15>] next to "Automatic" [<WIDGET_17>].
User:
Where do I turn off the "Raise to Wake" feature?
Model:
The "Raise to Wake" feature is currently on, you can tap on the toggle [<WIDGET_35>] to switch it off.

Figure 3: Example of set-of-mark visual prompting (Yang et al., 2023) (**left**) and one of its generated advanced task training examples (**right**).

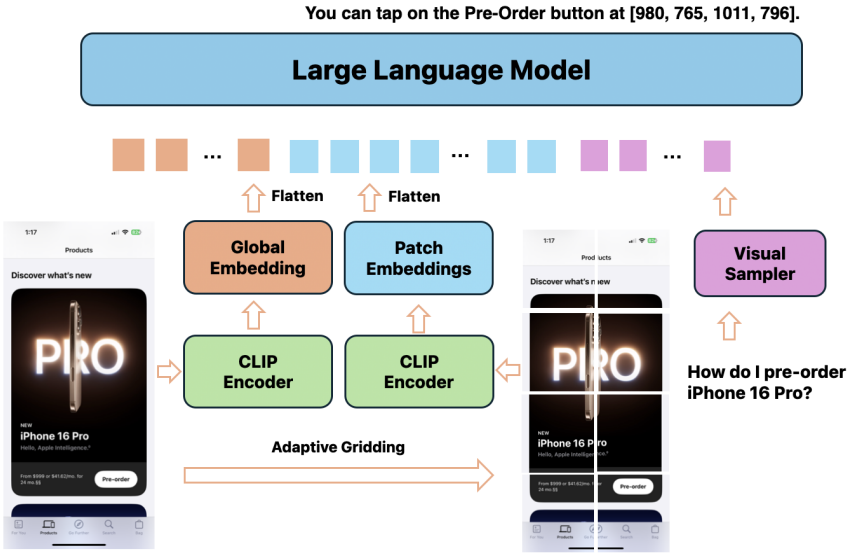


Figure 4: Overview of the Ferret-UI One model architecture, which allows for seamless UI understanding and user-centered single-step interactions with high-resolution support.

Elementary tasks (Figure 2) consist of 3 referring tasks and 3 grounding tasks. Specifically, referring tasks include (i) *OCR*: recognizing the text given a text bounding box, (ii) *widget classification*: predict the UI type of the elements, and (iii) *tapperbility*: predict whether the selected widget is tappable for interaction; meanwhile, grounding tasks include (i) *widget listing*: list all the widgets in the screen, (ii) *find text*: find the location of a given text, and (iii) *find widget*: find the widget given the widget description.

For advanced tasks, we prompt GPT-4o with bounding box annotations of a given screenshot and require GPT-4o to generate QA tasks related to the UI widgets in the screenshot. Unlike Ferret-UI, which mainly focuses on spatial descriptions due to the limitation of using textual prompts without

Algorithm 1: Adaptive N-gridding

Require: Original resolution: w $\times$ h, grid size: 336×336, size limit N
Ensure: Optimal gridding size N_w and N_h $(N_w, N_h \in \mathbb{N}^+)$

$N_{w_{\text{best}}}, N_{h_{\text{best}}} \leftarrow 0$, $\Delta_{\text{best}} \leftarrow \infty$, $N_{w_0} \leftarrow \frac{\text{w}}{336}$, $N_{h_0} \leftarrow \frac{\text{h}}{336}$
for $N_w = 1$ to N **do** ▷ Traverse all grid configuarations
  for $N_h = 1$ to $N - N_w$ **do**
    $\Delta_{\text{aspect}} \leftarrow \sqrt{\left|\frac{N_w}{N_h} - \frac{N_{w_0}}{N_{h_0}}\right| \left|\frac{N_h}{N_w} - \frac{N_{h_0}}{N_{w_0}}\right|}$ ▷ Get aspect ratio change
    $\Delta_{\text{pixel}} \leftarrow \frac{\left|N_w \times N_h - N_{w_0} \times N_{h_0}\right|}{N_{w_0} \times N_{h_0}}$ ▷ Get relative pixel change for resizing
    if $\Delta_{\text{best}} > \Delta_{\text{aspect}} \times \Delta_{\text{pixel}}$ **then**
      $(N_{w_{\text{best}}}, N_{h_{\text{best}}}) \leftarrow (N_w, N_h)$
      $\Delta_{\text{best}} \leftarrow \Delta_{\text{aspect}} \times \Delta_{\text{pixel}}$
    end if
  end for
end for
return $(N_{w_{\text{best}}}, N_{h_{\text{best}}})$

image information (*i.e.*, screenshots) for bounding box annotations, Ferret-UI One leverages GPT-4o to generate advanced task data that covers a variety of aspects of UI understanding. This is possible because GPT-4o demonstrates an improved ability to comprehend the spatial relationships between UI widgets when provided with the screenshot as input. Specifically, we prompt GPT-4o to generate 3 types of advanced tasks (shown in Figure 2) including: (i) *comprehensive description*: describe global and local functionalities of the screen, (ii) *multi-round perception QA*: multi-round question answering regarding the UI perception capabilities, and (iii) *multi-round interaction QA*: multi-round question answering regarding the single-step and user-centric UI interactions based on the current screen status. More detailed requirements and prompts for GPT-4o when generating advanced tasks are provided in Appendix B.

We empirically find it hard for GPT-4o to find the location of referred UI widgets with the original screenshot as input (*i.e.*, bad grounding capability). To address this, we use Set-of-Mark (SoM) visual prompting (Yang et al., 2023) when generating multi-round perception and interaction QA training samples. One example of SoM prompting and its generated data sample is shown in Figure 3, where each UI widget is marked with a corner-style bounding box and a unique number tag for easy identification. Furthermore, the same class of UI widgets have the same color for visual prompting to help GPT-4o better differentiate the bounding boxes of spatially close or nested widgets. Please refer to Figure 6 for visual prompts on other platforms, and Appendix D for additional examples of generated data for advanced tasks.

As aforementioned in Table 1, in addition to the tasks generated on the Core-set, we augment training data with additional third-party datasets, including GroundUI-18k (Zheng et al., 2024b), GUIDE (Chawla et al., 2024) and Spotlight (Li & Li, 2023).

3.2 Model Architecture

As shown in Figure 4, the model architecture of Ferret-UI One directly builds upon Ferret-UI (You et al., 2024), which uses the Any-Resolution (AnyRes) method (Liu et al., 2024a) to enhance referring and grounding, enabling the encoder to capture diverse image resolutions.

Specifically, the CLIP image encoder first extracts both global (derived from the low-resolution overview image) and local features (corresponding to high-resolution sub-images) from the UI screenshot. Then, these image features are flattened and sent into the LLM. The Visual Sampler identifies and selects the relevant UI regions based on user instructions. The model then outputs grounded descriptions for perception or interaction with the UI elements.

Adaptive Gridding. Local image features are extracted by calculating the optimal grid size using our proposed *adaptive N-gridding* mechanism, then resizing and encoding the visual features of each grid. This is a key model innovation compared to Ferret-UI. As shown in Algorithm 1, the optimal gridding size N_w and N_h is determined when the gridding and resizing based on (N_w, N_h)

Table 2: Results on our constructed benchmarks for elementary and advanced tasks, as well as the GUIDE benchmark (Chawla et al., 2024). Results on elementary and advanced tasks are averaged over all platforms, including iPhone, Android, iPad, Webpage, and AppleTV. Each platform includes 6 elementary tasks and 3 advanced tasks. SeeClick model Cheng et al. (2024) trained on their original data is compared. (†) In tasks that require referring, GPT-4o is equipped with set-of-mark (SoM) prompting by adding a red rectangular box to screenshots for the referred widget. Note that SoM visual prompting is not used for Ferret-UI and Ferret-UI One.

Model	Backbone	Elementary		Advanced		GUIDE Bench	
		Refer	Ground	GPT-4o Score	Multi-IoU	BertScore	IoU
Ferret-UI	Vicuna-13B	64.15	57.22	45.81	18.75	41.15	26.91
Ferret-UI One	Gemma-2B	75.20	78.13	80.25	40.51	83.71	51.13
	Llama3-8B	80.28	**82.79**	**89.73**	41.15	**91.37**	**55.78**
	Vicuna-13B	**81.34**	81.31	86.25	**41.71**	88.81	54.71
SeeClick Cheng et al. (2024)	QWen-VL-9.6B	51.58	62.82	67.49	21.56	54.70	39.51
GPT-4o	-	56.47	12.14	77.73	7.06	75.31	9.64
GPT-4o + SoM-Prompt†	-	**87.91**	-	84.33	7.36	-	-

lead to minimal *aspect ratio change* times *the relative pixel number change*, under the constraint $N_w + N_h \leq N$, where N is the *size limit*. With the size limit N, the total number of grids is upper bounded by $\lfloor \frac{N^2}{4} \rfloor$. Compared to the AnyRes module that has unbounded cost, the key differences of adaptive N-gridding is it automatically finds the optimal gridding configuration, *i.e.*, least resolution distortion (aspect ratio change and pixel number change) within a predefined inference cost limit that $N_w \times N_h \leq \lfloor \frac{N^2}{4} \rfloor$, which is both information-preserving and efficient for local encoding. With adaptive gridding, Ferret-UI One understands UI screens and provides user-centered interactions with an optimal configuration at any resolution given the inference cost limit specified as N.

4 Experiments

4.1 Experiment Setup

Training Data. The training datasets are summarized in Table 1, which can be divided into two categories: (i) datasets constructed by our own, which include elementary task data and advanced task data across all platforms as introduced in Section 3.1, and (ii) public datasets including GroundUI-18k (Zheng et al., 2024b), a simple user-centered interaction dataset on webpage screenshots, GUIDE (Chawla et al., 2024), a next-action prediction dataset on webpage screenshots, and Spotlight (Li & Li, 2023), an Android UI understanding and interaction dataset.

Model. Following Ferret-UI (You et al., 2024), Ferret-UI One uses a CLIP ViT-L/14 model as the image encoder; for the LLM backbone, besides Vicuna-13B (Chiang et al., 2023) as used in the original Ferret-UI, we also tried 2 additional LLMs at mobile scales, including Gemma-2B (Team et al., 2024) and Llama3-8B (Dubey et al., 2024). As to dynamic high-resolution image encoding, we set the size limit N to 8, so that the maximal grid number is 16 for adaptive gridding.

Evaluation. At a high level, model evaluation falls into two broad categories: (i) benchmarks we constructed, and (ii) public benchmarks. For our benchmarks, we created a total of 45, including 6 elementary tasks and 3 advanced tasks per platform type, across 5 platforms. For elementary tasks, we follow the evaluation metrics outlined by You et al. (2023). For advanced tasks, we use GPT-4o to score generated answers for a given screenshot and user query, visually prompting GPT-4o with a red rectangular bounding box. Advanced tasks are tested using GPT-4o evaluation score and multi-IoU. The multi-IoU is calculated by first matching predicted bounding boxes with ground truth bounding boxes and then calculating the average IoU of each pair of bounding boxes (IoU $= 0$ if no match). Furthermore, we conduct a next-action prediction test given previous action history on the GUIDE benchmark (Chawla et al., 2024), and evaluate the semantic similarity w.r.t. the reference answer and grounding Intersection-over-Union (IoU). Additionally, we evaluate our model on the recently released GUI-World benchmark (Chen et al., 2024a) on the supported platforms following the original GPT-4 evaluation protocol as in Chen et al. (2024a), which does not include evaluating the grounding capability for interaction-related UI tasks.

Table 3: Zero-shot performance of Ferret-UI One on the GUI-World benchmark (Chen et al., 2024a).

Model	GPT-4 Score			
	iOS	Android	Webpage	Average
MiniGPT4Video (Ataallah et al., 2024)	1.501	1.342	1.521	1.455
VideoChat2 (Li et al., 2024a)	2.169	2.119	2.221	2.170
Chat-Univi (Jin et al., 2024)	2.337	2.390	2.349	2.359
GUI-Vid (Chen et al., 2024a)	2.773	2.572	2.957	2.767
QWen-VL-MAX (Bai et al., 2023)	2.779	2.309	2.656	2.580
SeeClick Cheng et al. (2024)	2.614	2.650	2.848	2.704
Ferret-UI (You et al., 2024)	2.713	2.791	2.411	2.638
Ferret-UI One	**2.881**	**2.954**	**3.013**	**2.948**
Gemini-Pro 1.5 (Reid et al., 2024)	3.213	3.220	3.452	3.295
GPT-4o	3.558	3.561	3.740	3.619

Table 4: Zero-shot cross-platform transfer results of Ferret-UI One. For simplicity, we train and test the model only using data corresponding to the elementary tasks.

Training	Test - Referring					Test - Grounding				
	iPhone	iPad	AppleTV	Web	Android	iPhone	iPad	AppleTV	Web	Android
iPhone	**86.3**	68.1	31.2	45.3	71.2	**84.1**	65.2	43.1	51.7	63.1
iPad	67.5	**80.2**	40.7	51.5	63.3	64.5	**82.1**	32.1	38.5	53.8
AppleTV	29.1	45.1	**79.3**	54.2	36.4	33.7	41.2	**81.6**	52.1	29.7
Web	59.2	57.4	41.2	**85.5**	41.7	54.0	51.2	46.5	**87.5**	45.9
Android	72.5	60.7	35.7	51.2	**86.2**	66.7	48.9	29.7	44.1	**83.9**

4.2 Experiment Results

Main results. Our main results are summarized in Table 2, which shows the comparative performance of different models on our constructed elementary and advanced tasks, as well as the GUIDE benchmark (Chawla et al., 2024). Note, that for each data entry corresponding to the elementary and advanced tasks, it is an average across all platforms. The detailed results on each platform are provided in Table 6 of Appendix A. Below, we highlight a few observations.

- Ferret-UI One, powered by Llama-3-8B, delivers the best results across most metrics. It achieves the highest GPT-4o score on advanced tasks, with a notable 89.73, surpassing Ferret-UI by 43.92 points and GPT-4o by 12.0 points. Notably, Ferret-UI One with Llama-3-8B also achieves the highest IoU score on the GUIDE benchmark with 55.78, indicating superior grounding capability.

- Ferret-UI One, equipped with Vicuna-13B, also performs well, *e.g.*, achieving a strong Multi-IoU score of 41.71 on advanced tasks. Despite being six times smaller, Ferret-UI One with Gemma-2B delivers competitive performance across the board.

- In contrast, GPT-4o struggles with fine-grained UI understanding, as shown by its low referring (56.47) and grounding (12.14) scores in the elementary tasks. Its Multi-IoU and IoU scores in advanced tasks and the GUIDE benchmark are also low.

Overall, the results demonstrate the versatility of Ferret-UI One in handling UI understanding tasks across different platforms.

Results on GUI-World. To further demonstrate the zero-shot performance of using Ferret-UI One out of the box, we further test our model on the recently released GUI-World benchmark (Chen et al., 2024a). Results are summarized in Table 3. Clearly, Ferret-UI One does not overfit the training data and can generalize well to the test data in the wild. Notably, Ferret-UI One outperforms GUI-Vid (Chen et al., 2024a), a model developed in the GUI-World paper, on supported platforms including iOS, Android, and Webpages.

Table 5: Ablation results of the architecture and dataset improvements of Ferret-UI One w.r.t. Ferret-UI-anyRes (You et al., 2024), *i.e.*, the high-resolution version of Ferret-UI equipped with the any-resolution module. **iPhone v1** refers to the dataset on the iPhone platform originally used by Ferret-UI, while **iPhone v2** is the data used by Ferret-UI One. Models are evaluated on the advanced tasks.

Training Data	**Model**	iPhone v1		iPhone v2	
		GPT-4o Score	Multi-IoU	GPT-4o Score	Multi-IoU
iPhone v1	Ferret-UI-anyRes	91.3	36.89	68.3	27.13
	Ferret-UI One	93.7 (+2.4)	37.12 (+0.23)	70.2 (+1.9)	28.21 (+1.08)
iPhone v2	Ferret-UI-anyRes	86.2	35.89	85.97	39.81
	Ferret-UI One	88.1 (+1.9)	36.43 (+0.54)	89.7 (+3.73)	41.73 (+1.92)

4.3 Ablation Study

Cross-Platform Transferability. In Table 4, we evaluate the zero-shot platform (domain) transfer capability of the Ferret-UI One model by training it on elementary tasks from one platform and testing it on other platforms. These results provide insights into how well the model generalizes across different platforms. We observe similar performance patterns across tasks. Specifically,

- iPhone transfers well to iPad and Android platforms on both tasks, achieving at least 68.1 and 71.2 scores on referring tasks and 65.2 and 63.1 scores on grounding tasks due to its diverse screenshot contents (because of the large screenshot number) and similar resolutions and aspect ratios with other two platforms. iPad and Android also transfer fairly well to iPhone at around 65 scores.
- AppleTV and Web do not transfer very well to the mobile domains, including iPhone, Android, and iPad, achieving the highest referring score of 59.2 and grounding score of 54.0, possibly because they are mostly landscape screenshots, which is in contrast to the mostly portrait screenshots in mobile platforms. Models trained on other domains all achieve poor performance on the AppleTV test domain, with the highest score of around 40 on both kinds of tasks, which is reasonable due to a large domain gap in terms of AppleTV's content distribution compared to other domains.

The results suggest that (i) iPhone, iPad and Android have similar content distribution, which helps them generalize to each other; (ii) models trained on more diverse contents (*e.g.*, around 100k iPhone data) can generalize better to other platforms; (iii) platforms with similar resolution and aspect ratios may transfer better to each other; and (iv) good transferability among some of these platforms contributes to the cross-platform performance of Ferret-UI One.

Ablation on Architecture and Dataset Improvements. Table 5 presents a comparison between the Ferret-UI and Ferret-UI One model trained on different versions of the iPhone dataset. Models are evaluated on the test set corresponding to advanced tasks. The results indicate that both the adaptive N-gridding and improved dataset (iPhone v2) contribute to performance gains.

Specifically, when evaluated on the iPhone v1 test set, Ferret-UI One shows a slight improvement over Ferret-UI, with the GPT-4o score increased from 91.3 to 93.7 and the Multi-IoU score increased from 36.89 to 37.12. However, the improvements are more pronounced on the iPhone v2 dataset. Here, Ferret-UI One achieves a GPT-4o score of 89.7, outperforming Ferret-UI's 85.97, along with a substantial Multi-IoU score boost from 39.81 to 41.73. These results suggest that while both architecture and dataset enhancements contribute to overall performance, the new dataset plays a more significant role in driving improvements, particularly on more challenging tasks.

5 Conclusions

In this paper, we presented Ferret-UI One, a multimodal large language model designed to improve UI understanding and interaction across diverse platforms. With multi-platform support, high-resolution image encoding with adaptive gridding, and improved data generation, Ferret-UI One outperforms Ferret-UI on all tested benchmarks. The model demonstrates strong zero-shot transferability across platforms, establishing Ferret-UI One as a solid foundation for universal UI understanding. Future work will focus on incorporating additional platform types and building a generalist agent for universal UI navigation.

REFERENCES

Tamer Abuelsaad, Deepak Akkil, Prasenjit Dey, Ashish Jagmohan, Aditya Vempaty, and Ravi Kokku. Agent-e: From autonomous web navigation to foundational design principles in agentic systems. *arXiv preprint arXiv:2407.13032*, 2024.

Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Deyao Zhu, Jian Ding, and Mohamed Elhoseiny. Minigpt4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens. *arXiv preprint arXiv:2404.03413*, 2024.

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *arXiv preprint arXiv:2406.11896*, 2024.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Justin Wagle, Kazuhito Koishida, Arthur Bucker, Lawrence Jang, et al. Windows agent arena: Evaluating multi-modal os agents at scale. *arXiv preprint arXiv:2409.08264*, 2024.

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.

Rajat Chawla, Adarsh Jha, Muskaan Kumar, Mukunda NS, and Ishaan Bhola. Guide: Graphical user interface data for execution. *arXiv preprint arXiv:2404.16048*, 2024.

Dongping Chen, Yue Huang, Siyuan Wu, Jingyu Tang, Liuyi Chen, Yilin Bai, Zhigang He, Chenlong Wang, Huichi Zhou, Yiqiang Li, et al. Gui-world: A dataset for gui-oriented multimodal llm-based agents. *arXiv preprint arXiv:2406.10819*, 2024a.

Xingyu Chen, Zihan Zhao, Lu Chen, Danyang Zhang, Jiabao Ji, Ao Luo, Yuxuan Xiong, and Kai Yu. Websrc: A dataset for web-based structural reading comprehension. *arXiv preprint arXiv:2101.09465*, 2021.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Jiangning Liu, Wenwei Zhang, Kai Chen, and Feng Zhao. Mindsearch: Mimicking human minds elicits deep ai searcher. *arXiv preprint arXiv:2407.20183*, 2024b.

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 arXiv:2401.10935*, 2024.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

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*, pp. 845–854, 2017.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *NeurIPS*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Difei Gao, Lei Ji, Zechen Bai, Mingyu Ouyang, Peiran Li, Dongxing Mao, Qinchen Wu, Weichen Zhang, Peiyi Wang, Xiangwu Guo, et al. Assistgui: Task-oriented desktop graphical user interface automation. *arXiv preprint arXiv:2312.13108*, 2023.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*, 2023.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023.

Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *CVPR*, 2024.

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. *arXiv preprint arXiv:2402.17553*, 2024.

Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, et al. Autowebglm: Bootstrap and reinforce a large language model-based web navigating agent. *arXiv preprint arXiv:2404.03648*, 2024.

Gang Li and Yang Li. Spotlight: Mobile ui understanding using vision-language models with a focus, 2023.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024a.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on computer control agents. *arXiv preprint arXiv:2406.03679*, 2024b.

Yanda Li, Chi Zhang, Wanqi Yang, Bin Fu, Pei Cheng, Xin Chen, Ling Chen, and Yunchao Wei. Appagent v2: Advanced agent for flexible mobile interactions. *arXiv preprint arXiv:2408.11824*, 2024c.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024a. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. Visualwebbench: How far have multimodal llms evolved in web page understanding and grounding? *arXiv preprint arXiv:2404.05955*, 2024b.

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Yifan Xu, Xixuan Song, Shudan Zhang, Hanyu Lai, Xinyi Liu, Hanlin Zhao, et al. Visualagentbench: Towards large multimodal models as visual foundation agents. *arXiv preprint arXiv:2408.06327*, 2024c.

Quanfeng Lu, Wenqi Shao, Zitao Liu, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, Yu Qiao, and Ping Luo. Gui odyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. *arXiv preprint arXiv:2406.08451*, 2024.

Kaixin Ma, Hongming Zhang, Hongwei Wang, Xiaoman Pan, Wenhao Yu, and Dong Yu. Laser: Llm agent with state-space exploration for web navigation. *arXiv preprint arXiv:2309.08172*, 2023.

Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

Songqin Nong, Jiali Zhu, Rui Wu, Jiongchao Jin, Shuo Shan, Xiutian Huang, and Wenhao Xu. Mobileflow: A multimodal llm for mobile gui agent. *arXiv preprint arXiv:2407.04346*, 2024.

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. *arXiv preprint arXiv:2406.12373*, 2024.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*, 2024a.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Androidinthewild: A large-scale dataset for android device control. *NeurIPS*, 2024b.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Daniel Toyama, Philippe Hamel, Anita Gergely, Gheorghe Comanici, Amelia Glaese, Zafarali Ahmed, Tyler Jackson, Shibl Mourad, and Doina Precup. Androidenv: A reinforcement learning platform for android. *arXiv preprint arXiv:2105.13231*, 2021.

Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *arXiv preprint arXiv:2406.01014*, 2024a.

Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception, 2024b.

Luyuan Wang, Yongyu Deng, Yiwei Zha, Guodong Mao, Qinmin Wang, Tianchen Min, Wei Chen, and Shoufa Chen. Mobileagentbench: An efficient and user-friendly benchmark for mobile llm agents. *arXiv preprint arXiv:2406.08184*, 2024c.

Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. Autodroid: Llm-powered task automation in android. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pp. 543–557, 2024.

Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey Bigham. Webui: A dataset for enhancing visual ui understanding with web semantics. *ACM Conference on Human Factors in Computing Systems (CHI)*, 2023.

Zhiyong Wu, Chengcheng Han, Zichen Ding, Zhenmin Weng, Zhoumianze Liu, Shunyu Yao, Tao Yu, and Lingpeng Kong. Os-copilot: Towards generalist computer agents with self-improvement. *arXiv preprint arXiv:2402.07456*, 2024.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972*, 2024.

Tianqi Xu, Linyao Chen, Dai-Jie Wu, Yanjun Chen, Zecheng Zhang, Xiang Yao, Zhiqiang Xie, Yongchao Chen, Shilong Liu, Bochen Qian, et al. Crab: Cross-environment agent benchmark for multimodal language model agents. *arXiv preprint arXiv:2407.01511*, 2024.

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.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *NeurIPS*, 2022.

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, 2023.

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. *arXiv preprint arXiv:2404.05719*, 2024.

Chaoyun Zhang, Liqun Li, Shilin He, Xu Zhang, Bo Qiao, Si Qin, Minghua Ma, Yu Kang, Qingwei Lin, Saravan Rajmohan, et al. Ufo: A ui-focused agent for windows os interaction. *arXiv preprint arXiv:2402.07939*, 2024a.

Danyang Zhang, Hongshen Xu, Zihan Zhao, Lu Chen, Ruisheng Cao, and Kai Yu. Mobile-env: an evaluation platform and benchmark for llm-gui interaction. *arXiv preprint arXiv:2305.08144*, 2023.

Haotian Zhang, Mingfei Gao, Zhe Gan, Philipp Dufter, Nina Wenzel, Forrest Huang, Dhruti Shah, Xianzhi Du, Bowen Zhang, Yanghao Li, et al. Mm1. 5: Methods, analysis & insights from multimodal llm fine-tuning. *arXiv preprint arXiv:2409.20566*, 2024b.

Haotian Zhang, Haoxuan You, Philipp Dufter, Bowen Zhang, Chen Chen, Hong-You Chen, Tsu-Jui Fu, William Yang Wang, Shih-Fu Chang, Zhe Gan, et al. Ferret-v2: An improved baseline for referring and grounding with large language models. *arXiv preprint arXiv:2404.07973*, 2024c.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024a.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In *ICLR*, 2023.

Longtao Zheng, Zhiyuan Huang, Zhenghai Xue, Xinrun Wang, Bo An, and Shuicheng Yan. Agentstudio: A toolkit for building general virtual agents. *arXiv preprint arXiv:2403.17918*, 2024b.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

A DETAILED RESULTS ON ELEMENTARY AND ADVANCED TASKS

Table 6 shows the performance on 9 subtasks $\times$ 5 platforms of the Ferret-UI One model. We empirically find that OCR, widget listing, and comprehensive description are relatively difficult tasks compared to other tasks within the same task type.

Table 6: The performance breakdown of Ferret-UI One with Llama-3-8B backbone. Note that we only report GPT-4o evaluation scores and omit the multi-IoU scores for advanced tasks for simplicity.

Task type	Task	Test Domain				
		iPhone	iPad	AppleTV	Web	Android
Refer	OCR	75.3	69.3	74.3	80.5	74.5
	Widget Classify	79.1	78.7	82.5	83.6	82.0
	Tapperbility	89.2	86.0	-	84.3	85.6
Ground	Widget Listing	76.7	74.9	71.6	79.7	76.8
	Find Text	85.2	84.0	82.2	90.2	82.2
	Find Widget	88.6	86.8	87.7	87.8	86.4
Advanced Tasks	Comprehensive	89.0	86.5	84.1	83.7	85.6
	Perception	94.1	87.5	86.6	93.2	92.1
	Interaction	93.5	92.7	86.1	96.4	94.7

B DETAILED GPT-4O PROMPTS FOR GENERATING ADVANCED TASK DATA

In this section, we elaborate on how we prompt GPT-4o to generate training data for advanced tasks. In particular, we have the following requirements when generating each type of advanced task.

1. **Comprehensive Description**: Provide a one-sentence description of the overall functionality of the UI page shown in the screenshot. Then, describe the screenshot in detail by dividing it into several areas/groups and explaining the functionality of each area/group.
2. **Multi-Round Perception QA**:
   (a) **Basic Perceptual Understanding**: Interpret the content of referred widgets, ground the positions of the widgets based on the widget descriptions, and describe the functionality and status (enabled, disabled, selected, hovered) of each widget.
   (b) **Contextual Awareness**: Understand the context in which UI widgets are presented, including the relationships between different UI components and how they contribute to the overall user experience.
   (c) **Layout and Hierarchy Recognition**: Recognize how widgets are grouped and nested within the UI layout.
3. **Multi-Round Interaction QA**:
   (a) Generate instructions for user-centered interaction. For example, "Please help me confirm submission" instead of "please click on [Box0] button".
   (b) Identify and interact with buttons, links, icons, scrollbars, and toggles, and recognize widgets that cannot be interacted with, such as static text and background images.
   (c) Identify and interact with input fields, dropdowns, checkboxes, and radio buttons, as well as their current states (*e.g.*, checked, unchecked, filled, empty) in the context of specific interaction tasks.
   (d) Predict the current step for potential navigation paths based on currently visible widgets like tabs, menus, and breadcrumbs controls.

Table 7: Statictics of different multi-platform GUI understanding datasets.

Dataset	# Sample	Platforms	Tasks
OmniAct	9,802	Desktop, Web	Code Generation
OS-World	369	Desktop, Web	General Control
AITW	715,142	Android(Apps+Web)	Navigation
GUI-world	12,379	iPhone, Android, Web, XR, Desktop	GUI understanding, instruction following
Ferret-UI	123,702	iPhone, Android	Elementary: Referring, Grounding Advanced: Function/detailed description, perception and interaction QA
Ferret-UI One	529,411	iPhone, Android, iPad, AppleTV, Web	Elementary: Referring, Grounding Advanced: Comprehensive description, perception and interaction QA

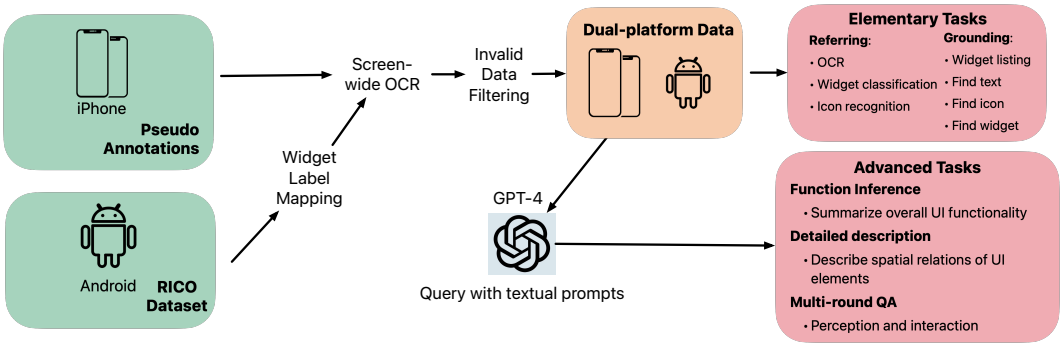


Figure 5: Illustration of the data generation pipeline of Ferret-UI (You et al., 2024).

C DATASET COMPARISON

In this section, we compare the dataset of Ferret-UI One with previous multi-platform datasets as shown in appendix C. In particular, we outline the data generation pipeline of Ferret-UI (You et al., 2024) in fig. 5. Through a comparison of fig. 2 and fig. 5, we can easily identify the differences of data generation.

Importantly, we highlight following key differences between Ferret-UI One and Ferret-UI:

- **Multi-platform support**: The training data of Ferret-UI One consists of 5 platforms compared to 2 platforms of Ferret-UI.
- **Raw annotation qualities**: The majority part of bounding boxes, labels, and on-screen text annotations of Ferret-UI One are either extracted from source data or annotated by humans, while these annotations of Ferret-UI are all generated from model detection.
- **Bounding box prompting**: When generating advanced tasks, Ferret-UI One uses GPT-4o+SoM visual prompting for bounding boxes, while Ferret-UI uses purely textual prompting containing coordinates, bbox labels, and text annotations.
- **Advanced task quality**: Due to the constraint of textual prompting, the Ferret-UI pipeline cannot perceive the visual content of UI elements, thus limiting the quality and content diversity of generated advanced tasks.

D ADVANCED TASK EXAMPLES

Comprehensive Description

OVERALL: This UI page is for adjusting settings related to downloads, language, news, and notifications.
DETAIL:

- **Top Navigation Bar:** Contains menu options such as Home, Account, Settings (currently selected), and Downloads, allowing users to navigate between different sections.
- **Games Section:** Features a checkbox labeled "Download over WiFi only," which is currently checked, indicating that downloads are restricted to WiFi connections.
- **Language Section:** Offers a dropdown menu to select the language, currently set to English, with a pink arrow indicating more options.
- **Title Screen Section:** Includes a checkbox for "Show Sago News," which is checked, meaning news updates will be displayed on the title screen.
- **Notifications Section:** Contains a checkbox for "Display notifications," which is unchecked, allowing the user to enable or disable notifications.
- **Close Button:** Located in the top right corner, represented by an 'X', used to close the settings page.

Multi-Round Perception QA

Q: *What does the "SERVEWARE" text represent?*
A: The text "SERVEWARE" [718, 72, 948, 108] is likely a category label indicating that the items displayed are related to serving ware.

Q: *Can you identify any product names and their corresponding prices?*
A: Yes, the product "ARV BRÖLLOP" [35, 847, 233, 907] is priced at $12.99 [10, 6, 112, 32] and "IKEA 365+" [609, 787, 809, 823] is priced at $2.99 [1174, 1336, 1627, 2119].

Q: *What is the function of the shopping cart icon?*
A: The shopping cart icon [630, 2157, 690, 2217] is typically used to view the items in your cart or proceed to checkout.

Q: *How is the product rating displayed?*
A: The product rating is displayed with star symbols and a numerical rating. For example, "ARV BRÖLLOP" has a rating of 4.9 next to the stars [[38, 956, 100, 986], [248, 1004, 294, 1030]].

Q: *What does the "Filter" button do?*
A: The "Filter" button [1564, 170, 1630, 198] is likely used to apply specific criteria to narrow down the product search results.

Q: *Are there any navigation elements on the screen?*
A: Yes, there seems to be a navigation bar at the bottom with icons for different sections, like a house icon for home [611, 268, 1057, 714] and a pencil icon [1320, 2157, 1380, 2217] which might be for editing or creating a list.

Multi-Round Interaction QA

Q: *How can I refresh the content shown?*
A: You can tap on the refresh icon [272, 736, 424, 882] to reload the content.

Q: *What does the "General" tag refer to?*
A: The "General" tag [1074, 900, 1443, 926] categorizes the content related to general events, like the Caltech Beaver Name Reveal.

Q: *How can I view more recent activities?*
A: You can tap the "Recent" section label [90, 598, 202, 630] to view more recent activities.

Q: *What is the logo displayed on this page?*
A: The logo at the top of the page is the SCIAC Network logo [198, 56, 444, 128] representing the network conducting the event.

Q: *How do I check the institution hosting the event?*
A: The event is hosted by the California Institute of Technology, as shown in the text [11, 499, 483, 542] below the event details.

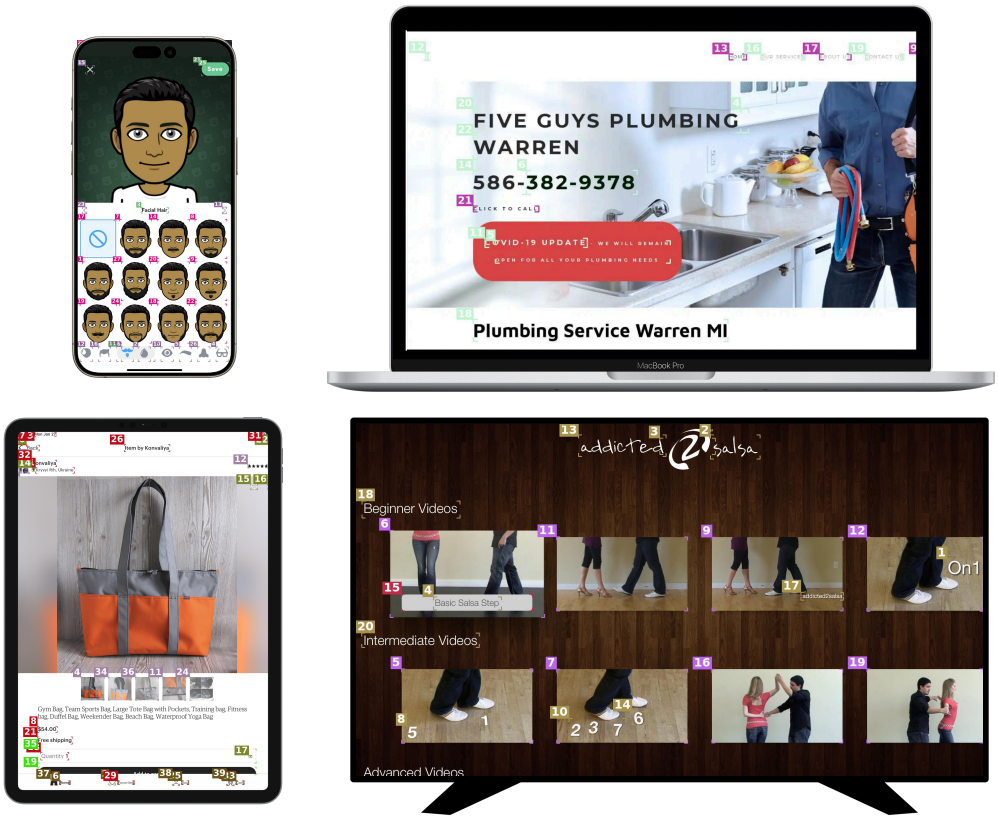


Figure 6: Examples of visual prompting using GPT-4o to generate task data for Multi-Round Perception QA and Multi-Round Interaction QA. Each UI widget is annotated with a corner-style bounding box, where only the corners of the widget are highlighted by small lines, leaving the rest of the box open. This minimalistic bounding style is accompanied by a unique number tag placed near one of the corners, making it easy to identify and reference specific UI widgets for further interaction or perception analysis.

Table 8: We test Ferret-UI One on Screenspot benchmark Cheng et al. (2024) to evaluate the grounding performance. We compare Ferret-UI One with CogAgent Hong et al. (2023) and SeeClick Cheng et al. (2024) using original results from Cheng et al. (2024). **I/W** refers to Icon/Widget data type. Note that the data distribution of Screenspot is unseen during Ferret-UI One training. The results indicate that Ferret-UI One achieve superior grounding accuracy on Mobile and Web screenshots, while underperforms other two models on the unseen Desktop screenshots, which means Ferret-UI One transfers well to unseen data in supported platforms.

Model	**Model Size**	***Mobile***		***Desktop***		***Web***		**Avg**
		Text	I/W	Text	I/W	Text	I/W	
CogAgent	8B	67.0%	24.0%	**74.2%**	20.0%	70.4%	28.6%	47.4%
SeeClick	9.6B	78.0%	52.0%	72.2%	**30.0%**	55.7%	32.5%	53.4%
Ferret-UI One	8B	**80.3%**	**55.4%**	52.1%	21.7%	**81.2%**	**33.5%**	**54.0%**

E GROUNDING PERFORMANCE TEST ON SCREENSPOT DATASET

We test Ferret-UI One on the Screenspot Cheng et al. (2024) benchmark to evaluate its grounding performance on unseen data. The results are shown in table 8. On Screenspot benchmark, the Ferret-UI One achieves 54.0% average accuracy, outperforming CogAgent and SeeClick, and in particular, achieving good performance on our supported Mobile and Web platforms while achieving fair performance on the unseen desktop platform.

F ADDITIONAL INFERENCE EXAMPLES

In fig. 7 to fig. 10, we show more qualitative inference examples of Ferret-UI One on different platforms. Additionally, in Figure 11, we show an example of Ferret-UI One performing multi-step interactions on real-time webpages following GUIDE-style QAs.

G RESOLUTION STATISTICS

In this section, we present the resolution statistics for images collected from various platforms, categorized by device types and resolutions, as summarized in Table 9.

H LABEL STATISTICS AND MAPPING RESULTS OF ORIGINAL DATA ACROSS PLATFORMS

We demonstrate the label statistics of original data across platforms and their mapping results into a uniform label space for better joint training as shown in Figures 12, 13, 14 and 15. The mapping results are obtained via GPT-4 suggestions and additional human reviews. Note that the "Other" label after mapping indicates the widget information will be deprecated.

I FAILURE CASES OF FERRET-UI ONE

In fig. 16 to fig. 20, we showcase several typical and interesting failure cases, in particular on AppleTV and iPad platforms where the raw data scale is relatively small.

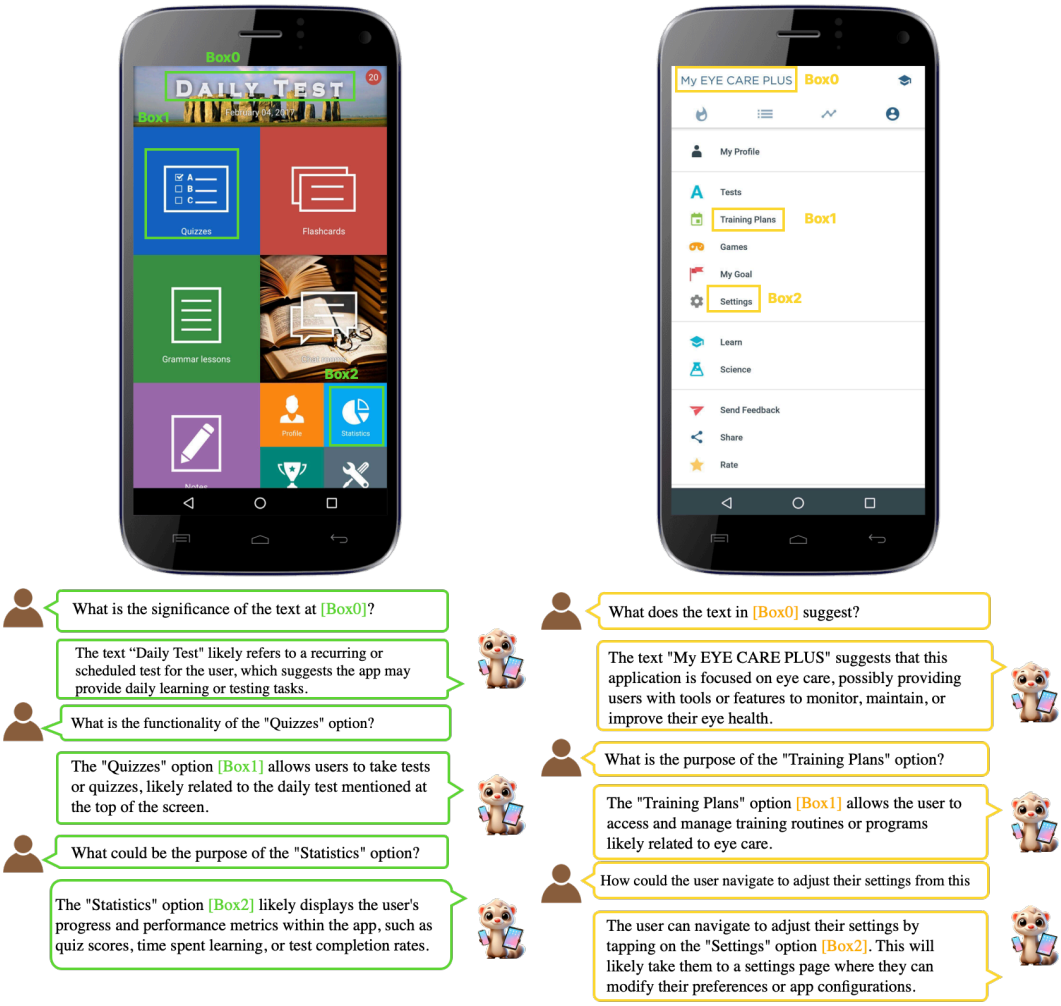


Figure 7: Real examples of the Ferret-UI One model interacting with Android.

Figure 8: Real examples of the Ferret-UI One model interacting with iPhone.

Table 9: Resolution statistics by device type.

Device Type	Resolution	Number of Images
iPhone	828×1792	83,250
	1125×2436	6,055
	1792×828	4,686
	2436×1125	104
iPad	2224×1668	4,829
	1668×2224	14,312
	1242×2208	19
AppleTV	1920×1080	16,152
WebUI	1280×720	53,500
	1366×768	53,500
	1536×864	53,500
	1920×1080	53,500
	2048×2732	53,500
	1170×2532	53,500
Android	540×960	14,092
	1080×1920	52,102
	1920×1080	55
	960×540	12

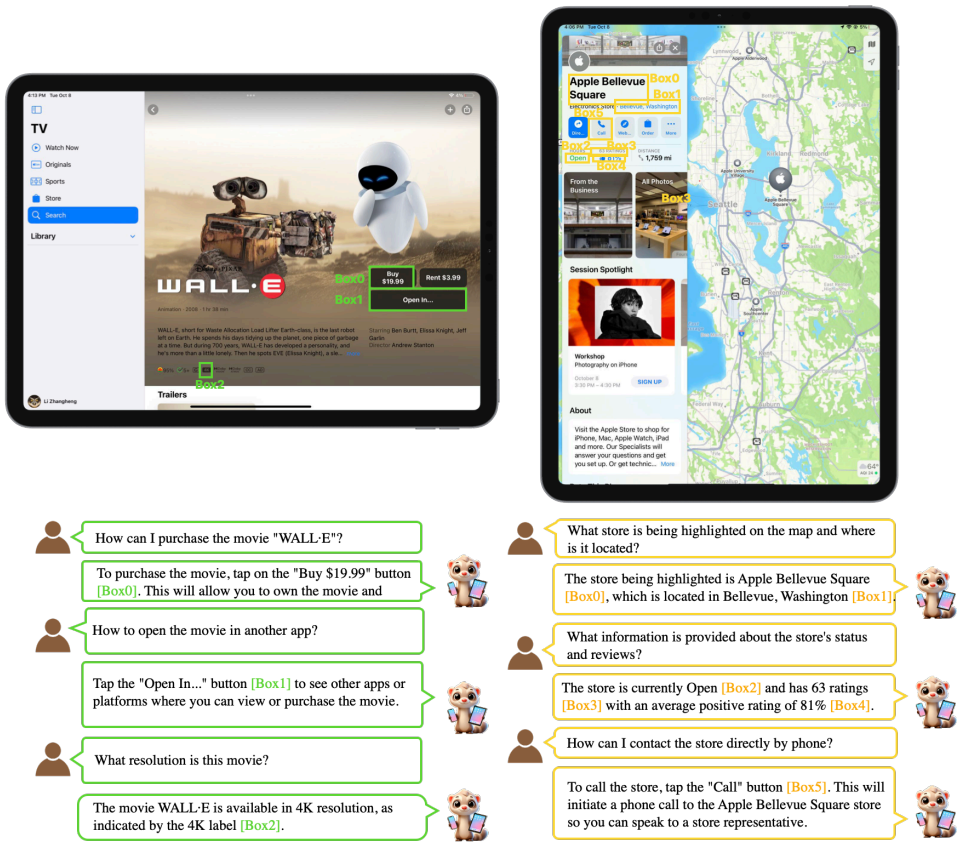


Figure 9: Real examples of the Ferret-UI One model interacting with iPad.

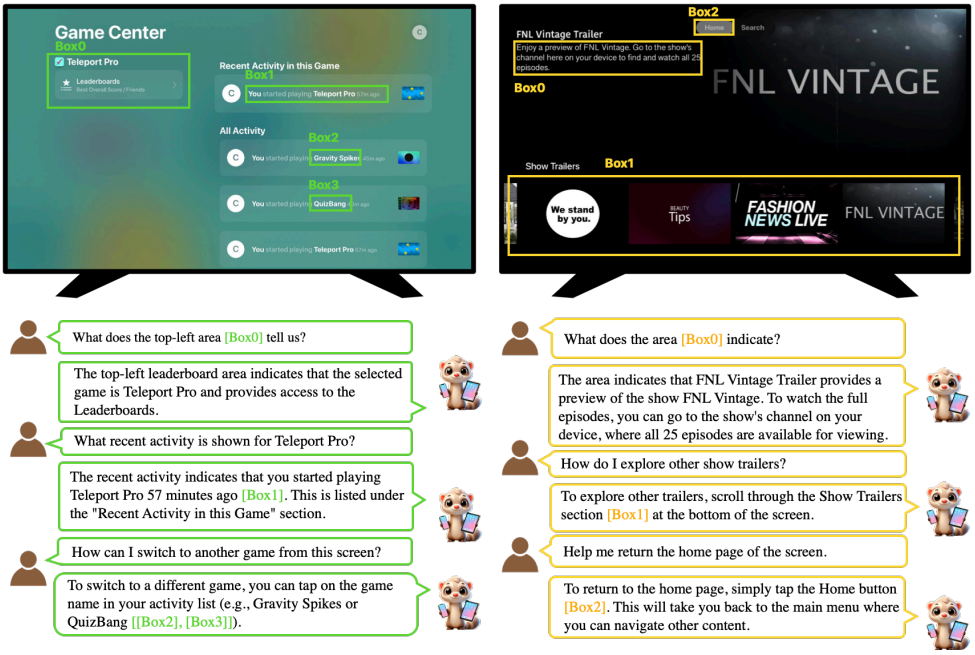


Figure 10: Real examples of the Ferret-UI One model interacting with AppleTV.

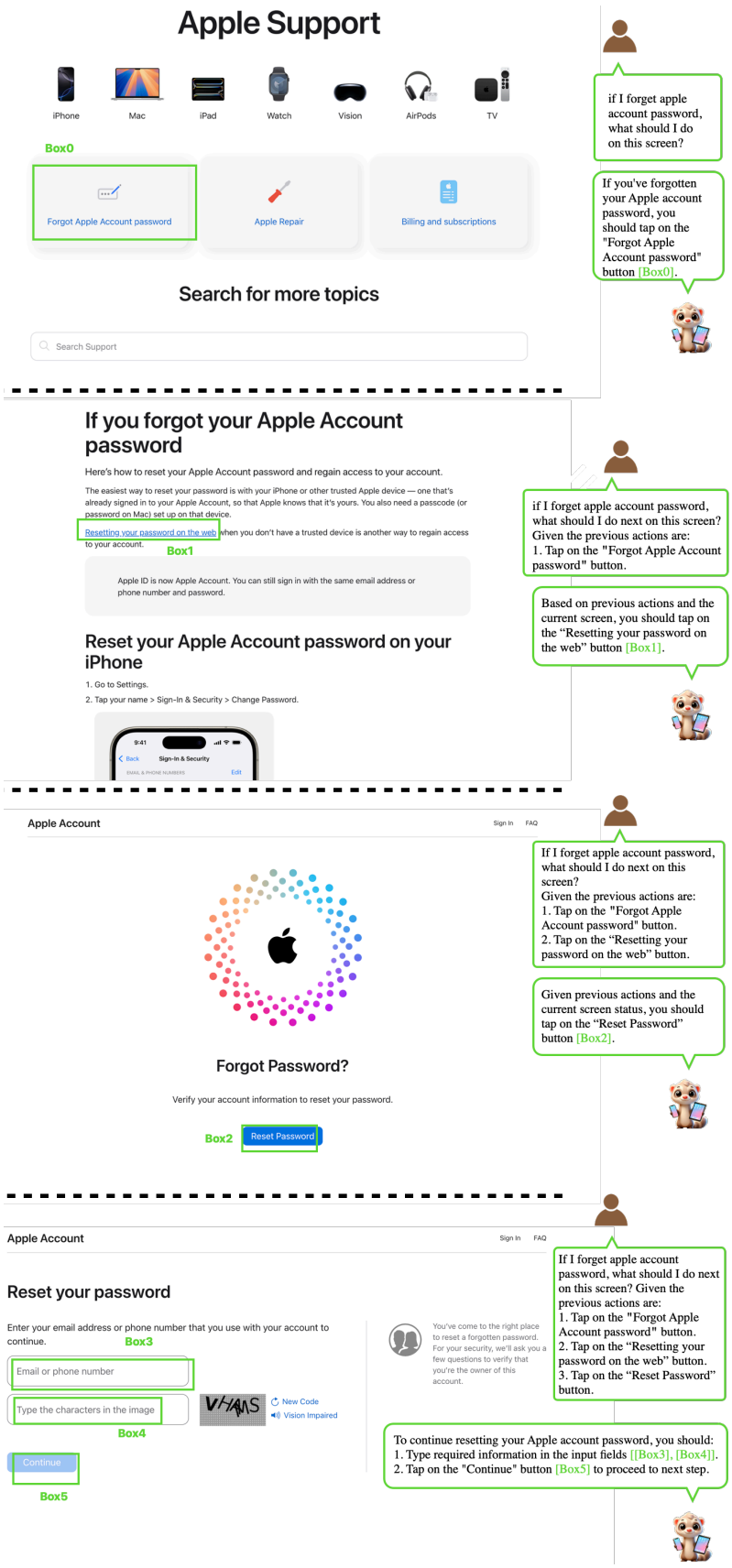


Figure 11: An example of the Ferret-UI One model performing multi-step interactions on real-time webpages following GUIDE-style QAs.

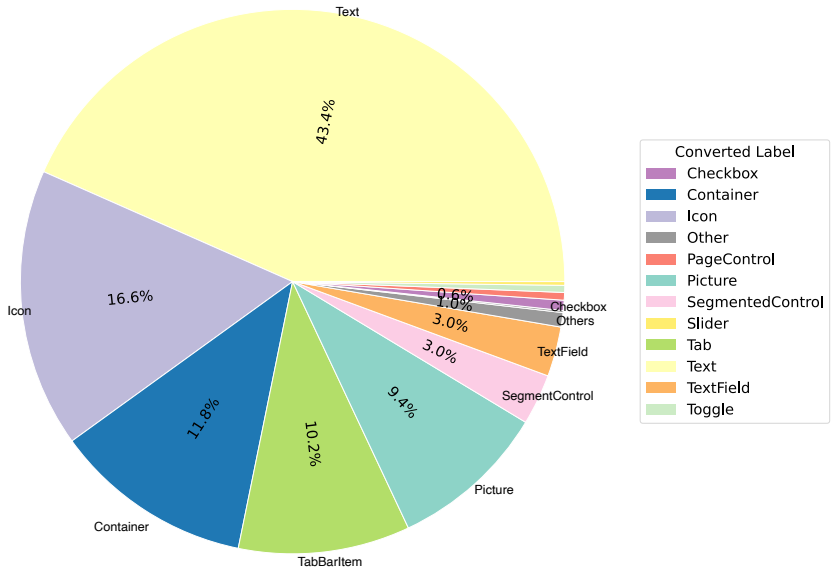


Figure 12: The statistics of original labels (**wedge**) and converted labels (**legend**) on iOS (iPhone + iPad) platforms. Each color represents one converted label, and each wedge represents one original label from source data.

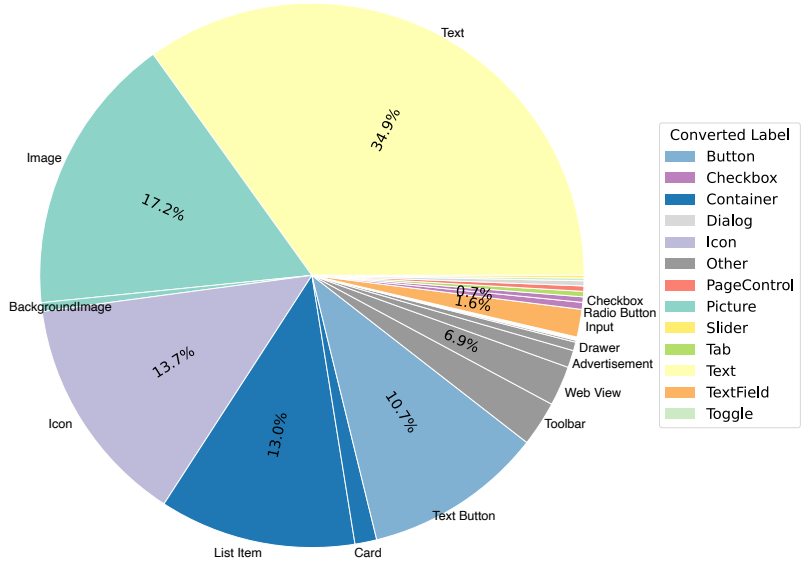


Figure 13: The statistics of original labels (**wedge**) and converted labels (**legend**) on Android platform. Each color represents one converted label, and each wedge represents one original label from source data.

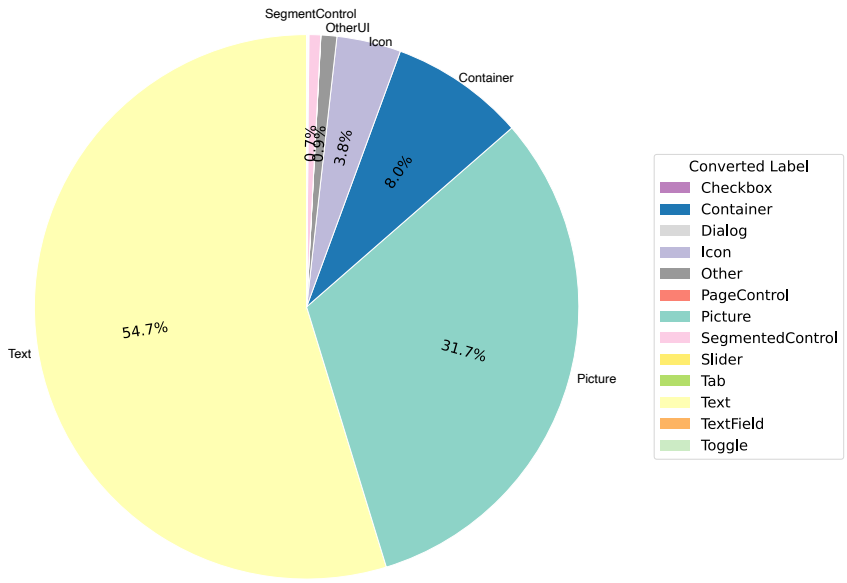


Figure 14: The statistics of original labels (**wedge**) and converted labels (**legend**) on AppleTV platform. Each color represents one converted label, and each wedge represents one original label from source data.

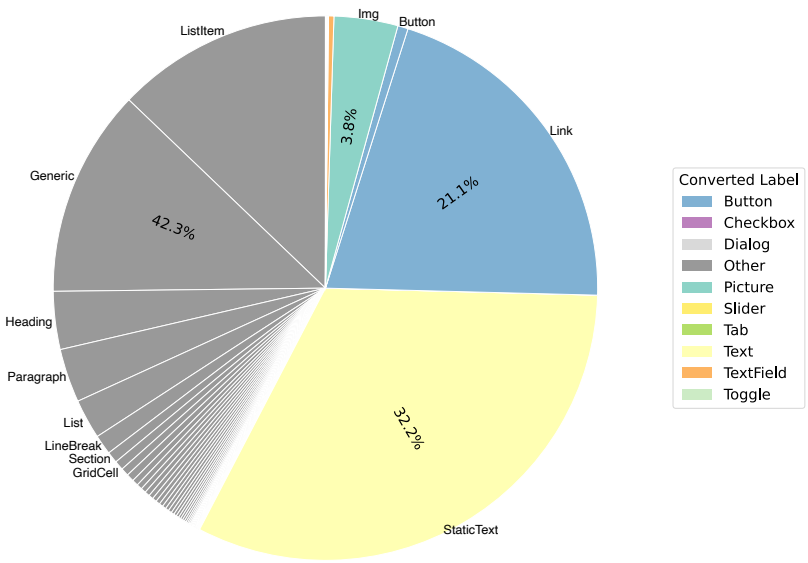


Figure 15: The statistics of original labels (**wedge**) and converted labels (**legend**) on webpage platform. Each color represents one converted label, and each wedge represents one original label from source data.

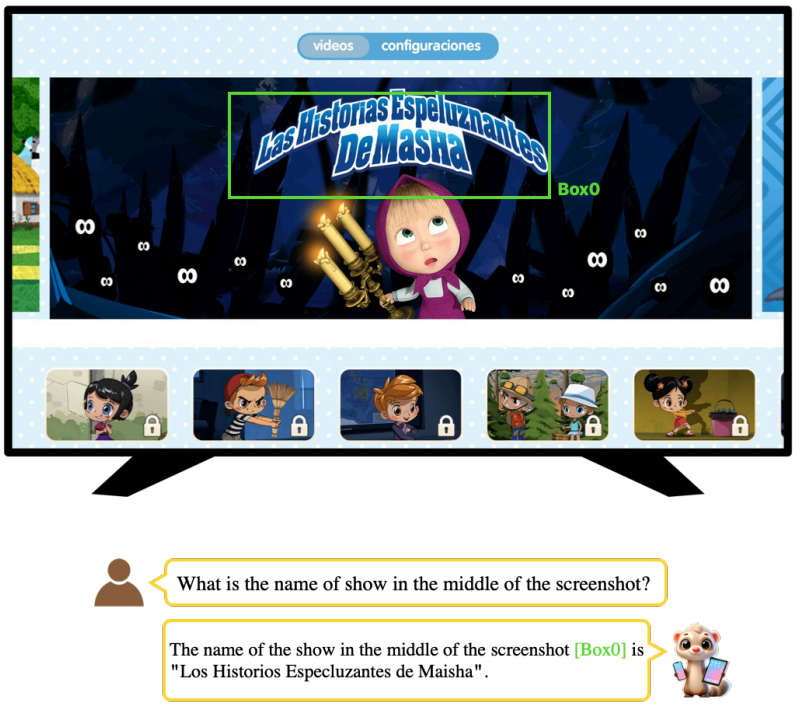


Figure 16: A typical failure case where Ferret-UI One fails to accurately recognize the rendered text contained in an image element. This is because we are unable to obtain the ground truth rendered text in the image element during dataset construction. Instead, we use OCR to generate pseudo annotations for such text, which typically introduce noises in training.

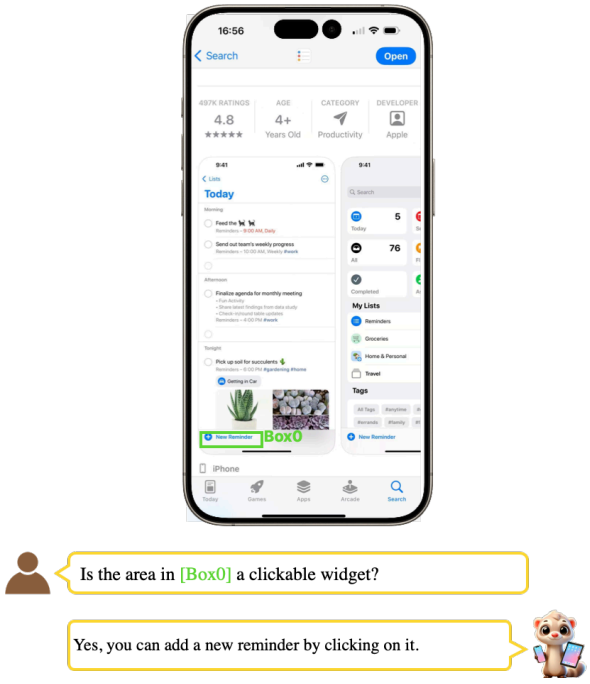


Figure 17: A typical failure case where Ferret-UI One fails to identify the area as unclickable, indicating that the model ignores the fact that the "Add new reminder" button is only a sub-area contained in an image element.

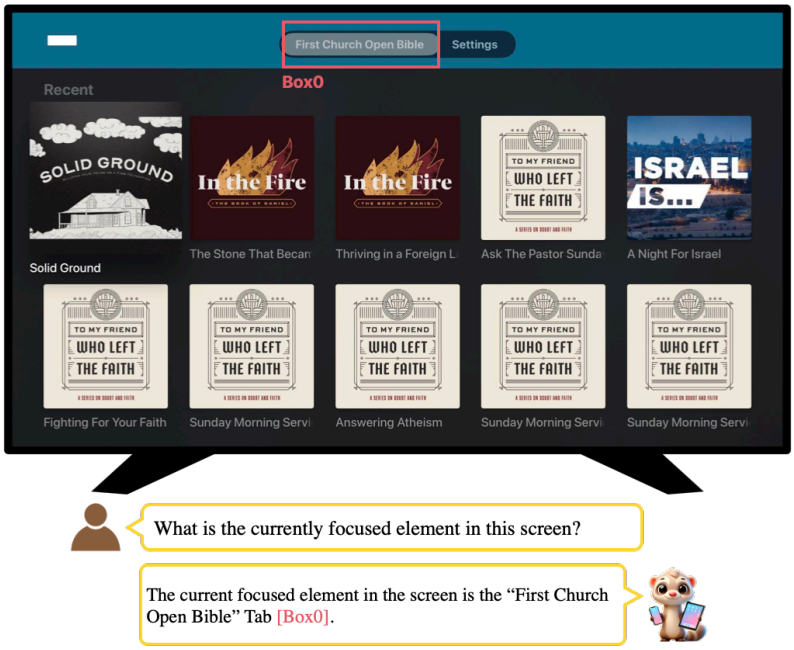


Figure 18: A typical failure case where Ferret-UI One fails to recognize the correct focused element on AppleTV. The ground truth-focused element should be on the left side, above the "Solid Ground" text. Ferret-UI One makes this error primarily because the blurry edge of this element makes it hard to judge whether this element has been enlarged.

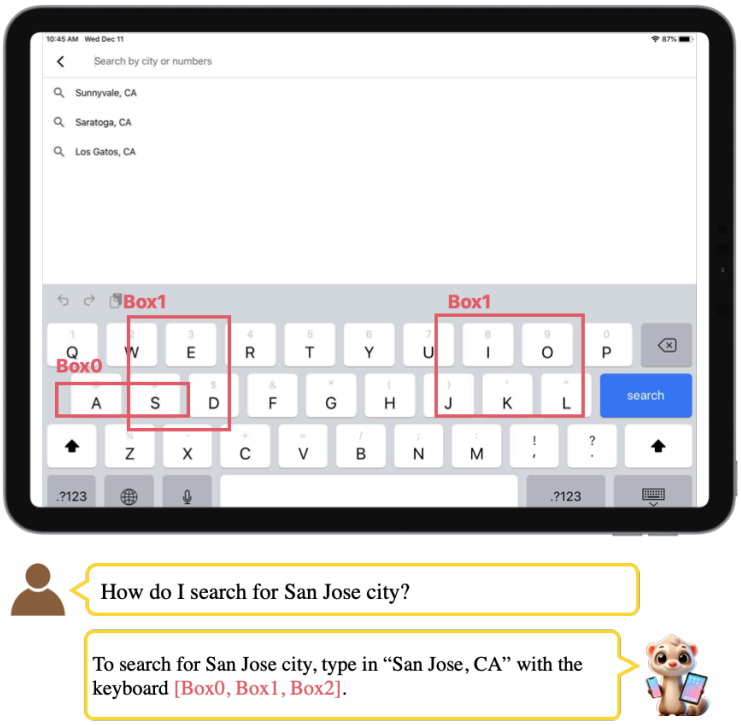


Figure 19: A typical failure case where Ferret-UI One produces multiple wrong grounding results about typing with the on-screen keyboard. This is reasonable since this requires many steps to complete the typing via the on-screen keyboard. In real-time, we can also utilize standalone typing API to help agents type instead of using the on-screen keyboard. (i.e. bypassing the on-screen keyboard.)

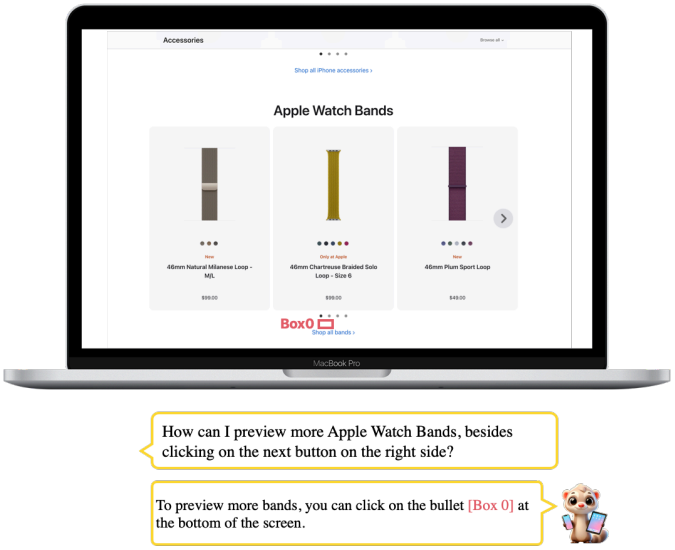


Figure 20: A typical failure case where Ferret-UI One fails to ground tiny UI widgets.

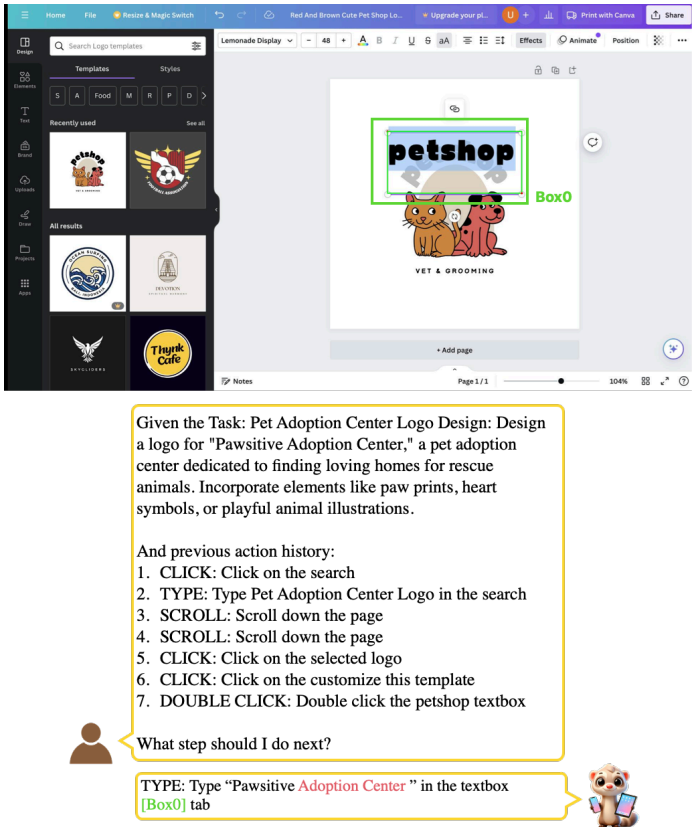


Figure 21: A typical failure case in GUIDE benchmark where Ferret-UI One fails to figure out the optimal spatial arrangement strategy that "Petshop" textbox should be only replaced with the text "Pawsitive", while text "Adoption Center" should be used later while replacing "VET & GROOMING" text bar.