

GPT-4V in Wonderland: Large Multimodal Models for Zero-Shot Smartphone GUI Navigation

An Yan^{*◊}, Zhengyuan Yang^{*♣}, Wanrong Zhu[♡], Kevin Lin[♣], Linjie Li[♣], Jianfeng Wang[♣], Jianwei Yang[♣], Yiwu Zhong[♣], Julian McAuley[◊], Jianfeng Gao[♣], Zicheng Liu[♣], Lijuan Wang[♣]

[◊]UC San Diego [♣]Microsoft Corporation [♡]UC Santa Barbara [♣]University of Wisconsin-Madison

{ayan,jmcauley}@ucsd.edu, wanrongzhu@ucsb.edu, yzhong52@wisc.edu

{zhengyang,keli,lindsey.li,jianfw,jianwei.yang,jfgao,zliu,lijuanw}@microsoft.com

Abstract

We present MM-Navigator, a GPT-4V-based agent for the smartphone graphical user interface (GUI) navigation task. MM-Navigator can interact with a smartphone screen as human users, and determine subsequent actions to fulfill given instructions. Our findings demonstrate that large multimodal models (LMMs), specifically GPT-4V, excel in zero-shot GUI navigation through its advanced screen interpretation, action reasoning, and precise action localization capabilities. We first benchmark MM-Navigator on our collected iOS screen dataset. According to human assessments, the system exhibited a 91% accuracy rate in generating reasonable action descriptions and a 75% accuracy rate in executing the correct actions for single-step instructions on iOS. Additionally, we evaluate the model on a subset of an Android screen navigation dataset, where the model outperforms previous GUI navigators in a zero-shot fashion. Our benchmark and detailed analyses aim to lay a robust groundwork for future research into the GUI navigation task. The project page is at <https://github.com/zxslp/MM-Navigator>.

1 Introduction

Building autonomous agents capable of interacting with computing devices and following human commands has been a long-standing topic in the machine learning community (Bolt, 1980; Lieberman et al., 1995). Since the advent of smartphones, there has been a practical demand for creating virtual assistants, like Siri, Cortana, and Google Assistant, which have the potential to significantly enhance user experience and assist individuals who are physically or situationally impaired. Ideally, these assistants would competently carry out everyday tasks based on natural language instructions, ranging from simple actions like setting a timer to

more complex tasks such as locating the ideal hotel for a family vacation.

Recent studies have started to explore mobile device control and smartphone task automation following human instructions (Rawles et al., 2023; Wen et al., 2023; Zhan and Zhang, 2023; Wang et al., 2023). Representative approaches include describing screen images with text and processing converted text with large language models (LLMs) (Rawles et al., 2023; Wen et al., 2023), or training a vision-language model to generate actions in a supervised manner (Rawles et al., 2023; Zhan and Zhang, 2023). However, these supervised models, when trained on specific types of screens and instructions (Rawles et al., 2023), exhibit limited effectiveness in generalizing to real-world scenarios. On the other hand, the LLM-based approaches generalize better, but the intermediate step of converting screen images to text results in information loss and consequently hurts performance. Inspired by the efficacy and broad applicability of recent large multimodal models (LMMs), we explore utilizing an LMM, GPT-4V (OpenAI, 2023a,b,c; gpt, 2023; Yang et al., 2023c), for zero-shot smartphone GUI navigation, aiming to set a new strong baseline for this intriguing task.

We identify two primary challenges for GUI navigation with LMMs, namely intended action description and localized action execution. First, the model should understand the screen image and text instruction input, and reason over the query to determine the appropriate action to take, such as providing a natural language description “clicking the Amazon icon in the third row and fourth column.” Second, the model should convert such high-level understanding into a formatted action that can be easily executed based on rules, such as “{Action: Click, Location: (0.31, 0.57)}.” In our approach, we prompt GPT-4V with an image and text for action planning, and place set-of-mark tags (Yang et al., 2023b) to anchor the generated

^{*} equal contributions

outputs. Specifically, we associate these marks with spatial locations with the help of segmentation or OCR models. To this end, our proposed GPT-4V-based system, namely MM-Navigator, can generate executable actions conditioned on the screen image, the text instruction and its interaction history.

We benchmark MM-Navigator on two datasets. We start with an iOS GUI navigation dataset with screenshots and user instructions that we manually collected. This clean analytic dataset is designed to probe insights for the two challenges in GUI navigation: intended action description and localized action execution. Human evaluations are used to assess GPT-4V on these two tasks, with accuracy rates of 91% and 75%, respectively. Additionally, we assess the model on a random subset from the recently released Android navigation benchmark (Rawles et al., 2023). We follow the proposed evaluation protocol in the benchmark, together with extra human evaluations. The strong performance demonstrates that MM-Navigator is an effective GUI navigator for smartphones, significantly outperforming previous LLM-based approaches. We provide in-depth analyses of the representative success and failure cases. We find that the current state of GPT-4V may already be effective in aiding humans in various real-world GUI navigation scenarios, as evidenced by the multi-screen results in Figure 4. However, continued enhancements are still essential to further increase the system’s reliability, as revealed in our analyses.

Our contributions are summarized as follows.

- We present MM-Navigator, an agent system built on GPT-4V for smartphone GUI navigation. MM-Navigator effectively incorporates action histories and set-of-mark tags to produce precise executable actions.
- We collect a new analytic dataset with diverse iOS screens and user instructions, which evaluates two main challenges in GUI navigation with LMMs: intended action description and localized action execution.
- We perform extensive evaluations, both automatic and human, on two datasets and provide detailed analyses. The impressive results demonstrate the effectiveness of MM-Navigator for GUI navigation.

2 Related Work

Autonomous GUI navigation. Autonomous GUI navigation involves a model following in-

structions to maneuver through different graphical user interfaces, such as websites or applications, to perform the user-queried task. Current benchmarks collected either synthetic or real-world user-generated instructions to evaluate models’ abilities in identifying specific UI elements (Shi et al., 2017; Li et al., 2020; Bai et al., 2021), or achieving overarching task objectives by interacting with a series of GUI views (Li et al., 2020; Burns et al., 2021; Venkatesh et al., 2022; Deng et al., 2023; Rawles et al., 2023). To understand the visual information from these GUI views, one line of work adopts a model structure that can process multimodal inputs (Sun et al., 2022; Redmon et al., 2016). Other methods focus on converting the UI scene text and icons into the text-only HTML format, such as single-module LLMs can process these text inputs for GUI navigation (Zhang et al., 2021; Rawles et al., 2023; Wen et al., 2023).

Multimodal agents. Recent advancements in LLMs (Brown et al., 2020; OpenAI, 2023a; Chowdhery et al., 2022; Anil et al., 2023; Touvron et al., 2023; Hoffmann et al., 2022) have catalyzed the exploration of LLM-based agent systems (Madaan et al., 2023; Shinn et al., 2023; Pan et al., 2023; Yao et al., 2022; Schick et al., 2023; Paranjape et al., 2023; Pryzant et al., 2023; Guo et al., 2023; Zhao et al., 2023; Yang et al., 2023a), which integrate reasoning logic and external tools for a variety of complex language tasks. Inspired by the success in the NLP domain, multimodal researchers delve into multimodal agents. The line of research begins with LLM-based multimodal agents (Gupta and Kembhavi, 2023; Surís et al., 2023; Wu et al., 2023; Yang* et al., 2023; Shen et al., 2023; Lu et al., 2023; Yu et al., 2023; Li et al., 2023), such as MM-ReAct (Yang* et al., 2023) for advanced visual reasoning and Visual ChatGPT (Wu et al., 2023) for iterative visual generation and editing. Propelled by the rapid advancements of LMMs (Alayrac et al., 2022; Driess et al., 2023; OpenAI, 2023a,b,c; gpt, 2023; Yang et al., 2023c; Google, 2023), the latest studies have begun to investigate the LMM-powered multimodal agents (Yang et al., 2023; Liu et al., 2023), thereby surpassing the need for basic visual description tools like caption models (Wang et al., 2022a; Wu et al., 2022). Our proposed methodology represents a specialized LMM-based agent for GUI navigation. We aim to provide a comprehensive analysis and a strong baseline for this task.

3 MM-Navigator

3.1 Problem Formulation

When presented with a user instruction X_{instr} in natural language, the agent is asked to complete a series of actions on the smartphone to complete this instruction. The entire process of agent-environment interactions from initial to final states is called an episode. At each time step t of an episode, the agent will be given a screenshot I^t , and decide the next step action to take in order to complete the task.

3.2 Screen Grounding and Navigation via Set of Mark

GPT-4V serves as a multimodal model that takes visual images and text as inputs and produces text output. One challenge is how do we communicate with GPT-4V to perform actions on screen. A possible solution is to ask the model to reason about coordinates to click given a screen. However, based on our preliminary exploration, though GPT-4V have a good understanding of the screen and approximately where to click to perform an instruction by describing the corresponding icon or text, it appears to be bad at estimating accurate numerical coordinates.

Therefore, in this paper, we seek a new approach, to communicate with GPT-4V via Set-of-Mark prompting (Yang et al., 2023b) on the screen. Specifically, given a screen, we will detect UI elements via the OCR tool and IconNet (Sunkara et al., 2022). Each element has a bounding box and either OCR-detected text or an icon class label (one of the possible 96 icon types detected by (Sunkara et al., 2022)) are contained. At each step time t , we add numeric tags to those elements, and present GPT-4V with the original screen I^t and the screen with tags I_{tags}^t . The output text Y_{action} of GPT-4V will be conditioned on the two images. If GPT-4V decides to click somewhere on the screen, it will choose from the available numeric tags. In practice, we found this simple method works well, setting up a strong baseline for screen navigation with large multimodal models.

3.3 History Generation via Multimodal Self Summarization

Set-of-Mark prompting bridges the gap between text outputs from GPT-4V and executable localized actions. However, the agent’s ability to maintain a

Table 1: Zero-shot GPT-4V iOS screen navigation accuracy on the “intended action description” and “localized action execution” tasks, respectively.

Setting	Accuracy
Intended Action Description	50/55 = 90.9%
Localized Action Execution	41/55 = 74.5%

historical context is equally important in successfully completing tasks on smartphones. The key difficulty lies in devising a strategy that allows the agent to effectively determine the subsequent action at each stage of an episode, taking into account both its prior interactions with the environment and the present state of the screen. The naive approach of feeding all historical screens or actions into the agent is computationally expensive and may decrease the performance due to information overload. For example, screens at each step can change rapidly, and most of the historical screen information is not useful for reasoning about future actions. Humans, on the other hand, can keep track of a short memory of the key information after performing a sequence of actions. We aim to find a more concise representation than a sequence of screens or actions. Specifically, at each time step, we ask GPT-4V to perform multimodal self summarization, which converts the historical actions and current step information into a concise history in the form of natural language, which is formulated as follows:

$$Y_{action}^t = \Theta_{gpt}(X_{instr}, I^t, I_{tags}^t, Y_{history}^{t-1}) \quad (1)$$

$$Y_{history}^t = \Theta_{gpt}(Y_{action}^t, Y_{history}^{t-1}) \quad (2)$$

where Y_{action}^t is the action to take at current step t , $Y_{history}^t$ is the summarized history based on Y_{action}^t and $Y_{history}^{t-1}$, Θ_{gpt} is the parameterized GPT-4V model. In this way, the trace of history will be generated auto-regressively when an episode is rolled out.

4 iOS Screen Navigation Experiment

4.1 Experimental Setup

Dataset We begin by conducting analytical experiments on iOS screens to understand GPT-4V’s capability in GUI navigation. Successfully operating smartphones in a human-like manner involves different types of screen understanding abilities. Firstly, there is the semantic reasoning ability, which involves comprehending screen inputs and

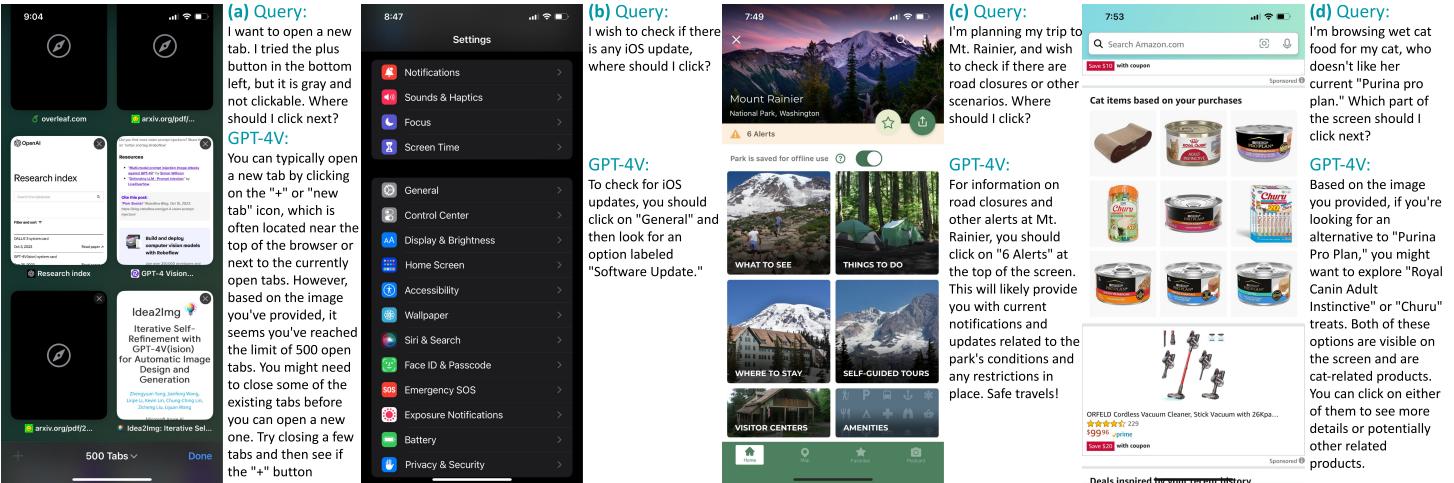


Figure 1: Intended action description examples. Best viewed by zooming in on the screen.

articulating the necessary actions to fulfill given instructions. Secondly, there is the need to translate these action descriptions into specific localized actions, such as determining the precise location for a screen click. Correspondingly, we develop two sets of test screens to disentangle these two aspects, which are referred to as “intended action description” and “localized action execution,” respectively.

In this study, we gather 110 instructions from 6 human annotators, evenly divided into two distinct sets, containing iOS screens with and without added marks. The first set, “intended action description,” involves GPT-4V taking an iOS screenshot image and an instruction as inputs, and generating an open-ended text description of the desired action to perform. This set aims to assess GPT-4V’s ability to reason the correct action to perform. Moving beyond having someone click the screen for GPT-4V (Yang et al., 2023c; Lin et al., 2023), we investigate directly generating formatted executable actions. In the second set, “localized action execution,” we add marks (Yang et al., 2023b) to ground screen locations with interactive SAM (Kirillov et al., 2023), and let GPT-4V use the mark indexes to perform localized actions. Other approaches, such as textualized box coordinates (Chen et al., 2022; Yang et al., 2022; Wang et al., 2022b), screen visual grounding (Yu et al., 2016; Mao et al., 2016; Plummer et al., 2015; Yang et al., 2019; Deng et al., 2021), object detectors (Ren et al., 2015; Carion et al., 2020) could also translate action descriptions into executable actions.

Human evaluation metrics. We use human evaluation for the analytical experiments on iOS screens, with a binary score for each sample indicating if the output is correct. For “intended action description”, human annotators determine if the output text description could lead to the correct output. For “localized action execution,” human annotators assess if clicking the location (*i.e.*, location of the selected mark) fulfills the given instruction. Each sample is assigned a binary score, either 0 or 1, to reflect its correctness.

4.2 Intended Action Description

Table 1 reports an accuracy of 90.9% on generating the correct intended action description, quantitatively supporting GPT-4V’s capability in understanding the screen actions to perform (Yang et al., 2023c; Lin et al., 2023). Figure 1 showcases representative screen understanding examples. Given a screen and a text instruction, GPT-4V gives a text description of its intended next move. For example, in Figure 1(a), GPT-4V understands the Safari browser limits of “the limit of 500 open tabs,” and suggests “Try closing a few tabs and then see if the “+” button becomes clickable.” Another example is telling the procedure for iOS update: “You should click on “General” and then look for an option labeled “Software Update” in (b). GPT-4V also effectively understands complicated screens with multiple images and icons. For example, in (c), GPT-4V mentions, “For information on road closures and other alerts at Mt. Rainier, you should click on “6 Alerts” at the top of the screen.” Figure 1(d) gives an example in online shopping, where GPT-4V suggests the correct product to check based on the user

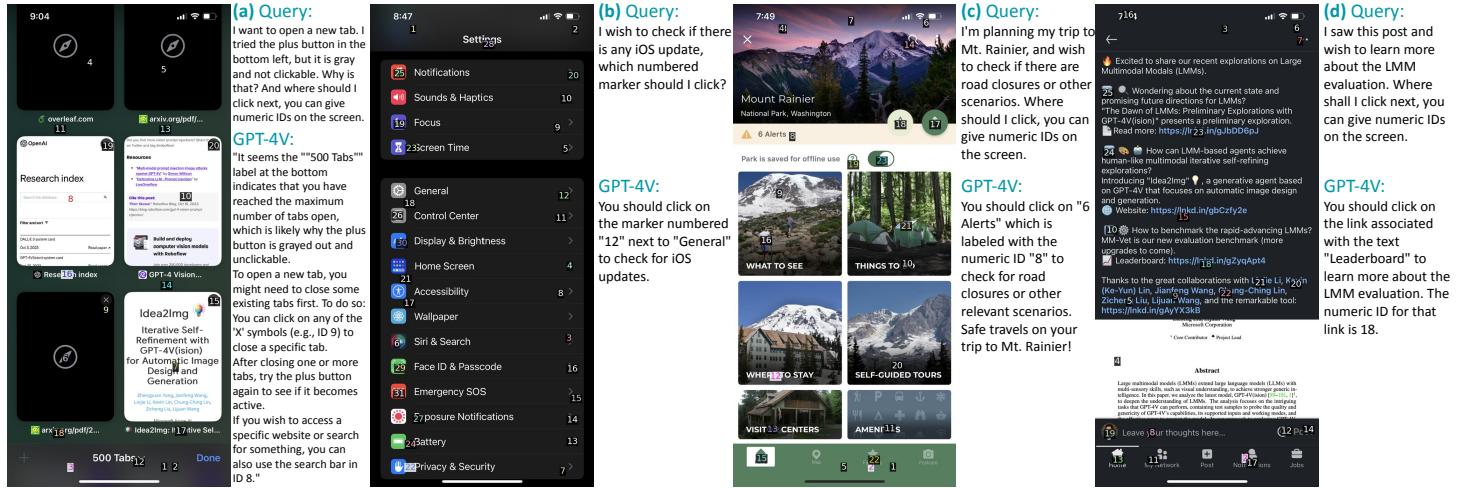


Figure 2: Localized action execution examples. Best viewed by zooming in on the screen.

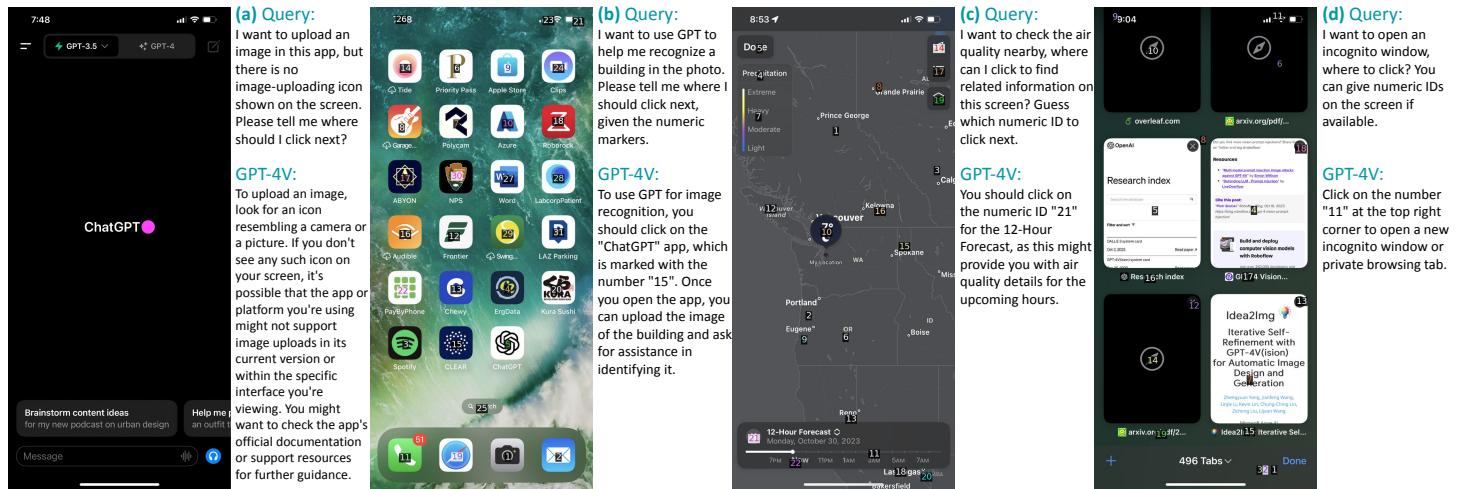


Figure 3: Representative failure cases in iOS screen navigation. Best viewed by zooming in on the screen.

input of the desired “wet cat food.”

4.3 Localized Action Execution

A natural question is how reliable GPT-4V can convert its understanding of the screen into executable actions. Table 1 shows an accuracy of 74.5% on selecting the location that could lead to the desired outcome. Figure 2 shows the added marks with interactive SAM (Yang et al., 2023b; Kirillov et al., 2023), and the corresponding GPT-4V outputs. As shown in Figure 2(a), GPT-4V can select the “X” symbol (ID: 9) to close the tabs, echoing its previous description in Figure 1(a). GPT-4V is also capable of selecting the correct location to click from the large portion of clickable icons, such as the screen shown in (b). Figure 1(c) represents a complicated screen with various images and icons, where GPT-4V can select the correct mark 8 for

the reading the “6 Alerts.” Within a screen with various texts, such as (d), GPT-4V can identify the clickable web links, and locate the queried one with the correct position 18.

4.4 The Current State with GPT-4V

From the analytical experiments on iOS screens, we find GPT-4V is capable of performing GUI navigation. Although several types of failure cases still occur, as outlined below, MM-Navigator shows promise for executing multi-screen navigation to fulfill real-world smartphone use cases. We conclude the section with qualitative results on such episode-level navigation queries.

Failure cases. Despite the promising results, GPT-4V does make errors in the zero-shot screen navigation task, as shown in Table 1. These errors are illustrated through representative failure cases

Instruction query:

You are asked to shop for a milk frother, your budget is between \$50 and \$100.

Prompt for all steps:

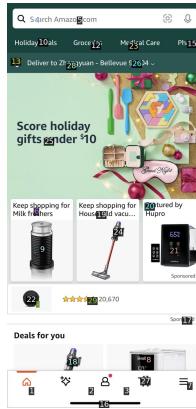
Imagine that you are a robot operating an iPhone.

You are asked to shop for a milk frother, your budget is between \$50 and \$100.

Your previous action is:

....

Below is what you see on the screen, with numeric IDs added. For the next action, which ID would you click on the screen next?



(1) GPT-4V:

To shop for a milk frother within the specified budget using the iPhone interface shown in the image, I would select an application that is capable of online shopping. In this scenario, I would click on ID 16 to use the Amazon app, as it is a comprehensive and user-friendly platform for purchasing a wide variety of products, including kitchen appliances like milk frothers.

(2) GPT-4V:

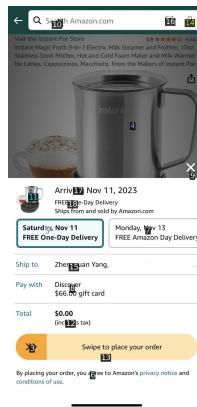
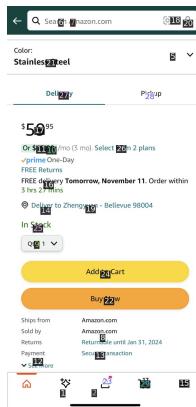
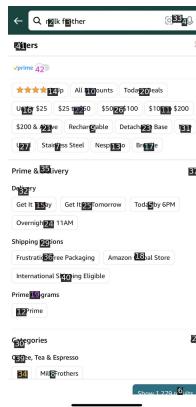
To proceed with shopping for a milk frother within the specified budget using the Amazon app, the next step would be to click on ID 6, which represents the Amazon app on the screen. This will open the app and allow me to search for milk frothers within the budget range.

(3) GPT-4V:

I would click on the search bar at the top of the Amazon app interface, which is typically where you would enter a search query. This search bar is labeled with ID 5. By clicking on this, I can then type in "milk frother" and set the price filter to show results that are between \$50 and \$100.

(4) GPT-4V:

The next step would be to refine the search results by setting the price filter. In this interface, the ID most likely to represent the filter or sorting options is 8 labeled "Filters," which seems to be the right place to start refining the search results according to the budget. So the next action would be to click on ID 8 to access filtering options, including setting the desired price range.



(5) GPT-4V:

The logical next step would be to click on the price filter that matches your budget range. Looking at the screen, you would choose the price filter that includes \$50 to \$100. The closest available options are ID 25 for "\$25 to \$50" and ID 26 for "\$50 to \$100". Since your budget starts at \$50, you would click on ID 26 to filter the search results to show only milk frothers that are priced between \$50 and \$100.

(6) GPT-4V:

The next action would be to click on ID 13, "All results," to view all available milk frothers within the selected price range of \$50 to \$100.

(7) GPT-4V:

The next action would be to click on ID 7, which shows the "Instant Magic Froth 9-in-1 Electric Milk Steamer and Frother" listed at \$59.95. This item is within the specified budget of \$50 to \$100.

(8) GPT-4V:

The next action would be to click on ID 22, which is the "Buy Now" button for the item that was selected in the previous action, the "Instant Magic Froth 9-in-1 Electric Milk Steamer and Frother" priced at \$59.95.

(9) GPT-4V:

The next action would be to click on ID 3, which is the button to "Swipe to place your order". This would finalize the purchase of the milk frother selected.

Figure 4: Episode examples on iOS screen navigation. Best viewed by zooming in on the screen.

as follows. **(a)** GPT-4V might not generate the correct answer in a single step when the query involves knowledge the model lacks. For example, GPT-4V is not aware that only "GPT-4" can support image uploads, hence it fails to click the "GPT-4" icon before attempting to find the image uploading function. **(b)** Although usually reliable, GPT-4V might still select the incorrect location. An example of this is selecting the mark 15 for the "ChatGPT" app instead of the correct mark 5. **(c)** In complex scenarios, GPT-4V's initial guess might not be correct, such as clicking the "numeric ID 21 for the 12-Hour Forecast" instead of the correct answer of mark 19. **(d)** When the correct clickable area is not marked, like a "+" icon without any marks,

GPT-4V cannot identify the correct location and may reference an incorrect mark instead. Finally, we note that many of those single-step failures may be corrected with iterative explorations, leading to the correct episode-level outcome.

From single screens to complete episodes. MM-Navigator shows an impressive capability in performing GUI navigation in a zero-shot manner. We further extend MM-Navigator from processing a single cellphone screen to recursively processing an episode of screen inputs. Figure 4 shows the qualitative result. In each step, we include the objective, "You are asked to shop for a milk frother, your budget is between \$50 and \$100." and its previous

Table 2: The Android in the Wild (AITW) dataset (Rawles et al., 2023) statistics.

Dataset	Episodes	Screens	Instructions
General	9,476	85,413	545
Install	25,760	250,058	688
GoogleApps	625,542	4,903,601	306
Single	26,303	85,668	15,366
WebShopping	28,061	365,253	13,473

action in the prompt to GPT-4V. We show that the model can effectively perform multi-step reasoning to accomplish the given shopping instruction.

5 Android Screen Navigation Experiment

5.1 Experimental Setup

Dataset. We use the AITW dataset (Rawles et al., 2023) for our evaluation on Android screen navigation. AITW is a large-scale benchmark dataset for UI control, which contains natural language instructions, screenshots on different Android systems with different resolutions, and user-annotated actions. It covers diverse multi-step tasks such as various web and application operations, app installation, and tasks with Google apps, with 715K episodes and 30K unique instructions in total. Table 2 shows the basic statistics of the dataset. We follow the split from previous work (Zhan and Zhang, 2023). Following the previous experiment setting (Rawles et al., 2023) that evaluates PaLM 2 on a randomly sampled 288 episodes, we sample 300 episodes from the test split as our test set.

Metrics. Following previous work (Rawles et al., 2023; Zhan and Zhang, 2023), we compute the screen-wise partial action matching score as the main evaluation metric, defined as the number of correct actions divided by the episode length, then this score is averaged over all tested episodes. A predicted action from GPT-4V is considered correct if both the action type and gesture match the gold ones, i.e., user actions. For click actions, it is considered correct if the selected element falls within a 14% screen distance from the gold gestures or occurs within the same detected bounding box with user gestures. For scroll actions, it is considered correct if the selected direction has the same scroll direction (up, down, left, and right) as user gestures. The partial score has been shown to correlate with the task complete score estimated by human evaluations (Rawles et al., 2023) to measure the action success rate of this task.

Baselines. We compare with the following baselines (Rawles et al., 2023; Zhan and Zhang, 2023):

- PaLM-2 ZS (Rawles et al., 2023): Zero-shot performance with PaLM-2 (Anil et al., 2023), by feeding a textual description of the screen and ask it to predict an action among the supported actions in AITW. We adopt a previously proposed LLM-based design for device control (Wang et al., 2023), where the input screen description is converted to HTML syntax.
- PaLM-2 5-shot (Rawles et al., 2023): Five examples of navigation are designed as Chain-of-thought prompts. The history of prior actions taken by the agent is also fed into the model input.
- ChatGPT 5-shot (Zhan and Zhang, 2023). The input prompts are of the same format as PaLM-2 5-shot. Experiments are conducted via the ChatGPT API.
- Fine-tuned Llama-2 (Zhan and Zhang, 2023): Fine-tuning Llama-2 model (Touvron et al., 2023) with LoRA (Hu et al., 2021), by feeding the model with the user instruction and screen descriptions in HTML syntax (the same that are used for in-context learning LLMs) and predict user actions. The model is fine-tuned with 1% randomly sampled training data to help adapt to this task.

5.2 Performance Comparison

Our main results are shown in Table 3. First, GPT-4V outperforms previous LLMs that take ground-truth descriptions of the screens as inputs. Compared with previous text-only LLMs, taking screen images as visual inputs provides an easier way for human-model interactions. It also better preserves the screen information and avoids the information loss when converting screens to text descriptions. Additionally, adding screen descriptions still improves the performance of GPT-4V. Giving the agent access to its historical interactions is helpful for better conditioned and grounded generation, and our in-context self-summarization module provides an efficient way to achieve this. Overall, we find GPT-4V presents a strong level of screen understanding of icons and text, showing the potential of visual-based device control with LMMs.

Table 3: Main results (%). Segment 1: fine-tuned Llama 2 baseline; Segment 2: in-context learning LLM baselines. “ZS” stands for “zero-shot.”; Segment 3: GPT-4V zero-shot results: “image-only” means only screen images are fed into the agent. “+text” adds parsed screen descriptions. “+history” allows the agent to access its history actions. “Training Free” means a model with zero-shot performance or in-context learning. “Text Free” means no parsed screen description is needed. The overall score is computed as the average over all the subsets.

Model	Training Free	Text Free	Overall	General	Install	GoogleApps	Single	WebShopping
Fine-tuned Llama 2	✗	✗	28.40	28.56	35.18	30.99	27.35	19.92
PaLM 2 ZS	✓	✗	30.90	-	-	-	-	-
PaLM 2 5-shot	✓	✗	39.60	-	-	-	-	-
ChatGPT 5-shot	✓	✗	7.72	5.93	4.38	10.47	9.39	8.42
GPT-4V ZS image-only	✓	✓	50.54	41.66	42.64	49.82	72.83	45.73
GPT-4V ZS +text	✓	✗	51.92	42.44	49.18	48.26	76.34	43.35
GPT-4V ZS +history	✓	✗	52.96	43.01	46.14	49.18	78.29	48.18

Table 4: Ablation studies on different tagging methods.

Model	Overall	General	Install	Apps	Single	Webshop
By side	48.39	35.24	42.18	42.46	81.50	40.53
Red	49.05	41.61	35.00	43.81	76.50	48.32
Center	49.72	47.93	36.06	44.54	79.50	40.58

Table 5: Ablation studies on different prompts.

Model	Overall	General	Install	Apps	Single	Webshop
Baseline	48.39	35.24	42.18	42.46	81.50	40.53
Think	46.66	35.01	39.12	40.50	76.50	42.18
Specific	48.77	50.77	40.54	42.32	69.50	40.71

5.3 Ablation Studies

For the ablation studies, we randomly sampled 50 episodes in total from 5 categories, which is a different subset used by the main results.

Different tagging methods. We first perform an ablation study to compare the performance with different methods to add tags on screen, shown in Table 4. We consider three methods: (1) *By side* which adds tags with black squares (same style as (Rawles et al., 2023) by the left side of each detected icon; (2) *Red* which uses red circles for each tag; (3) *Center* which adds tags with black squares at the center of each detected box. First, adding tags by the left side of boxes may cause problems, for example, some icons may be too close to each other, hence leading to slightly worse results. For tagging styles, we didn’t find a significant difference between red cycles and black rectangles, though empirically black rectangles (Yang et al., 2023b) perform slightly better.

Different prompts. We then perform robustness check with different prompting variants: (1) *Baseline*: Simply ask GPT-4V to take actions; (2) *Think*: Prompt GPT-4V to think step by step (Kojima et al., 2022); (3) *Detail*: Provide more context for this task. Overall, we did not observe improvements by “thinking step by step”, but adding more task descriptions helps GPT-4V to better execute actions.

5.4 Error Analysis

We look into GPT-4V prediction traces and attempt to categorize common types of errors that cause mismatching between GPT-4V predictions and human annotations.

We notice false negative cases where the mismatches are rooted in inaccurate Set-of-Mark (Yang et al., 2023b) annotation parsing or imperfect dataset annotation. In these cases, the predictions made by GPT-4V are correct after manual justification, but are classified as wrong predictions in automatic evaluation because the target regions are over-segmented (*e.g.*, Figure 5(a)(b)), or because the ground-truth annotation only covers one of the many valid actions (*e.g.*, Figure 6(a) has two Google Play logo; Figure 6(b) has multiple ways of accessing Google Search; and users may lookup “Setting” by direct search as GPT-4V, or by scrolling down as the human annotation in Figure 6(c)).

Figure 7 shows a few true negative examples of GPT-4V failing the designated tasks. In our zero-shot testing setup, GPT-4V is not provided with demonstrative examples to learn user action patterns. In this case, while users may scroll down or up to explore the GUI, we notice GPT-4V is more likely to perform the action of “click” on each screen, leading it to occasionally make short-

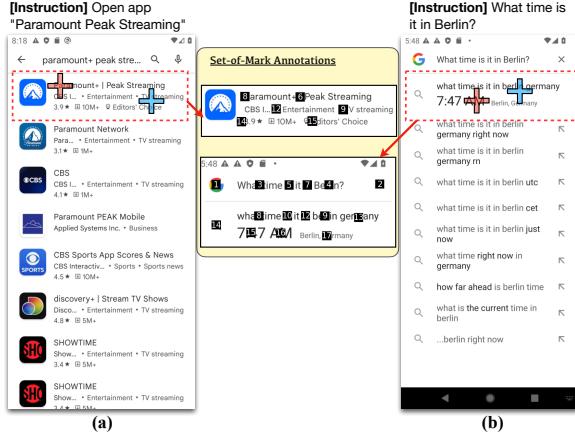


Figure 5: Examples of false negatives that are caused by inaccurate parsing in Set-of-Mark annotations. “+” denotes human annotation, and “+” is GPT-4V prediction.

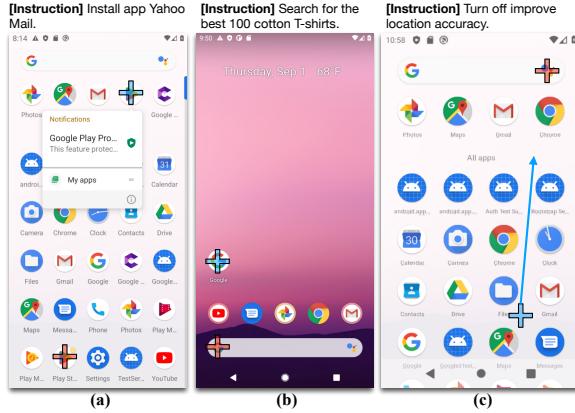


Figure 6: Examples of false negative scenarios that are caused by imperfections in ground truth dataset annotations. “+” denotes human annotation, “↗” shows the trace of scrolling, and “+” is GPT-4V prediction.

sighted decisions. In Figure 7(a), GPT-4V attempts to look for “improve location accuracy” in “Network&Internet” among the listed visible tabs, while the user decides to scroll down and look for more aligned setting tabs. In Figure 7(b), GPT-4V clicks on “Accept All”, which is not a button. In Figure 7(c), GPT-4V also shows a more literal understanding of the instruction and the current observation as in (b), clicking the “News” tab in the Google Search platform instead of actually visiting the news website.

6 Discussion

Future benchmarks for device-control. For future benchmarks, more dynamic interaction environments are needed. Even humans can make mistakes sometimes, and in this case, it is important that the evaluation benchmark would allow the model to explore and return to previous status when

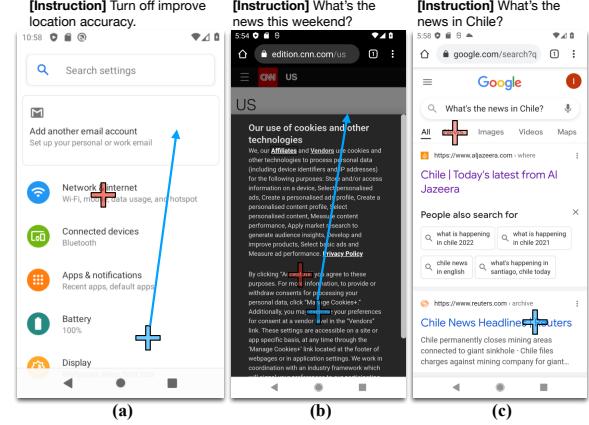


Figure 7: Examples of true negative cases where GPT-4V makes mistakes. “+” denotes human annotation, “↗” shows the trace of scrolling, and “+” is GPT-4V prediction.

a mistake is made and realized by the model. It is also interesting to explore how to automatically evaluate success rates for this task, *e.g.*, by using LMMs (Zhang et al., 2023). Another direction is to build GUI navigation datasets with different devices and diverse contents, *e.g.*, personal computers and iPads.

Error correction. A pretrained LMM may make mistakes due to data or algorithm bias. For example, if the agent fails to complete tasks in certain novel settings, how do we correct its errors to avoid mistakes in the future? Moreover, it would be interesting to study this in a continual learning setting, where the agent keeps interacting with new environments and receives new feedback continually.

Model distillation. Using a large-scale model such as GPT-4V for GUI navigation is costly. In the future, it would be interesting to explore model distillation (Polino et al., 2018) for this task, to obtain a much smaller model with competitive navigation performance, which may achieve lower latency and higher efficiency.

7 Conclusion

We have presented MM-Navigator, a GPT-4V-based multimodal agent system designed for the GUI navigation task. The system is benchmarked on both our collected iOS dataset and a public Android navigation dataset, revealing GPT-4V’s exceptional capabilities in understanding, reasoning, and planning over the screen environments. For future works, one promising direction is to establish a simulator-based benchmark that incorporates multi-step and episode-level automatic evaluations.

References

2023. Chatgpt can now see, hear, and speak. <https://openai.com/blog/chatgpt-can-now-see-hear-and-speak>.
- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.
- Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.
- Chongyang Bai, Xiaoxue Zang, Ying Xu, Srinivas Sunkara, Abhinav Rastogi, Jindong Chen, and Blaise Agüera y Arcas. 2021. Uibert: Learning generic multimodal representations for ui understanding. In *International Joint Conference on Artificial Intelligence*.
- Richard A Bolt. 1980. “put-that-there” voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*.
- Andrea Burns, Deniz Arsan, Sanjna Agrawal, Ranjitha Kumar, Kate Saenko, and Bryan A. Plummer. 2021. Mobile app tasks with iterative feedback (motif): Addressing task feasibility in interactive visual environments. *ArXiv*, abs/2104.08560.
- Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *ECCV*.
- Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. 2022. Pix2seq: A language modeling framework for object detection. In *ICLR*.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. 2021. Transvg: End-to-end visual grounding with transformers. In *ICCV*.
- Xiang Deng, Yu Gu, Bo Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *ArXiv*, abs/2306.06070.
- Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*.
- Google. 2023. Bard. <https://bard.google.com>. Accessed: 2023-07-17.
- Yiduo Guo, Yaobo Liang, Chenfei Wu, Wenshan Wu, Dongyan Zhao, and Nan Duan. 2023. Learning to program with natural language. *arXiv preprint arXiv:2304.10464*.
- Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643*.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. 2023. Multimodal foundation models: From specialists to general-purpose assistants. *arXiv preprint arXiv:2309.10020*.
- Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. 2020. Mapping natural language instructions to mobile UI action sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8198–8210, Online. Association for Computational Linguistics.
- Henry Lieberman et al. 1995. Letizia: An agent that assists web browsing. *IJCAI(1)*, 1995:924–929.

- Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, et al. 2023. Mmvid: Advancing video understanding with gpt-4v (ision). *arXiv preprint arXiv:2310.19773*.
- Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, Lei Zhang, Jianfeng Gao, and Chunyuan Li. 2023. Llava-plus: Learning to use tools for creating multimodal agents.
- Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv preprint arXiv:2304.09842*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.
- Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *CVPR*.
- OpenAI. 2023a. [Gpt-4 technical report](#).
- OpenAI. 2023b. [Gpt-4v\(ision\) system card](#).
- OpenAI. 2023c. Gpt-4v(ision) technical work and authors. <https://cdn.openai.com/contributions/gpt-4v.pdf>.
- Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*.
- Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Túlio Ribeiro. 2023. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014*.
- Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*.
- Antonio Polino, Razvan Pascanu, and Dan Alistarh. 2018. Model compression via distillation and quantization. *arXiv preprint arXiv:1802.05668*.
- Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chen-guang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.
- Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy P. Lillicrap. 2023. [Android in the wild: A large-scale dataset for android device control](#). *ArXiv*, abs/2307.10088.
- Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *CVPR*.
- Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.
- Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugging-gpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580*.
- Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. [World of bits: An open-domain platform for web-based agents](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3135–3144. PMLR.
- Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: Language agents with verbal reinforcement learning](#).
- Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. 2022. [Meta-gui: Towards multi-modal conversational agents on mobile gui](#). In *Conference on Empirical Methods in Natural Language Processing*.
- Srinivas Sunkara, Maria Wang, Lijuan Liu, Gilles Baechler, Yu-Chung Hsiao, Abhanshu Sharma, James Stout, et al. 2022. Towards better semantic understanding of mobile interfaces. *arXiv preprint arXiv:2210.02663*.
- Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. [Vipergpt: Visual inference via python execution for reasoning](#). *arXiv preprint arXiv:2303.08128*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Sagar Gubbi Venkatesh, Partha P. Talukdar, and Srini Narayanan. 2022. [Ugif: Ui grounded instruction following](#). *ArXiv*, abs/2211.07615.
- Bryan Wang, Gang Li, and Yang Li. 2023. Enabling conversational interaction with mobile ui using large language models. In *Proceedings of the 2023 CHI*

Conference on Human Factors in Computing Systems, pages 1–17.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022a. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022b. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR.

Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. 2023. Empowering llm to use smartphone for intelligent task automation. *arXiv preprint arXiv:2308.15272*.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*.

Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. 2022. Grit: A generative region-to-text transformer for object understanding. *arXiv preprint arXiv:2212.00280*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023a. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyu Li, and Jianfeng Gao. 2023b. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2022. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *European Conference on Computer Vision*, pages 521–539. Springer.

Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. 2019. A fast and accurate one-stage approach to visual grounding. In *ICCV*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023c. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*.

Zhengyuan Yang*, Linjie Li*, Jianfeng Wang*, Kevin Lin*, Ehsan Azarnasab*, Faisal Ahmed*, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023.

Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.

Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. Idea2img: Iterative self-refinement with gpt-4v (ision) for automatic image design and generation. *arXiv preprint arXiv:2310.08541*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *ECCV*.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Zhuosheng Zhan and Aston Zhang. 2023. You only look at screens: Multimodal chain-of-action agents. *arXiv preprint arXiv:2309.11436*.

Xiaoyi Zhang, Lilian de Greef, Amanda Sweeny, Samuel White, Kyle I. Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, Aaron Everitt, and Jeffrey P. Bigham. 2021. Screen recognition: Creating accessibility metadata for mobile applications from pixels. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.

Xinlu Zhang, Yujie Lu, Weizhi Wang, An Yan, Jun Yan, Lianke Qin, Heng Wang, Xifeng Yan, William Yang Wang, and Linda Ruth Petzold. 2023. Gpt-4v (ision) as a generalist evaluator for vision-language tasks. *arXiv preprint arXiv:2311.01361*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2023. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144*.