VGA: Vision GUI Assistant - Minimizing Hallucinations through Image-Centric Fine-Tuning

Ziyang Meng, **Yu Dai**, **Zezheng Gong**,
Shaoxiong Guo, **Minglong Tang**, **Tongquan Wei***
East China Normal University
51255901089@stu.ecnu.edu.cn, daiyuu@126.com
{51255901046, 51265901013, 51275901042}@stu.ecnu.edu.cn
tqwei@cs.ecnu.edu.cn

Abstract

Large Vision-Language Models (VLMs) have already been applied to the understanding of Graphical User Interfaces (GUIs) and have achieved notable results. However, existing VLMs often overly rely on internal text-based knowledge while neglecting visual inputs. This imbalance may lead models to produce answers that do not align with the visual content in GUI comprehension tasks. Such inaccuracies are termed as 'hallucinations' where models generate incorrect or illogical responses upon visual verification against GUI elements. These errors result in misinterpretations and diminish the model's practical utility in applied settings. To address these issues, we introduce VGA, a fine-tuned model designed for comprehensive GUI understanding. Our model aims to balance attention image and text to enhance interpretation and reduce hallucinations. We construct a Vision Question Answering (VQA) dataset of 63.8k high-quality examples with our propose *Referent Method*, focusing on response with visual content of images. We then design a two-stage fine-tuning method to enhance both the model's accuracy to extract information from image content and alignment with human intent. Experiments show that our approach enhances the model's ability to extract information from images and achieves state-of-the-art results in GUI understanding tasks. Our dataset and fine-tuning script are available at https://github.com/Linziyang1999/VGA-visual-GUI-assistant

1 Introduction

Large Vision-Language Models (VLMs) have recently emerged as a powerful approach for various multimodal tasks. These models acquire textual knowledge through pre-training and develop image understanding abilities during instruction-tuning (Cao et al., 2023; Liu et al., 2023). VLMs can effectively process both visual and linguistic information by using a visual-language projector, which maps different types of data into a shared latent space. For instance, LLaVA (Liu et al., 2024a) and InstructBLIP (Dai et al., 2023) have demonstrated strong zero-shot capabilities in tasks like image captioning, visual reasoning, and complex conversations. UniChart (Masry et al., 2023), on the other hand, has demonstrated the powerful capabilities of VLMs in the understanding of formatted charts.

With the proliferation of mobile applications, the importance of Graphical User Interfaces (GUIs), which serve as a critical bridge between end users and applications, has increasingly garnered scholarly attention. GUIs, characterized by their structured layouts, rich graphical and textual content, and the inclusion of human operational logic (Banerjee et al., 2013), present a complex challenge: *Can the success of VLMs be applied to the GUI domain?*

Traditional GUI comprehension method focus on conveying the actual user interface interactions and mirror the user's direct experience with the GUI. For instance, LabelDroid (Chen et al., 2020) utilizes deep learning to predict labels for image-based buttons from a variety of commercial apps available on Google Play. Similarly, TANGO (Cooper et al., 2021) employs custom computer vision (CV) and text retrieval techniques to analyze visual and textual information on mobile screens.

However, traditional visual GUI comprehension methods focus mainly on identifying GUI components but fail to fully comprehend the graphical and textual information, layout, and the interaction context within the interface. In contrast, modern approaches such as those employed by ferret-UI and CogAgent, which utilize LVLMs, demonstrate significant advantages in this domain. These methods are designed to process both detailed textual and graphical content, leveraging their pre-trained capabilities. Nevertheless, both ferret-UI (You et al.,

*Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 1261–1279
November 12-16, 2024

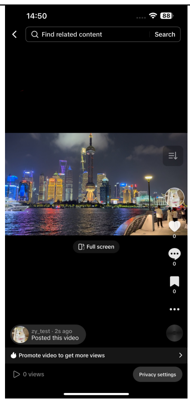
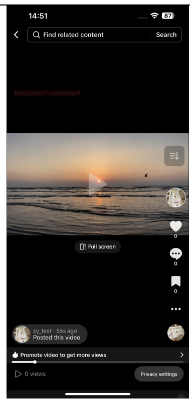
Hallucination in GUI comprehension		
		
User	How can I give a like to this video?	How can I login?
LLaVA-v1.6-34B	Open the video you want to like. Look for a thumbs-up or heart icon, which is usually located near the bottom of the screen...	Open the website or app of the social media platform you want to log in to. Look for the "Login" or "Sign In" button, which is usually located at the top right corner of the page.
GPT-4V	...there should be a heart icon or similar button typically found below the video or in a sidebar. Tapping on this icon...	Open the App or website... Find login section... Enter your credentials...
VGA	You can click the heart-shaped icon located on the right side of the screen, just below the person's image.	This page seems to be focused on displaying content and interacting with it, but it doesn't appear to have a login function.

Table 1: Example of hallucination in GUI comprehension, red means element in image relate to answer and orange means location of the element which indicate if model really extract information from image.

2024) and CogAgent (Hong et al., 2023) frequently encounter limitations by over-relying on their pre-trained knowledge, which leads to a neglect of critical visual content. This often results in the generation of inaccurate or irrelevant responses, indicating a gap in their ability to achieve a balanced and holistic understanding of GUIs (Shahgir et al., 2024; Zhang et al., 2023).

To address these issues, we propose VGA, a model fine-tuned on a self-construct 63.8K dataset, using a novel training method we design. During the dataset construction, we employ knowledge distillation from large language models to construct 63.8k dataset based on Rico (Deka et al., 2017), and we adopt the *Referent Method* to enhance the model's focus on image content by using visual and position information in constructing dataset, thereby increasing the relevance and accuracy of the responses. In the training process, we apply a Foundation and Advanced Comprehension (FAC) approach: the Foundation Stage enhances the model's understanding of GUI image, while the Advanced Comprehension stage improves the model's ability to respond to complex questions based on its understanding of the GUI. Additionally, we employ a task progression and reinforcement approach to create logical chains between tasks and within responses to strengthens the model's ability to reason and infer relationships within the provided context. Through these integrated strategies, VGA achieves state-of-the-art performance in GUI comprehension tasks. Our primary contributions are as follows:

- **Large-scale GUI corpus for LVLM Fine-tuning.** To fine-tune our model, we introduce a large-scale GUI corpus that includes a diverse array of apps accompanied by corresponding text descriptions and dialogues.
- **A Fine-tuned LVLM for GUI task.** We propose VGA, a LVLM for GUI comprehension, fine-tuned to fulfill both granular low-level and strategic high-level goals specialized for graphical user interfaces.
- **A Fine-tune Method to efficient improve LVLMs performance** We propose a two stage fine-tune method based on our dataset to achieve better understand of images.

- **Performance enhancements on real-world GUI tasks.** As shown in experiment 4, we apply our VGA to a GUI comprehension bench, yielding promising results.

2 Related Works

2.1 GUI Comprehension

GUI comprehension is essential in mobile agent design to ensure app quality and user experience. It focuses on assessing interface components such as buttons, text boxes, and menus to verify their operational performance, visual design, and usability, enabling mobile agents to interact effectively with the app interface (Yu et al., 2023b). The software engineering community has been long-term focused on the improvement of mobile app GUI comprehension effectiveness in all aspects since last century (Memon et al., 1999). The main purpose of these efforts is to understand GUI elements to advance GUI automation, including testing and other applications (Arnatovich and Wang, 2018; Vásquez et al., 2018; Said et al., 2020).

Existing methods for GUI comprehension can be divided into code-based approaches, which rely on functional specifications or code analysis (Paiva et al., 2005; El Ariss et al., 2010), and visual-based approaches, which improve performance by considering the GUI's visual representation through techniques like template matching and OCR (Cheng et al., 2019). However, these methods still lack the ability to interact with GUIs in a human-like manner. Recently, Large Vision-Language Models (LVLMs) have emerged as a promising approach by combining visual and textual understanding. These models leverage vast prior knowledge gained from pre-train to comprehend human intentions and interact more naturally with GUI elements (Wang et al., 2024; Cui et al., 2024).

In our work, we leverage the strengths of LVLMs and enhance their performance in GUI comprehension using our constructed dataset and our designed training methods.

2.2 Large Vision-Language Models

The introduction of the transformer architecture has revolutionized natural language processing, enabling models to efficiently capture long-range dependencies and contextual information. This advance laid the groundwork for pre-trained Large Language Models (LLMs). The pre-train, fine-tune, and predict paradigm (Liu et al., 2021), exemplified by models such as GPT (Radford et al., 2018) and BERT (Devlin et al., 2018), led to significant improvements in language understanding and generation (Cheng et al., 2023; Yao et al., 2023; Arefeen et al., 2024; Schick and Schütze, 2020; Yu et al., 2023a). Subsequent models like GPT-4 (Achiam et al., 2023), Llama (Touvron et al., 2023), and Qwen (Bai et al., 2023) have further expanded these capabilities. Explorations into the Mixture of Experts (MoE) architecture (Shen et al., 2023; Jiang et al., 2024) continue to enhance the scalability of Transformer-based models.

Large Vision-Language Models (LVLMs) harness the strengths of both LLMs and visual feature encoders, utilizing various methods to project visual data into an LLM-comprehensible space (Popescu et al., 2009; Liu et al., 2023, 2024b; Alayrac et al., 2022; Wang et al., 2023). Models like Flamingo (Alayrac et al., 2022) and CogVLM (Wang et al., 2023) exemplify the architectures that achieve this integration. Additionally, UniChart (Masry et al., 2023) excels in chart comprehension by leveraging LVLMs to accurately interpret formatted charts and complex informational structures. Similarly, LLaVA (Liu et al., 2023) integrates pretrained visual and language models to enhance the understanding of multimodal inputs, achieving superior performance in visual and textual comprehension tasks.

In the field of graphical user interface (GUI) analysis, traditional methods like Ferret-UI (You et al., 2024) and CogAgent (Hong et al., 2023) predominantly focus on text, largely because their Large Language Model (LLM) components have been thoroughly trained to emphasize textual information. This emphasis can lead to the visual components of the input being overlooked or misinterpreted in the responses, often resulting in the generation of irrelevant answers—a phenomenon known as "hallucination".

Inspired by these work, we investigate the performance of LVLMs in the context of complex GUI interfaces. Our study focuses on evaluating how well these models understand and interact with GUIs that feature structured layouts and rich interactions, aiming to further enhance their applicability and accuracy in real-world scenarios.

3 Problem in GUI Comprehension

The primary cause of hallucinations in Vision-Language Models (VLMs) can be attributed to their reliance on response patterns learned from tradi-

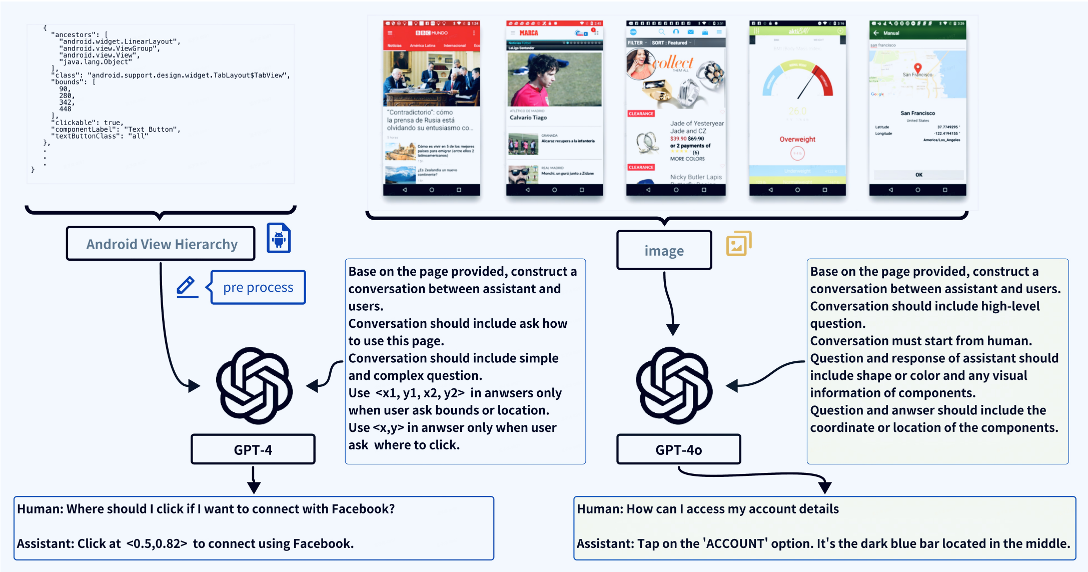


Figure 1: Data generation Method

tional Large Language Models (LLMs). In this section, we investigate the occurrence of hallucinations in contemporary GUI models and propose a method to reduce these inaccuracies.

3.1 Hallucination in GUI Comprehension

LLMs are typically trained on massive corpora of pure text data, where they learn to generate responses based on textual context and patterns. When a visual module is introduced, these learned textual response patterns can adversely influence the model's behavior during multimodal tasks.

Indeed, while Ferrent-UI and CogAgent are adept at understanding specific elements in graphical interfaces, they primarily emphasize textual and positional information when constructing their training datasets. This approach results in limited recognition capabilities for icons and other crucial visual details. Such a bias can lead to misinterpretations or incomplete understanding of GUIs that are visually complex in real-world applications, as reliance on text and layout alone may not adequately capture the full semantics and functionalities of the graphical interfaces. This models can face several types of illusions as follow:

Over-reliance on textual content: LVLMs like Ferrent-UI and CogAgent, heavily trained on text-intensive datasets and focused on text interpretation tasks, tend to overly prioritize textual data when analyzing GUIs. As a result, they might overlook integral visual cues like layout and visual styles, leading to impractical or irrelevant action suggestions within the GUI environment, as shown in Table 1. This text bias affects their capability to deliver effective and holistic interpretations, which is crucial for accurate and user-friendly interface interactions.

Word-to-image coincidences: This issue arises when models draw superficial connections between query words and text appearing on GUI elements, but these elements do not align with the intended functional requirements. As shown in Table 8, if a query includes "start painting," and there are two buttons on the GUI with "start now" and "drawing" in their labels, a model might incorrectly choose a "start now" button when the appropriate action was to "drawing". This misalignment leads to the selection of incorrect actions based on text matches rather than the functional relevance of the elements within the GUI context.

3.2 Referent Method

To address the previously mentioned issue, we propose the *referent method*. GUI design often uses layout, shape, and color to distinguish elements, providing critical visual details to guide user interactions. By explicitly incorporating coordinates, shapes, colors, and relative positions between elements in the construction of our dataset, we aim to enhance the model's focus on the image content when generating answers. Thereby reducing the chances of hallucination. During the design of our dataset, we ensured that most responses involving GUI elements include at least one of the following referents, aligning the element information with the image content:

Shape: The shape of the element, like a rectangular button with rounded corners.
Color: The color of the element, for example, a blue button with white text.
Position: The exact coordinates or bounds of the element, format as <x,y> and <x1,y1,x2,y2>.
Relative Position: The position of the element in relation to other elements, such as being below the text input field.

4 GUI Comprehension Dataset

To address the imbalance between image and text in GUI analysis, we propose a high-quality comprehension dataset tailored for LVLM training. This dataset includes detailed annotations of both textual and visual elements, ensuring a balanced and comprehensive training environment.

4.1 Existing General Dataset

Ferret-UI (You et al., 2024) and CogAgent (Hong et al., 2023), while capable of understanding certain elements in graphical interfaces, primarily focus on extracting and utilizing textual and positional information during the construction of their training datasets. This bias can lead to misunderstandings or inadequate interpretations when dealing with GUIs rich in visual information in actual applications, as relying solely on text and layout might not fully comprehend the semantics and functionalities of the graphical interfaces.

Additionally, since the datasets relied upon by these models are not open to the public, it poses additional challenges for external researchers or developers, as they might struggle to obtain data of similar quality and scale for effective model fine-tuning or further research.

4.2 Data collection

Given that both Ferrent-UI and CogAgent are not open source, the availability of the Rico dataset (Deka et al., 2017) represents a valuable resource for the research community involved in mobile app design and development.

The Rico (Deka et al., 2017) dataset is a huge GUI dataset created to support research in mobile app design and development, including areas such as GUI design, interaction. It consist of 66k unique GUI screens and 3M elements from 27 categories, over 9.3k applications. Each GUI comprise a screenshot and an augmented Android view hierarchy that capture all of the elements comprising a GUI, their properties, and relationships between them. However, the rich data provided by Rico cannot be directly utilized to train LVLMs as the text data in Rico do not align with human perception in the same manner as VQA datasets. This necessitates the transformation of this data into a format compatible with VQA dataset.

4.3 Task Design

The dataset includes the following sub-tasks, each complemented by human annotations to elevate the quality and applicability of the training data. Additionally, we have structured the dataset into two distinct categories based on these tasks: the instruction dataset and the conversation dataset. These cater to scenarios of directive compliance and multi-turn dialogue, respectively.

- *Description*: This task involves providing a basic description of the GUI's layout, identifying the function and placement of components. This foundational task facilitates deeper analytical tasks, as all GUI-related tasks rely on a clear understanding of the overall layout.

- *Bounds & Location*: This task adds complexity by requiring the model to incorporate precise descriptions of element bounds and coordination. By introducing this, our approach ensures that the model focuses on the accurate positional information within the images.

- *Function*: This task requires the model to understand the individual functions of each GUI element, considering not only the overall function but also the relative positions of elements to each other. Identical buttons might serve different purposes depending on their specific context within a GUI. while relative positioning is significant because two elements placed in a specific arrangement might together convey a particular function or information. This enables the model to predict the design intent and functionality of the entire interface, resulting in more contextually aware and precise responses.

4.4 Generation Method

However, the rich data provided by Rico cannot be directly utilized to train LVLMs as the text data in Rico do not align with human perception in the same manner as VQA datasets. This necessitates the transformation of this data into a format compatible with VQA dataset. The Android

Inst dataset	Model	Number	Conv dataset	Model	Number
description_inst	GPT-4	3k	Conv_simple	GPT-4	5.4k
bound_inst	GPT-4	4.3K	Conv_complex	GPT-4	11K
function_inst	GPT-4	2K	Conv_4o_long	GPT-4o	10K
testing_inst	GPT-4	5K	Conv_4o_short	GPT-4o	10K
function_inst_4o	GPT-4o	8K	conv_4o_miss	GPT-4o	5k
Total					63.8K

Table 2: Dataset composition

view hierarchy in the Rico dataset provides detailed descriptions of GUI elements, allowing us to obtain precise positional data and interaction details. Therefore, we use GPT-4 (text-only) to extract this textual information from the Android view hierarchy and GPT-4o to extract visual information from images as shown in Figure 1. Our approaches are as follows:

Data Pre-processing: We remove elements in the Android View Hierarchy that have the 'visible-to-user' attribute set to 'false', and normalize the 'bounds' values to a range of 0 to 1. Additionally, for all elements with the 'clickable' attribute set to 'true', we add a 'click_coordinate' attribute, calculated as the midpoint of the range indicated by the 'bounds'.

GPT-4 (Text-Only) Generated Tasks: Inspired by the method of LLaVA (Liu et al., 2024b), this category consists of dataset generated by GPT-4 (text-only), based on pre-processed Android View Hierarchy data. Leveraging the detailed information available in the Android View Hierarchy, this approach generates a variety of QA pairs that accurately reflect the textual and positional data and interactions.

GPT-4o (Image-Based) Generated Tasks: Inspired by ShareGPT4 (Chen et al., 2023), this category comprises dataset generated by GPT-4o (image-based). Given that vision models often struggle with capturing precise positional information of elements, this method focus on relative positions, shapes, and colors of the visual elements.

Our GUI comprehension dataset consists of 63.8k images, covering a diverse range of apps and tasks. It includes 22.3k instruction-following data pairs and 41.4k conversation data pairs. Of the instruction data, 35.8% is generated by GPT-4o, and 60.2% of the conversation data is produced by GPT-4o. Table 2 illustrates the composition of our dataset.

5 Tuning Script

Existing fine-tuning methods in LVLM treat question tokens and image tokens equally, which neglects the need for LVLM to prioritize image information. As a result, models may understand tasks well but generate responses based solely on questions. This results in hallucinations in GUI comprehension (see Table 1). To address this, we propose a two-stage training method. The first stage aligns responses with image content. The second stage aligns responses with human intent. We also adopt the chain-of-thought method to enhance the model's reasoning capability. Our approaches are as follows:

Foundation Stage:

The Foundation Task dataset is an instruction-following dataset that fix the command and response formats. This stage incorporates direct visual information and employs a fixed format for questions and responses to train the model to correlate with image content. The fixed format is intended to "freeze" the variations in questioning and responding styles, focusing the model's learning process on understanding and interpreting visual attributes effectively. This controlled environment ensures that the model develops a robust capability to recognize and interpret visual data independently from textual content.

Advanced Comprehensive Stage:

In this stage, we introduce multi-turn dialogues and complex questions from the advanced comprehension dataset. We utilize the Referent Method (detailed in sec 3.2) to directly incorporate intuitive visual information into responses. This method is designed not only to enhance the referencing of visual details within responses but also to align the model's attention distribution more towards image tokens.It ensures that the model does not merely rely on learned text patterns but actively engages with and interprets the visual context, enabling a deeper understanding and more accurate generation

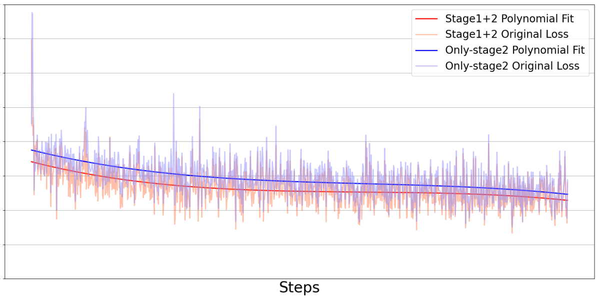


Figure 2: Loss convergence of model trained with foundation task and without during advanced task training.

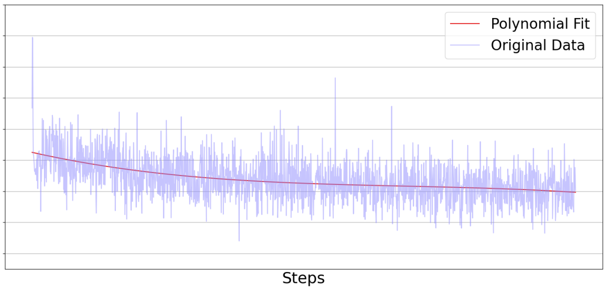


Figure 3: Loss convergence of model trained with mixed foundation task and advanced task data.

based on the actual GUI layout and visual cues.

Task Progression and Reinforcement:

We organize tasks in a sequence from simple to complex, ensuring that the model progressively builds the necessary foundational skills before training on more advanced tasks. In complex tasks, we start with some related simple tasks to reinforce foundational knowledge, followed by the real response which demands deeper analysis and synthesis of the information gathered from these foundational tasks.

Model	First	Second	Third	Average
VGA-no-referent	50.68	48.18	46.14	48.33
VGA-mix-stage	62.05	62.96	62.27	62.42
VGA-only-stage2	67.27	65.00	67.50	66.59
VGA-7b-v1	**90.68**	**90.68**	**91.17**	**90.83**

Table 3: We conducted ablation experiments on our fine-tuning methods, and the results prove that our fine-tuning methods are effective.

5.1 Experiments Setup

We use our training method and dataset to fine-tune llava-v1.6-mistral-7b (Liu et al., 2024a). Table 6 shows the hyperparameters we use during training. In Appendix A.2, we show the loss convergence behavior under various learning rates and batch sizes during training. All our experiments are carried out using one A100(80GB) machine.

Model	First	Second	Third	Average
GPT-4o	80.68	80.45	81.14	80.75
GPT-4V	81.82	81.14	82.50	81.82
MiniCPM	63.86	64.09	64.32	64.09
CogAgent	69.77	69.77	69.09	69.55
llava-next	53.41	53.18	53.86	53.48
idefics2-8b	39.77	43.86	41.81	41.82
VGA-7b-v1*	59.00	59.75	57.75	58.5
VGA-7b-v1	**90.68**	**90.68**	**91.17**	**90.83**

Table 4: Scores on GUI Bench.

6 Experiment

6.1 Baselines & Evaluation Metric

GUI comprehension Bench: To evaluate our model, we follow previous works (Masry et al., 2023; Liu et al., 2023). Due to the lack of GUI comprehension bench, we sample 22 images from the Rico dataset (excluding training data). And based on these images, we collect 44 user questions which require truly understanding of GUI to response correctly. Inspired by the evaluation method of LLaVA-bench (in-the-wild) (Liu et al., 2024b), we use ChatGPT to evaluate our model. We compare our model with five recent best performance LVLMs: the top non-open-source LVLMs GPT-4V and GPT-4o, the two best LVLMs based on Mistral-7b-instruction-v2 (llava-v1.6-mistral-7b(Liu et al., 2024a) and idefics2-8b(Laurençon et al., 2024)), and MiniCPM-llama3-V-2.5(Hu et al., 2024), the leading open-source model on the LVLM leaderboard.

Fine-tuning Method Evaluation: To evaluate our two-stage fine-tuning approach (Foundation and Advanced Comprehension method), we track loss convergence over the training period. We compare loss convergence during the advanced task training between models pre-trained with the foundation task and which without it. We also compare previous models' loss convergence with which trained on mixed foundation task and advanced task data. Their learning rates during training are all set to 2e-5, batch sizes are 16.

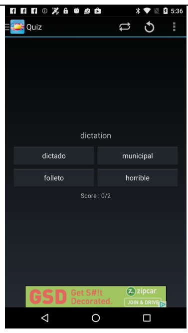
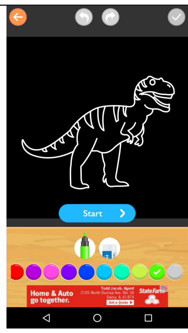
Case in bench		
		
User	How can I score?	How can I draw a picture?
VGA-7b-v1	You can score the 'dicatado' button, which is located in the middle left of the page.	You can click the blue 'Start' button in the center of the page to begin drawing a picture.
VGA-7b-v1*	You should click the 'dictado' button located in the top left of the four options.	You can start drawing by clicking the blue 'Start' button
llava-mistral-v1.6-7b	The click to score is the one that says "horrible"	To draw a picture, you can use a drawing app on your phone or tablet.
Idefics2-8B	To get the score, you should click on the "Score" button.	To draw a picture, simply tap on the colors you want to use and then touch the screen where you would like to place the color...
MiniCPM-V-2_5	To score, you click on the word .'horrible'	To draw a picture, you can use the drawing tools provided in the app interface.

Table 5: Case in the GUI-bench, note that VGA show the best comprehension of the image

6.2 Main Result

GUI Comprehension: As shown in Table 4, VGA-7b-v1 has shown promising results, achieving the best performance across three separate GPT evaluations. Our model attains a score of 90.83, which is relatively 41% better than the base model llava-v1.6-mistral-7b, 54% better than idefics2-8b and 29% better than MiniCPM-llama3-V-2.5. Furthermore, VGA-7b-v1 also outperforms GPT-4o and GPT-4V. Overall, these results establish VGA as the SOTA model for GUI comprehension.

Resource Efficiency: We also evaluate our model's performance with low-resolution input. As shown in Table 4, VGA-7b-v1* (336x336) still outperforms llava-v1.6-mistral-7b and idefics-8b, even though the number of image tokens for VGA-7b-v1* is only one-fourth of those in llava-v1.6-mistral-7b and idefics-8b. This significantly improves the inference speed, computational cost and memory usage. Our research indicates that reducing pixel count primarily affects detail recognition accuracy, thereby impacting response accuracy. Nevertheless, the model retains a strong understanding of GUI images, as demonstrated in Table 5.

Hallucination Analysis: Compared with existing models, our model demonstrates a precise understanding of GUI. As shown in Table 1 and Table 8, our model responds to questions based on the input image content and accurately describe GUI components using details such as relative position, color, and shape. By focusing more on image content, our model certainly reduces the likelihood of generating hallucinated responses.

FAC method analysis: As shown in Figure 2, the orange and purple curves represent the original loss values for models with and without the foundation task stage, respectively. The red and blue curves, fitted with a degree-3 polynomial, illustrate the loss trends. Models trained with the foundation task exhibit more stable convergence and lower loss values. Figure 3 shows that mixing datasets during training leads to unstable convergence. In Table 3, we present the results of ablation experiments on the GUI comprehension bench using the FAC fine-tuning method and those not using the FAC method. The experiment proves that the models using the FAC method have stronger GUI understanding capabilities, consistent with the comparison of the loss curves. In Appendix A.3, we discuses the influence of the foundation stage on model convergence. In Appendix A.4, we conducted ablation studies to evaluate the individual contributions of the two stages to overall model performance. In Appendix A.5, we analyzed the contribution of image tokens when generating answers to demonstrate how FAC enhances the model's attention to image content.

7 Conclusion

We introduce VGA, a fine-tuned model for GUI comprehension, leveraging a custom dataset and novel fine-tuning method. The dataset, constructed using a *Reference Method*, focuses on image-centric responses. The training can be divided into two stage: foundational skill alignment and human intent alignment. The foundation stage emphasizes learning the relationship between images and responses, while the advanced stage focuses on intent following. Ablation studies confirm the efficacy of our training method. We believe that our training data and method will serve as valuable resources for future research.

Limitations

Despite the promising results, our work has some limitations that need to be addressed in future research. Firstly, our dataset contains some noisy data, which may affect the overall performance of the models. We plan to clean this data in future iterations to improve the quality and accuracy of the dataset. Secondly, the accuracy of responses is somewhat limited by the capability of the base model we used. Although our constructed dataset and proposed methodologies have demonstrated their effectiveness, there are still instances where the answers may be inaccurate. Future work will involve experimenting with more advanced base models to further validate and enhance the effectiveness of our approach. Due to the commercial licensing restrictions of the LLaMA model, we were unable to fine-tune based on the LLaMA model. At the time of our work, mistral-7b-instruction-v3 had not yet been released. Therefore, we utilized llava-v1.6-mistral-7b. In our research, it is one of the most advanced open-source and commercially viable models available. Thirdly, our dataset is constructed based on the open-source RICO dataset, which contains GUI images with older design styles. In future dataset construction, we aim to integrate GUI interfaces from current popular apps to ensure our models are up-to-date with modern design trends and practices. By addressing these limitations, we aim to refine our approach and further improve the effectiveness and reliability of our models in comprehending and interacting with complex GUI interfaces.

Ethics Statement

During the development process, we ensured compliance with the terms and conditions of the various models and datasets we utilized. The foundation model we employed, llava-v1.6-mistral-7b, is based on the mistral-7b-instruction-v2 and CLIP models. Both llava-v1.6-mistral-7b and mistral-7b-instruction-v2 are licensed under the Apache 2.0 License, which grants open-source and commercial-use permissions from both llava and mistral AI. Our dataset was constructed based on the RICO dataset. The University of Illinois provides a permissive license for the RICO dataset under the condition that users comply with their terms, allowing for open-source and commercial use. Due to the generative nature of our models, there is a potential risk that they may be misused to generate factually incorrect responses that could misinform the public. Additionally, we cannot guarantee that our models will not produce text containing hate speech or harmful content.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments. This research was conducted when author was an intern in ByteDance, we would like to extend our gratitude to BytetDance for their assistance and support. This work was also partially supported by the NSFC under Grant No. 62272169, Shanghai Municipal Science and Technology Major Project under Grant No. 2021SHZDZX, and Shanghai Trusted Industry Internet Software Collaborative Innovation Center.

References

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

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.

Md Adnan Arefeen, Biplob Debnath, and Srimat Chakradhar. 2024. Leancontext: Cost-efficient domain-specific question answering using llms. *Natural Language Processing Journal*, 7:100065.

Yauhen Leanidavich Arnatovich and Lipo Wang. 2018. A systematic literature review of automated techniques for functional GUI testing of mobile applications. *CoRR*, abs/1812.11470.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Ishan Banerjee, Bao Nguyen, Vahid Garousi, and Atif Memon. 2013. Graphical user interface (gui) testing: Systematic mapping and repository. *Information and Software Technology*, 55(10):1679–1694.

Rui Cao, Ming Shan Hee, Adriel Kuek, Wen-Haw Chong, Roy Ka-Wei Lee, and Jing Jiang. 2023. Pro-cap: Leveraging a frozen vision-language model for hateful meme detection. MM '23, New York, NY, USA. Association for Computing Machinery.

Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. 2024. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model. *Preprint*, arXiv:2402.11684.

Jieshan Chen, Chunyang Chen, Zhenchang Xing, Xiwei Xu, Liming Zhut, Guoqiang Li, and Jinshui Wang. 2020. Unblind your apps: Predicting natural-language labels for mobile gui components by deep learning. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 322–334.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2023. Sharegpt4v: Improving large multi-modal models with better captions. *Preprint*, arXiv:2311.12793.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2023. Adapting large language models via reading comprehension. *arXiv preprint arXiv:2309.09530*.

Yung-Pin Cheng, Ching-Wei Li, and Yi-Cheng Chen. 2019. Apply computer vision in gui automation for industrial applications. *Mathematical Biosciences and Engineering*, 16(6):7526–7545.

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. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Nathan Cooper, Carlos Bernal-Cárdenas, Oscar Chaparro, Kevin Moran, and Denys Poshyvanyk. 2021. A replication package for it takes two to tango: Combining visual and textual information for detecting duplicate video-based bug reports. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 160–161.

Chenhui Cui, Tao Li, Junjie Wang, Chunyang Chen, Dave Towey, and Rubing Huang. 2024. Large language models for mobile gui text input generation: An empirical study. *arXiv preprint arXiv:2404.08948*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Preprint*, arXiv:2305.06500.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. 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*, page 845–854.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Omar El Ariss, Dianxiang Xu, Santosh Dandey, Brad Vender, Phil McClean, and Brian Slator. 2010. A systematic capture and replay strategy for testing complex gui based java applications. In *2010 Seventh International Conference on Information Technology: New Generations*, pages 1038–1043. IEEE.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30).

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. 2023. Cogagent: A visual language model for gui agents. *Preprint*, arXiv:2312.08914.

Shengding Hu, Yuge Tu, Xu Han, and Chaoqun He. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *Preprint*, arXiv:2404.06395.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. 2024. What matters when building vision-language models? *Preprint*, arXiv:2405.02246.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Preprint*, arXiv:2107.13586.

Kiwan Maeng, Alexei Colin, and Brandon Lucia. 2017. Alpaca: intermittent execution without checkpoints. *Proc. ACM Program. Lang.*, 1(OOPSLA).

Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. 2023. Unichart: A universal vision-language pretrained model for chart comprehension and reasoning. *Preprint*, arXiv:2305.14761.

Atif M. Memon, Martha E. Pollack, and Mary Lou Soffa. 1999. Using a goal-driven approach to generate test cases for guis. In *Proceedings of the 21st International Conference on Software Engineering*, page 257–266.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Ana CR Paiva, Nikolai Tillmann, João CP Faria, and Raul FAM Vidal. 2005. Modeling and testing hierarchical guis. In *Proceedings of the 12th International Workshop on Abstract State Machines*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *Preprint*, arXiv:2304.03277.

Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. 2009. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Kabir Sulaiman Said, Liming Nie, Adekunle Akinjobi Ajibode, and Xueyi Zhou. 2020. Gui testing for mobile applications: objectives, approaches and challenges. *Proceedings of the 12th Asia-Pacific Symposium on Internetware*.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Haz Sameen Shahgir, Khondker Salman Sayeed, Abhik Bhattacharjee, Wasi Uddin Ahmad, Yue Dong, and Rifat Shahriyar. 2024. Illusionvqa: A challenging optical illusion dataset for vision language models. *Preprint*, arXiv:2403.15952.

Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, et al. 2023. Mixture-of-experts meets instruction tuning: A winning combination for large language models. *arXiv preprint arXiv:2305.14705*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Mario Linares Vásquez, Kevin Moran, and Denys Poshyvanyk. 2018. Continuous, evolutionary and large-scale: A new perspective for automated mobile app testing. *CoRR*, abs/1801.06267.

Fei Wang, Kamakshi Kodur, Michael Micheletti, Shu-Wei Cheng, and Yogalakshmi Sadasivam. 2024. Large language model driven automated software application testing.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. 2023. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

Binwei Yao, Ming Jiang, Diyi Yang, and Junjie Hu. 2023. Empowering llm-based machine translation with cultural awareness. *arXiv preprint arXiv:2305.14328*.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. 2024. Ferret-ui: Grounded mobile ui understanding with multimodal llms. *Preprint*, arXiv:2404.05719.

Shengcheng Yu, Chunrong Fang, Yuchen Ling, Chentian Wu, and Zhenyu Chen. 2023a. Llm for test script generation and migration: Challenges, capabilities, and opportunities. In *2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security (QRS)*, pages 206–217.

Shengcheng Yu, Chunrong Fang, Ziyuan Tuo, Quanjun Zhang, Chunyang Chen, Zhenyu Chen, and Zhendong Su. 2023b. Vision-based mobile app gui testing: A survey. *arXiv preprint arXiv:2310.13518*.

Yichi Zhang, Jiayi Pan, Yuchen Zhou, Rui Pan, and Joyce Chai. 2023. Grounding visual illusions in language: Do vision-language models perceive illusions like humans? *Preprint*, arXiv:2311.00047.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *Preprint*, arXiv:2304.10592.

A Appendices

A.1 Existing Method For Data Generation

To effectively train an LVLM, an extensive amount of data with high-quality is required (Zhu et al., 2023). Due to the huge labor consumption to create a large high-quality dataset, many research initiatives have explored methods for automatic generation of datasets (Gilardi et al., 2023)(Chiang et al., 2023)(Peng et al., 2023).

Current mainstream method focus on distilling knowledge from advanced large language models and large visual-language models, which are pre-trained on vast amounts of data, often exceeding terabyte (TB) scales, to capture a wide variety of patterns in language and visuals (Xu et al., 2023)(Chiang et al., 2023)(Maeng et al., 2017)(Chen et al., 2024). Therefore, they are often used as auxiliary tools to generate high-quality data. For instance, ShareGPT4v (Chen et al., 2023) distill the knowledge from GPT-4V to generate data for training, ensuring the image and text pairs are of high-quality and diversity. LLaVA (Liu et al., 2023) turns pictures into text descriptions and then uses GPT-4 (OpenAI, 2024) to generate questions and answers based on text descriptions. Both of them get a state-of-the-art performance.

A.2 Hyperparameter Analysis

Our fine-tuning process is based on the llava-v1.6-mistral-7b model. In this section, we examine the influence of different learning rates and batch sizes on model convergence across two stages.

Figures 8 and Figure 9 display the loss convergence during foundation task training at various learning rates, both of the batch sizes are 16. It is evident that a learning rate of 2e-6 achieves lower loss values and more stable learning, with significantly fewer fluctuations.

Figures 10 and Figure 13 show the loss convergence during advanced task training of models pre-trained with different learning rates on foundation task. The results indicate that model pre-trained with a learning rate of 2e-6 on foundation task results in markedly lower loss and reduced fluctuations during advanced task training.

Comparing Figure 12 and Figure 10, which both pre-trained with learning rate of 2e-6 on foundation task, and with the different batch size (32 and 16). This comparison shows that increasing the batch size significantly improves convergence stability and reduces fluctuations during advanced task, avoids over-fit in intent following.

In our experiments, the model trained under the conditions depicted in Figure 12 surpasses others. This model achieves higher accuracy in GUI element recognition compared to the which pre-trained on foundation task with a learning rate of 2e-5. and show more accuracy in human intent following compared with model trained with smaller batch size during advanced task training. This further underscores the importance of foundation task training and highlights the distinct focuses of the two stages. Using a larger batch size in advanced helps the model capture the nuances of user intent, thereby preventing over-fitting. Using a small learning rate in foundation task training helps the model more accurately map images to their corresponding responses.

A.3 Influence of Foundation Task

As shown in Figure 14, the performance during the foundation training phase significantly impacts the results of advanced task training. As shown in Figure 10 and Figure 13, if the model achieves stable convergence during the foundation task, it will also exhibit stable convergence during the advanced task. Furthermore, the final convergence value in the advanced task is influenced by the convergence value achieved during the foundation task. Using models with lower convergence values from the foundation task leads to smaller convergence values during the advanced task task training.

A.4 Ablation Study

We conduct ablation studies compare among base model (llava-v1.6-mistral-7b) and models fine-tuned by FAC method (VGA-7b-v1)and solely on advanced task. VGA-7b-v1 is fine-tuned based on model which pre-trained on foundation task, VGA-7b-stage2 is fine-tuned based on llava-v1.6-mistral-7b. As shown in Figure 7, we compare their loss convergence during advanced task and result shows that the two models exhibit highly consistent loss trends, but VGA-7b-v1 demonstrates lower loss values. This indicates that our foundation task training approach, which involves freezing the question and response formats, does not degrade the model's intent-following performance. On the contrary, this training method allows the model to focus more on image information, leading to lower loss values during the advanced task training while maintaining similar trends.

We present some case in Table 10, we evaluate this three models on real-world tasks. Compared to the baseline model llava-v1.6-mistral-7b, the responses of VGA-7b-stage2 incorporate the answer style of the reference method. Compared to VGA-7b-v1, the models that not trained on foundation task shows limitations in GUI recognition accuracy, particularly in positional accuracy. This highlights the functionality and effectiveness of the two-phase training approach.

A.5 Attention Analysis

To verify that our method increases the model's focus on extracting information from images when generating responses, we analyzed the attention values between image tokens and answer tokens. Specifically, we calculated the average attention contribution of each image token for every answer token during the generation process.. Additionally, we computed the total attention value of image tokens for each answer token during its generation.

As shown in Figure 23, we compared our model VGA-7b-v1 with the baseline model llava-v1.6-mistral-7b using the same images and questions as input, and we recorded the differences in attention value. The results show that our model has significantly higher attention values to image tokens when generating responses compared to the llava model. This indicates that our model is more capable of capturing the content of the images.

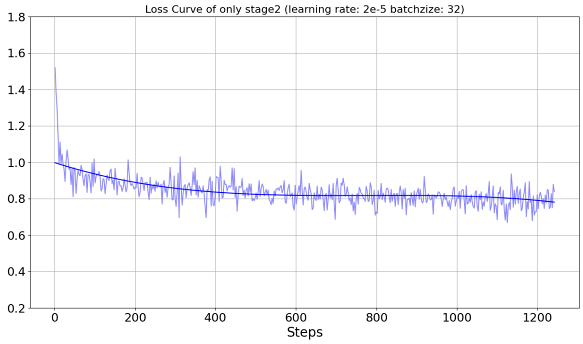


Figure 4: VGA-7b-stage2

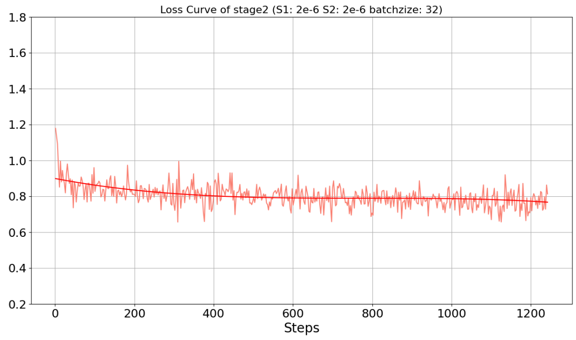


Figure 5: VGA-7b-v1 (with FAC method)

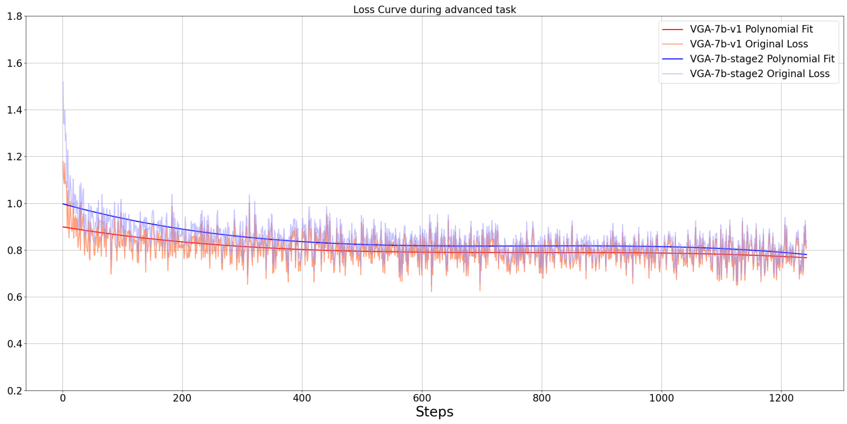


Figure 6: Loss convergence comparing between VGA-7b-v1 and VGA-7b-stage2

Figure 7

Experiment	**Data Size**	**Training Model**	**Learning Rate**	**Batch Size**	**GPUs**	**Time**
			Foundation Task Training			
Foundation Task	22.3k	Connector& LLM	2e-6	16	1xA100 80G	16h
			Advanced Comprehension Task Training			
Advanced Task	41.4k	Connector& LLM	2e-6	32	1xA100 80G	16h

Table 6: Hyperparameter in Training

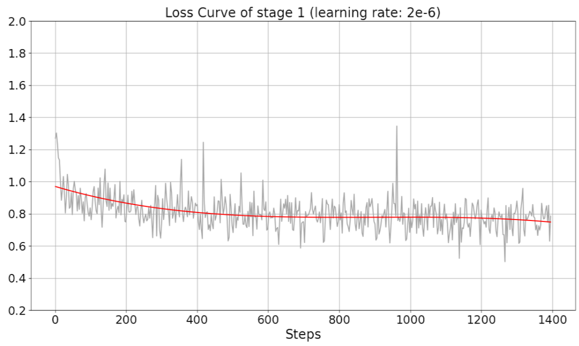


Figure 8

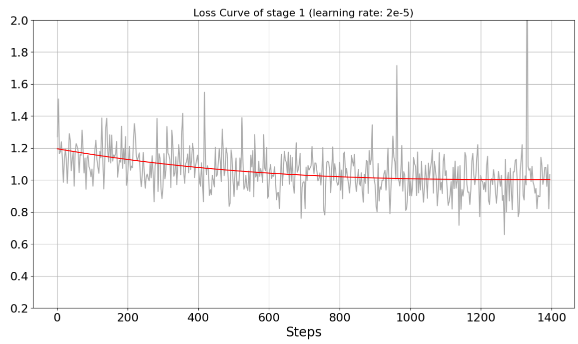


Figure 9

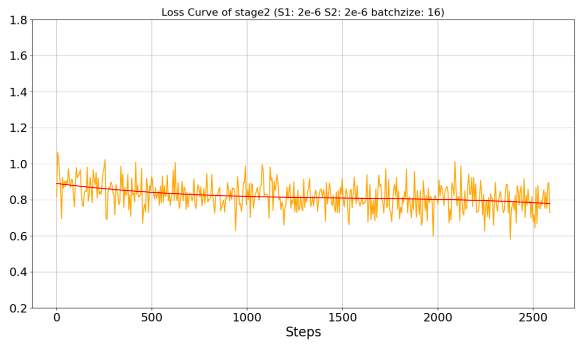


Figure 10

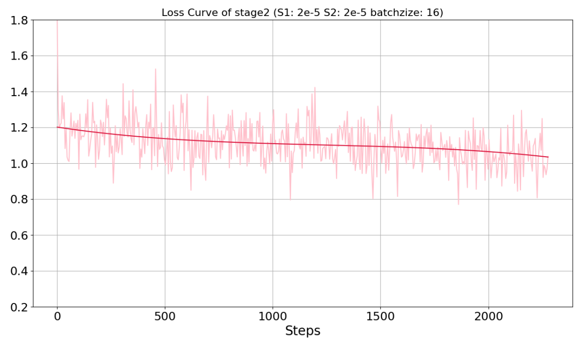


Figure 11

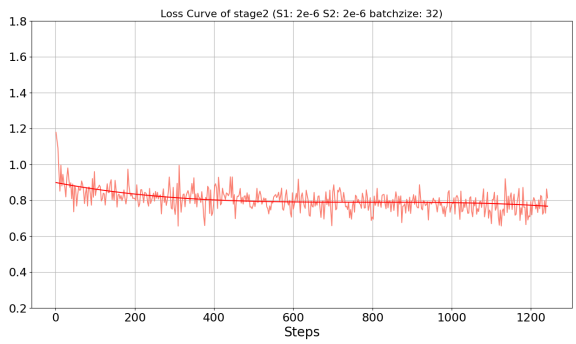


Figure 12: VGA-7b-v1

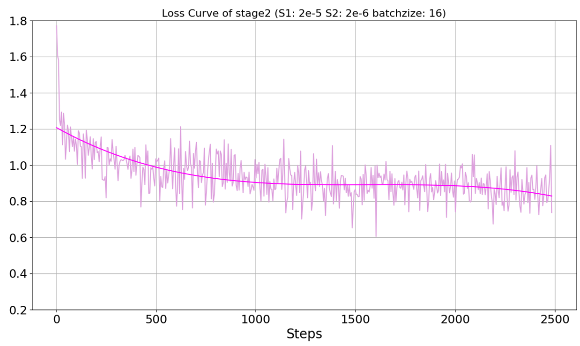


Figure 13

Figure 14: Train hyperparameter analysis

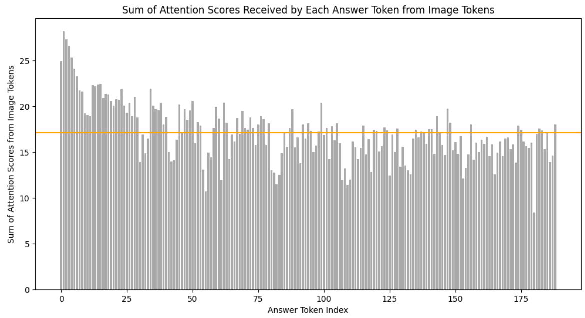


Figure 15: llava-v1.6-mistral-7b

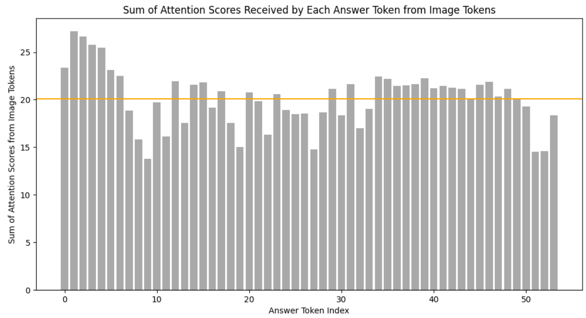


Figure 16: VGA-7b-v1

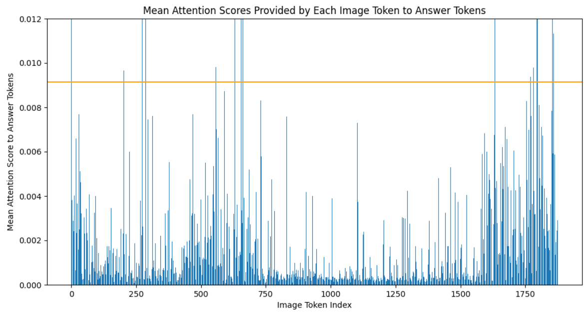


Figure 17: llava-v1.6-mistral-7b

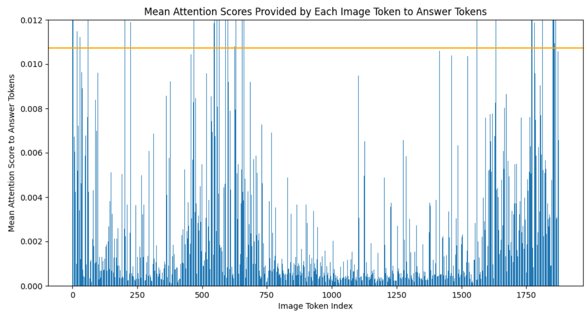


Figure 18: VGA-7b-v1

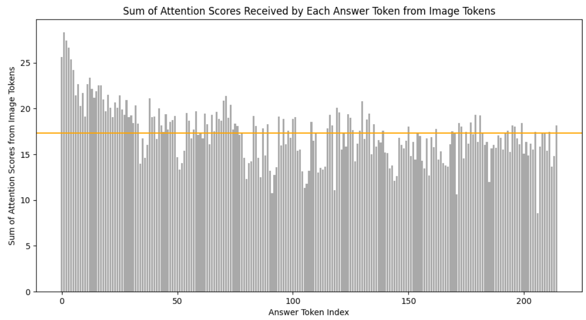


Figure 19: llava-v1.6-mistral-7b

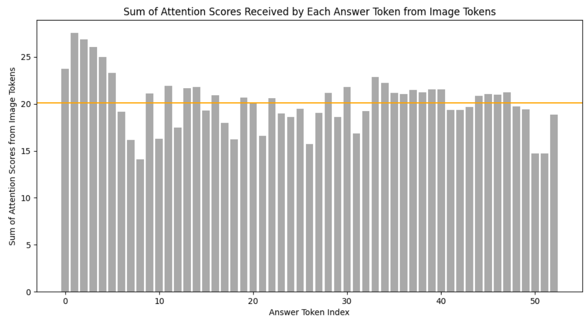


Figure 20: VGA-7b-v1

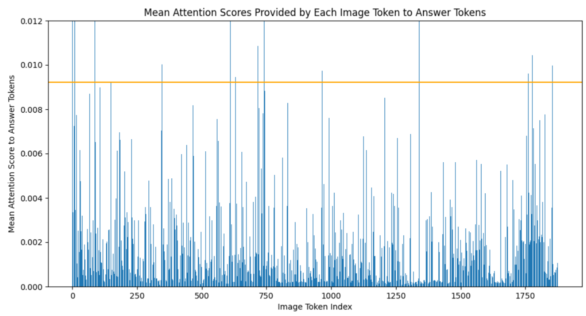


Figure 21: llava-v1.6-mistral-7b

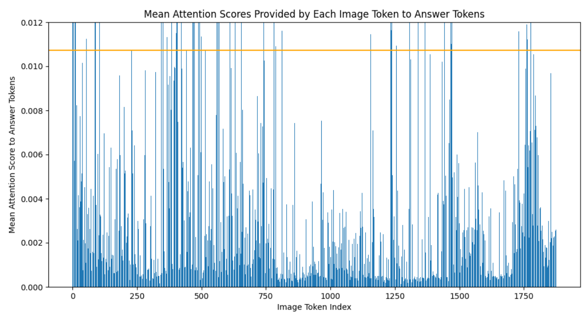


Figure 22: VGA-7b-v1

Figure 23

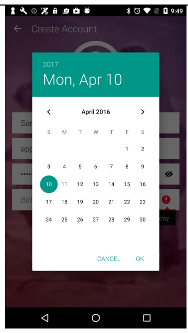
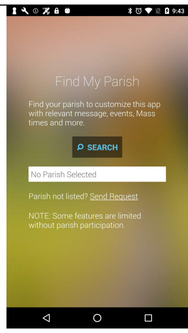
Case in bench		
		
User	What the day and date of the weekend before I selected in the middle of the page?	How can I find my Parish?
VGA-7b-v1	It is SAT, Apr 9, 2016	You can push "SEARCH" button in the middle of the page with a blue text.
llava-mistral-v1.6-7b	The image shows a calendar for April 2017, and you have selected the date April 10, which is a Monday. The weekend before April 10 is Saturday, April 8, and Sunday, April 9.	You can find your Parish by customizing this app with relevant message, events, Mass times and more.
Idefics2-8B	The day and date of the weekend before I selected are April 15th and 16th. *Year 2016 Missed.*	To find your parish, you can use the search bar provided on the app. This will allow you to input the name of your parish or search for it using other relevant information.
MiniCPM-V-2_5	The day and date of the weekend before April 10, which is a Monday, would be Sunday, April 9. *Year 2016 Missed.*	To find your Parish using the app, you can customize it with relevant messages, events, and Mass times by selecting a parish from the list provided.

Table 7: Case in the GUI-benchmark

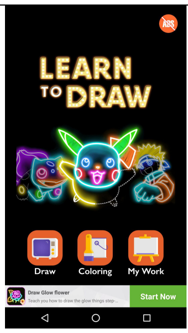
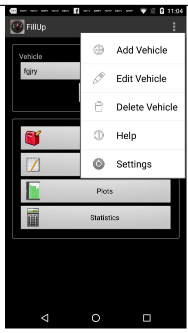
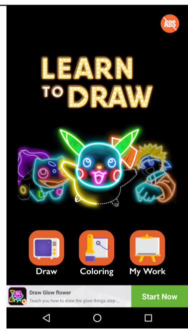
Case in bench		
		
User	How can I start Drawing?	Am I use current location right now?
VGA-7b-v1	You can click the orange "Draw" button in the bottom left corner of the page.	Yes, you are using you current location. The location icon is in the first line named "Use Current Location".
llava-mistral-v1.6-7b	To start drawing, you can use the Draw app on your device, which is available in the app store.	Yes, you are using your current location right now.
Idefics2-8B	To start drawing, you can either tap on the "Draw" button or select one of the three options provided (Coloring, My Work, or Start Now).	Yes, I am using your current location to provide information about nearby places of interest.
MiniCPM-V-2_5	You can click the "Draw" button.	No, you are not currently using your current location.

Table 8: Case in the GUI-benchmark

Case in bench

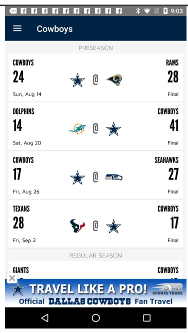


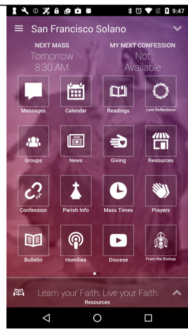


User	How can I remove the vehicle?	Where can I find the work I drew?
VGA-7b-v1	You can remove the vehicle by clicking on the 'Delete Vehicle' option in the dropdown menu at the top right of the page. It's the third option from the top.	You can find the work you've drawn by clicking on the orange square icon labeled 'My Work' at the bottom right of the page.
GPT-4o	To remove the vehicle, tap on "Delete Vehicle" from the menu shown.	You can find the paintings you drew by clicking on the "My Work" button.
GPT-4V	You can remove the vehicle by selecting the "Delete Vehicle" option in the menu shown on the app.	You can find the paintings you drew in the "My Work" section of the app shown in the image.
MiniCPM-V-2_5	To remove a vehicle from the application, you would typically select the 'Delete Vehicle' option from the menu.	The "My Work" icon suggests a section where users can view their past drawings.

Table 9: Case in the GUI-benchmark

Ablation Study

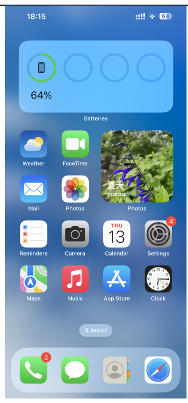


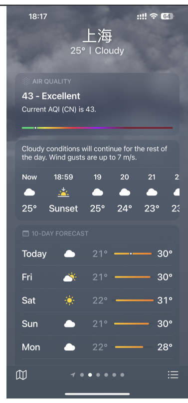


User	Who will play against each other on Sun, Aug 14?	where is the diocese?
VGA-7b-v1	The Cowboys and the Rams will play against each other on sun, aug 14.	The Diocese icon is in the bottom row, second from the left. It is a white icon with a video camera symbol.
VGA-7b-stage2	The Cowboys will play against the Dolphins on Sun, Aug 14..	The Diocese is in the second row, second column.
llava-v1.6-mistral-7b	Cowboys and dolphins.	The diocese icon is located in the bottom left corner of the app.

Table 10: Ablation study of the model trained with foundation task and without. llava-v1.6-mistral-7b is the base model, VGA-7B-stage2 is the model trained solely on advanced task.

Real world





User	Please open the weather app, click?	What is the weather on Sat?
VGA-7b-v1	<0.12, 0.28>	The weather in Sat is sunny with a temperature of 22°C.
User	Has anyone called me?	What is the highest temperature today?
VGA-7b-v1	Yes, there is a call icon with a red notification badge on it.	The highest temperature today is 30°C. You can find it in the bottom section of the page, under the '10-DAY FORECAST' section.
User	Could you help me to play some song?	what is the temperature in 19:00?
VGA-7b-v1	To play a song, tap on the icon that looks like a music note. It's located in the bottom row, second from the left.	The temperature in 19:00 is 25°C.
User	What is date of the week today?	What is the weather like now?
VGA-7b-v1	Today's date of the week is Thursday.	The current weather is 25°C and cloudy.
User	Open the camera.	How will the weather change next?
VGA-7b-v1	To open the camera, tap on the screen at <0.34, 0.7>.	The weather will continue for the rest of the day. Wind gusts are up to 7 m/s.
User	What is the setting bounds?	How about the air quality?
VGA-7b-v1	The setting bounds are <0.762, 0.566, 0.922, 0.692>.	The air quality is currently labeled as 'Excellent' with a value of 43.

Table 11: Case in the real world