LLaVA-Mob: Efficient Large Language and Vision Assistant for Mobile

Anonymous ACL submission

Abstract

Recent advancements in mobile GUI automation have leveraged multimodal large language models (MLLMs) for task automation. However, deploying these models on mobile devices poses significant challenges, including high computational costs, suboptimal performance, and limited adaptability to mobile-specific contexts. In this paper, we propose LLaVA-Mob, a lightweight multimodal agent designed for efficient smartphone GUI automation. LLaVA-Mob features a compact 1B-parameter language model and a GUI-optimized vision encoder, specifically tailored for mobile environments. Additionally, we introduce a synthetic data generation approach to produce high-quality, domain-aligned datasets, enhancing alignment between visual and textual modalities. Experiments on the AITW dataset demonstrate that LLaVA-Mob achieves performance comparable to larger models while significantly reducing computational costs, making it well-suited for resource-constrained mobile platforms. We will release our code, model, and datasets upon publication.

1 Introduction

Multimodal Large Language Models (MLLMs) have recently emerged as powerful agents capable of interacting with both real and virtual environments (Wang et al., 2023b; Zhang et al., 2023c; Yao et al., 2022; Xi et al., 2023; Li et al., 2023a). Among these, autonomous agents stand out for their ability to dynamically interact with their surroundings, creating feedback loops that influence successive states (Wang et al., 2023a; Richards, 2023; Liu et al., 2023b; Rawles et al., 2023). For practical applications such as graphical user interface (GUI) automation, these agents must combine precise perception with reliable action execution, demonstrating significant potential to manage tasks traditionally performed by humans. With multimodal capabilities, these agents can serve as robust GUI assistants, effectively perceiving and interacting with digital environments.

On resource-constrained mobile devices, achieving a balance between performance and efficiency is crucial. Most existing MLLMs face challenges that hinder their deployment in such environments, including high computational demands, complex inference, and limited adaptability to the mobile domain. These challenges can be summarized as follows: (1) Dependency on Large-Scale MLLMs: Many existing models rely on powerful, closed-source LLMs like GPT-4V (OpenAI, 2023), which require refined prompt and post-processing strategies (Richards, 2023; Shen et al., 2023; Yan et al., 2023). Such models, like Mobile-Agent, frequently call APIs for complex inference tasks, introducing privacy risks and limiting customization. By contrast, models built on open-source LLMs (e.g., LLaMA, Vicuna (Touvron et al., 2023a; Chiang et al., 2023)) offer greater flexibility and control, allowing direct training in the GUI domain while enhancing privacy through local deployment. (2) Multimodal Perception Challenges: GUI agents need robust multimodal perception to navigate complex, information-dense environments. Visual language models have shown promise in aligning visual and linguistic modalities (Dai et al., 2023; Ye et al., 2023; Zhao et al., 2023), but GUI environments involve nuanced details that general approaches fail to capture. For example, a small magnifier icon suggests a "search" function—an implicit semantic meaning that standard image captioning often misses. Recent methods use OCR and icon detectors to convert visual data into textual representations (e.g., XML layouts) (Zhang et al., 2021; Sunkara et al., 2022), but these approaches have significant limitations: (1) lengthy textual inputs slow down inference, and (2) reliance on parsed elements restricts adaptability, making them dependent on the accuracy of the parsing process.

To address these challenges, we propose LLaVA-

Mob, a model featuring a compact 1B-parameter LLM and a vision encoder pre-trained on GUI-specific tasks (Cheng et al., 2024). This architecture reduces fine-tuning and deployment costs while enhancing visual perception and action prediction for mobile environments. We also introduce a synthetic data approach that utilizes specialized models to generate high-quality, domain-aligned synthetic datasets. This improves feature alignment between visual and textual modalities, enabling more efficient and accurate action prediction.

Our contributions are summarized as follows:

- We propose LLaVA-Mob, a cognitive LLM agent tailored for GUI automation tasks. It utilizes a more lightweight model with lower training costs while achieving performance comparable to larger models.
- We introduce a new synthetic data approach that combines multiple expert models to generate high-quality synthetic datasets.
- Experiments show that our new mobile agent, built on a 1B model, achieves performance comparable to larger models on the AITW dataset.

2 Related Work

This section introduces studies on autonomous language agents and multimodal perception of LLMs.

2.1 Autonomous Language Agents

Recent work has highlighted the potential of *language agents*—language models capable of interacting with environments or other agents to solve complex tasks (Li et al., 2023a; Richards, 2023; Wu et al., 2024a). These agents either leverage large language models (LLMs) like GPT-4 for reasoning and planning through prompt engineering (Richards, 2023; Shen et al., 2023; Yan et al., 2023) or focus on trainable, open-source models for greater customization and privacy (Shao et al., 2023).

While GPT-based agents like AutoGPT and HuggingGPT showcase strong generalization abilities, they lack adaptability for specific environments. To overcome this, trainable approaches have been developed, such as m-BASH (Sun et al., 2022), which used ROI pooling for GUI tasks, Auto-UI (Zhang and Zhang, 2023), which reformulated GUI interactions into a VQA framework, and CogAgent (Hong et al., 2023), which added a high-resolution visual module with alignment pertaining. We follows the trainable approach, focusing on open-source language agents better suited for customizable and privacy-conscious applications.

2.2 Multimodal Integration in LLMs

The integration of multiple modalities with language models has become a key area of research, driven by the advancements in large language models (LLMs). Most current approaches adopt a language-centric framework, where data from other modalities is encoded into the language embedding space. These models typically consist of three components: a pre-trained encoder for the non-language modality, a language model, and an adapter (or projector) to bridge the two. Different designs of adapters have been proposed to achieve this fusion. For instance, BLIP-2 (Li et al., 2023b) employs a Q-former to generate query vectors that represent image features, while LLaVA (Liu et al., 2023a) uses a linear layer to map visual encodings from CLIP into the language space. These innovations have led to the development of various multimodal LLMs, including Flamingo (Alayrac et al., 2022), MiniGPT-4 and its v2 version (Zhu et al., 2023; Chen et al., 2023), mPLUG (Ye et al., 2023), Video-LLaMA (Zhang et al., 2023b), and SpeechGPT (Zhang et al., 2023a). By leveraging pre-trained encoders and sophisticated adapters, these models effectively align information across modalities, enabling applications that extend beyond traditional language modeling.

3 Methodology

Our approach introduces two primary innovations: (1) a lightweight model architecture optimized for mobile devices, and (2) a synthetic data approach that robustly aligns visual and textual modalities within GUI environments. Together, these advancements enhance the accuracy of GUI element perception and enable more efficient and effective command prediction tailored to mobile-specific tasks.

3.1 Model

Architecture We adapt the LLaVA framework (Liu et al., 2023a), extending it with components specifically optimized for GUI automation tasks. Our architecture integrates:

- Text Module: A lightweight Llama-3.2-1B (Dubey et al., 2024) model serves as the decoder, optimized for mobile tasks where simplicity and efficiency are prioritized.

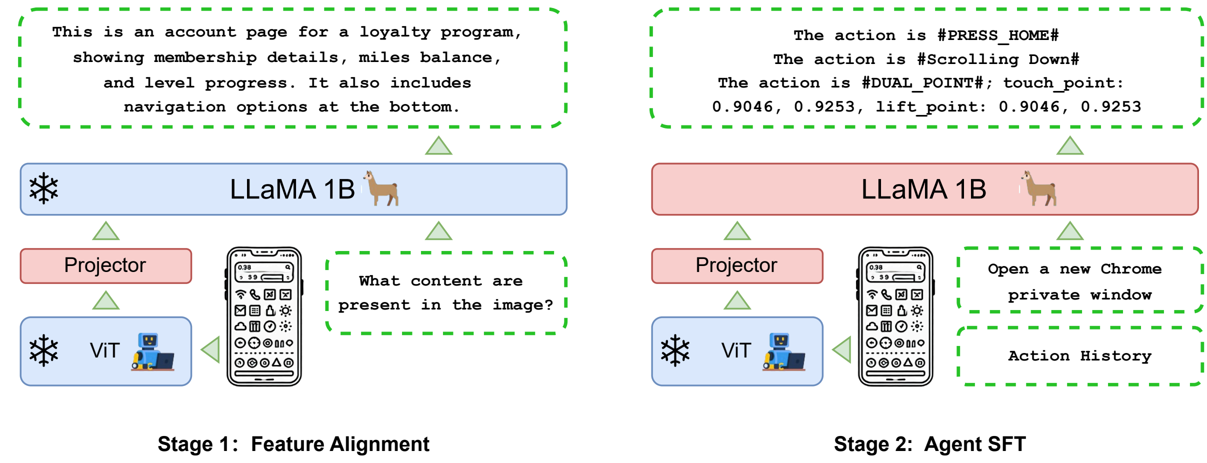

Figure 1: The architecture of LLaVA-Mob. It consists of a vision encoder with a pre-trained ViT from SeeClick (Cheng et al., 2024), two linear projection layers, and an advanced LLaMA-3.2-1B (Dubey et al., 2024) large language model.

- Vision Encoder: The SeeClick visual encoder (Cheng et al., 2024), based on a 48-layer ViT-bigG model, is pre-trained on GUI-specific data to enhance element recognition in dense GUI interfaces.

- Projection Module: A two-layer linear projection (PRJ) maps visual features to the language embedding space, ensuring effective alignment between modalities.

As shown in Figure 1, Our model architecture builds upon the LLaVA framework (Liu et al., 2023a), extending its capabilities for GUI automation. The adapted LLaVA structure in LLaVA-Mob integrates Llama-3.2-1B (Dubey et al., 2024) as the text module (DECODER), a SeeClick (Cheng et al., 2024) vision encoder (ENCODER_{image}), and a two-layer linear projection module (PRJ) to map image features to the language embedding space (EMBED_{text}). The input X consists of both text (X_{text}) and image (X_{image}), with the output represented as Y. The process begins with embedding the text and encoding the image:

$$
\begin{aligned}
H_{text} &= \text{EMBED}_{text}(\ X_{text} \circ \hat{Y}^{0:t-1}\ ), \\
Z_{image} &= \text{ENCODER}_{image}(\ X_{image}\ ), \\
H_{image} &= \text{PRJ}(\ Z_{image}\ ).
\end{aligned} \tag{1}
$$

Here, $\circ$ denotes the concatenation operation, allowing text and historical action outputs to be embedded together. The two-layer linear projection module PRJ is defined as:

$$
H_{image} = W_2\,\text{ReLU}\,(W_1 Z_{image} + b_1) + b_2,
$$

where W_1, W_2, b_1, and b_2 are learnable weights and biases, and ReLU is the activation function used between the two linear layers. The text module interprets instructions, while the vision encoder processes GUI screenshots to extract relevant visual features. The projection module bridges the visual and textual modalities, enhancing multimodal understanding and improving accuracy in command predictions for mobile-specific tasks.

This adapted architecture is specifically optimized for mobile GUI automation challenges, allowing LLaVA-Mob to maintain efficiency and achieve precise action prediction, despite the resource constraints typical of mobile devices.

Visual Encoder The core focus of mobile agent tasks is the visual encoder's ability to locate elements within GUI interfaces, especially when relying solely on screenshots. To address the challenge of accurate GUI element recognition, SeeClick (Cheng et al., 2024) introduced a GUI grounding pre-training strategy. This strategy involves automated data collection from diverse web and mobile sources, such as web layouts, mobile widget descriptions, and UI summaries, enabling the model to generalize across different GUI environments. Following the setup in SeeClick, which initializes from the visual encoder of Qwen-VL (Bai et al., 2023), we directly adopt this visual encoder—a 48-layer ViT-bigG (Ilharco et al., 2021) pre-trained on GUI grounding tasks—allowing LLaVA-Mob to leverage its robust ability to interpret visual information accurately.

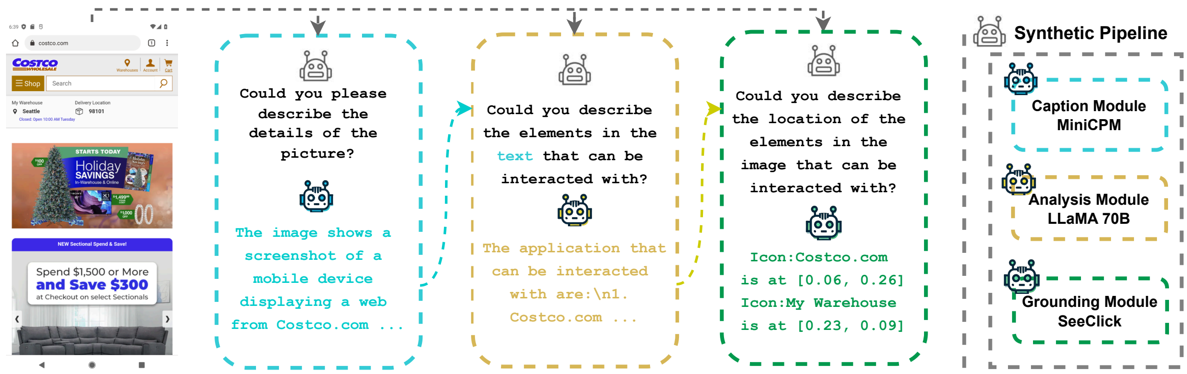

Figure 2: The workflow of our synthetic data approach: The Caption Module performs image captioning to generate descriptive summaries of the GUI. The Analysis Module provides textual elements within the GUI to extract meaningful insights and context. The Grounding Module identifies interactive elements such as buttons, icons, and links while determining their precise locations for interaction.

Small Large Language Model To optimize large language model deployment on mobile devices, balancing performance and efficiency, we selected Llama 3.2 1B (Dubey et al., 2024) as the new text decoder. Mobile tasks don't require the same complexity in language fluency and diversity as tasks like reading comprehension or dialogue. Instead, the priority is to understand task requirements within a fixed instruction format, make accurate judgments, and locate key features effectively. Therefore, a simpler text decoder is sufficient for mobile agents. Given the limited computing resources on mobile devices, tightly controlling the model's parameter size is also crucial for successful on-device deployment.

Training As shown in Figure 1, following the LLaVA settings, the training is divided into two stages. In the first stage, alignment data is used to align the representations between the visual encoder and the text decoder. During this stage, only the projector layers are trained. In the second stage, the agent is trained through visual instruction tuning using action prediction data, and this stage involves full fine-tuning of the text decoder.

3.2 Data

We train our model using a combination of established datasets including AITW and AMEX, and a newly introduced, GUI-focused synthetic dataset, specifically designed for alignment augmentation. Together, these resources span a range of complementary tasks, including action prediction, element grounding, and screen description, providing a robust foundation for comprehensive model training.

- AITW Dataset (Rawles et al., 2023): Comprising 1 million samples, AITW covers an extensive array of GUI-action prediction scenarios. Tasks include mobile-specific commands such as opening applications, typing text, and performing scrolling actions.

- AMEX Dataset (Chai et al., 2024): AMEX prioritizes detailed screen descriptions, functionality explanations, and element grounding tasks. It includes 30k screen description samples, 199k element grounding samples, and 280k functionality descriptions.

- Synthetic Dataset: Designed explicitly for GUI environments and derived from AITW images using our synthetic data approach, this dataset enriches the training process through automated data generation.

VLMs like LLaVA (Liu et al., 2023a) follow a two-stage training process, with the first stage aligning representations between two pre-trained models on different modalities. While this process has been extensively studied in general domains, creating high-quality alignment data for mobile platforms remains a challenge. Initially, we used 500k VQA samples from the AMEX (Chai et al., 2024) dataset for alignment. However, the use of visual information in this data is very limited, the descriptions of image content are not detailed enough, and there is a lack of correspondence between image elements and location information. Moreover, this data involves local descriptions of coordinate positions rather than performing grounding tasks. Additionally, our analysis shows that

55.2% of AITW dataset involves DUAL_POINT tasks, which require regression of coordinate data. Therefore, high-quality grounding data becomes even more crucial for such tasks. To address this, we develop a synthetic data approach to leverage existing models and build a robust pipeline for generating high-quality alignment data through synthetic data construction.

Synthetic Data Approach Our synthetic data approach consists of three modules, each performing a specific step to extract and refine information, as shown in Figure 2. First, MiniCPM-V-2.5 (Yao et al., 2024), with strong perceptual capabilities, generates detailed image descriptions and effectively captures ICON information due to its understanding of GUI elements. Second, LLaMA2-70B (Touvron et al., 2023b), known for its strong reasoning abilities, analyzes on these descriptions to extract interactive ICON elements from the text. Finally, SeeClick (Cheng et al., 2024), which specializes in grounding tasks, maps the ICON elements extracted by LLaMA2-70B (Touvron et al., 2023b) to their corresponding locations.

For the synthetic data, we randomly selected 8,000 images from the AITW dataset to create a 24k image-text dataset tailored for mobile platforms. The dataset includes detailed image descriptions, element descriptions, and precise location annotations, with 8,000 samples in each category. This 24k dataset was used for the first stage of training on LLaVA-Mob, enhancing the alignment of visual and textual representations for mobile-specific tasks. Unlike AMEX data, our Caption section generates a small paragraph of text rather than a simple sentence. Also, for the ICON position information, we give the coordinates of the content, contrary to AMEX.

4 Experiments

Our implementation builds on LLaVA (Liu et al., 2023a), incorporating the LLaMA-3.2-1B (Dubey et al., 2024) model and the SeeClick (Cheng et al., 2024) vision encoder (Cheng et al., 2024). First, we validated the model structure described in Section 3.1 by conducting fair comparisons of different vision encoders. This was done by keeping Stage 1 training on the AMEX500K (Chai et al., 2024) dataset and Stage 2 training and evaluation using the instruction format from the Auto-UI (Zhang and Zhang, 2023) version of the AITW (Rawles et al., 2023) data. After finalizing the vision module and model structure, as mentioned in Section 3.2, we enhanced model alignment by performing ablation experiments on alignment data, with Stage 2 settings remaining consistent.

Hyperparameter	AMEX	Synthetic	Synthetic	AiTW
Training Stage	1	1	1	2
Data Size	500K	24K	163K	1000K
Learning Rate	1e-3	1e-3	1e-3	2e-5
Epoch	1	3	3	3
Training Time	8	2	8	150
Batch Size	64			
Optimizer	AdamW			
Lr Schedule	cosine decay			
Lr Warmup Ratio	0.03			

Table 1: LLaVA-Mob's hyperparameters differ across training stages and datasets. The training time is measured in hours on a single A100 GPU.

4.1 Implementation

In stage 1 of aligning the vision encoder and language encoder, we respectively used the AMEX (Chai et al., 2024) data and the synthetic data. AMEX (Chai et al., 2024) is a comprehensive benchmark for Android OS GUI, containing over 104K high-resolution screenshots and 711K element-wise functionalities under real-world app contexts. We converted the AMEX (Chai et al., 2024) data into a VQA format suitable for instruction understanding to embed GUI-specific knowledge in the MLLM. AMEX (Chai et al., 2024) designed four distinct VQA tasks, three of which—Screen Description, Element Grounding, and Functionality Description—detail image features and related GUI elements, making them ideal for aligning the visual and text encoders. Therefore, we packaged these three tasks into a 500K dataset for the first-phase alignment training. Additionally, as shown in Tabel 2, we also processed different versions of synthetic data into VQA format for use in the first-phase alignment training.

In stage 2, only AiTW data is used, following the Auto-UI (Zhang and Zhang, 2023) settings. As shown in Figure 3, AiTW (Rawles et al., 2023) is a benchmark for smartphone GUI, containing 715K operation episodes under 30K reality intentions. Each entry includes a goal in natural language, screenshots, and actions. Humans collect data on various devices and operation systems in various screen resolutions. According to the applications domain, AITW consists of five subsets: General, Install, GoogleApps, Single, and WebShopping. This dataset, referencing LLaVA (Liu

Tasks	Source	Quantity	Examples of Task Templates
GUI-Action Prediction	AITW	1000k	`User`: Goal: open app Google Play Music `Agent`: Action Decision: action type: PRESS HOME, touch point: [-1.0, -1.0], lift point: [-1.0, -1.0], typed text: "".
Screen Description	AMEX	30k	`User`: Provide a one-sentence caption for the provided GUI image. `Agent`: This page is a film section of the IMDb app, displaying a list of movies or shows with sorting and filtering options.
Element Grounding	AMEX	199k	`User`: Identify all clickable elements and provide their 2D **BBox**. `Agent`: List of Elements with **BBox**es.
Functionality Description	AMEX	280k	`User`: What is the function of the icon at the coordinates: **BBox**. `Agent`: Click to expand or collapse the sort options.
Caption	Synthetic data	163k	`User`: Could you please describe the details of the picture? `Agent`: The image shows a screenshot of an online shopping platform, specifically a product listing for a MacBook Air ...
Analysis	Synthetic data	8k	`User`: Could you describe the elements in the image that can be interacted with? `Agent`: The image shows interactive elements within the Best Buy app interface, allowing users to engage with various features ...
Grounding	Synthetic data	8k	`User`: Could you please describe the location of the elements in the image that can be interacted with? `Agent`: Icon: Costco.com is at [0.06, 0.26]; Icon: My Warehouse is at [0.23, 0.09]; Icon: Delivery Location is at [0.23, 0.5]...

Table 2: Task template examples. The AMEX and synthetic data were used in the first phase of training, while the AITW data was used in the second phase.

et al., 2023a), was processed into the format of visual instruction tuning, totaling 1,000k instructions. All entries were used in the second phase of training to help the model understand instruction generation tasks. Leveraging the experience from Auto-UI (Zhang and Zhang, 2023), our text data is based on versions that incorporate historical instructions. Inspired by the latest research (Ma et al., 2024; Cheng et al., 2024), we have made appropriate adjustments to the instructions. For detailed information, please refer to the table 6 in the appendix.

4.2 Setup

Training We implemented four versions of alignment training: one using only the AMEX dataset and the other using a different version synthetic dataset. As shown in Tabel 1 .The AMEX version was trained on 500K samples for 1 epoch, while the synthetic data versions, with only 24K samples and 163k samples, were trained for 3 epochs. The performance differences between these two alignment strategies are analyzed in detail in our ablation experiments. Meanwhile, for both stages, we follow LLaVA's settings, using AdamW as the optimizer, a cosine decay learning rate schedule, and a warmup ratio of 0.03. The learning rate is set to 1e-3 for alignment and 2e-5 for fine-tuning, with a consistent batch size of 64 and 3 training epochs. DeepSpeed Stage 3 is applied throughout to enhance training efficiency.

Evaluation In our experiments on AITW subsets, we primarily trained on the entire dataset in a unified manner. Accuracy, measured at each time step across all parameters, serves as our main metric. Refactored actions are parsed into JSON format, with each parameter compared to the action label, following (Rawles et al., 2023). A predicted coordinate is considered correct if it falls within the labeled element's bounding box or within 7% of the screen distance from the labeled point. A scroll action is considered correct if its main direction is accurate. For other parameters, exact matches are required, except for *typed text* or dialogue responses. In AITW, typed text is correct if the label appears in the predicted text.

4.3 Baselines

For AITW, we compare our proposed approach with several baselines. Uni-modal API-based methods, such as those by Rawles et al. (2023) and Zhang and Zhang (2023), evaluate 5-shot performance on PaLM-2 (Anil et al., 2023) and ChatGPT(Ouyang et al., 2022), using pseudo-HTML

Model	Params	Overall	General	Install	GoogleApps	Single	WebShop.
ChatGPT-COT (Ding, 2024)	-	7.72	5.93	4.38	10.47	9.39	8.42
GPT-4V ZS+HTML (Ding, 2024)	-	50.54	41.66	42.64	49.82	72.83	45.73
GPT-4V ZS+History (Ding, 2024)	-	52.96	43.01	46.14	49.18	78.29	48.18
GPT-4o (Wu et al., 2024b)	-	55.02	47.06	49.12	52.30	80.28	46.42
MobileAgent (Wang et al., 2024a)	-	66.92	55.8	74.98	63.95	76.27	63.61
InternVL +History (Wu et al., 2024b)	6B	2.63	1.95	2.88	2.94	3.03	2.71
Qwen-VL +History (Wu et al., 2024b)	7B	3.23	2.71	4.11	4.02	3.89	2.58
PaLM-2 (Zhang and Zhang, 2023)	340B	39.6	–	–	–	–	–
MM-Navigator (Yan et al., 2023)	-	50.54	41.66	42.64	49.82	72.83	45.73
MM-Navigator$_{\text{w/ text}}$ (Yan et al., 2023)	-	51.92	42.44	49.18	48.26	76.34	43.35
MM-Navigator$_{\text{w/ history}}$ (Yan et al., 2023)	-	52.96	43.01	46.14	49.18	78.29	48.18
OmniParser (Wan et al., 2024)	-	50.54	41.66	42.64	49.82	72.83	45.73
BC (Rawles et al., 2023)	1B	68.7	–	–	–	–	–
BC $_{\text{w/ history}}$ (Rawles et al., 2023)	1B	73.1	63.7	77.5	75.7	80.3	68.5
Qwen-2-VL (Wang et al., 2024b)	2B	67.20	61.40	71.80	62.60	73.70	66.70
Show-UI (Qinghong Lin et al., 2024)	2B	70.00	63.90	72.50	69.70	77.50	66.60
Llama 2 (Zhang and Zhang, 2023)	7B	28.40	28.56	35.18	30.99	27.35	19.92
Llama 2+Plan+Hist (Zhang and Zhang, 2023)	7B	62.86	53.77	69.1	61.19	73.51	56.74
Auto-UI (Zhang and Zhang, 2023)	5B	74.27	68.24	76.89	71.37	84.58	70.26
MobileVLM (Wu et al., 2024b)	7B	74.94	69.58	79.87	74.72	81.24	71.70
SphAgent (Chai et al., 2024)	7B	76.28	68.20	80.50	73.30	85.40	74.00
CoCo-LLAVA (Ma et al., 2024)	7B	70.37	58.93	72.41	70.81	83.73	65.98
SeeClick (Cheng et al., 2024)	9.6B	76.20	67.60	79.60	75.90	84.60	73.10
CogAgent (Hong et al., 2023)	18B	76.88	65.38	78.86	74.95	93.49	71.73
LLaVA-Mob	1B	77.52	71.61	80.01	75.45	87.15	73.41

Table 3: Results on AITW: Action accuracy across main setups, highlighting overall performance in decision-making tasks. # means, CoCo-Agent relies on layout data to retrieve icon positions, making it not directly comparable to other end-to-end methods that do not depend on API or system-level data. However, we include this result for reference.

code to represent images and predicting action targets by item names or indices without verifying coordinates. Multimodal methods include MM-Navigator (Yan et al., 2023), a GPT-4V-based agent achieving few-shot state-of-the-art. Training-based methods feature models like Behavioral Cloning (Rawles et al., 2023), a Transformer-based agent with BERT (Devlin et al., 2019), LLaMA-2 for uni-modal tasks with pseudo HTML inputs (Zhang and Zhang, 2023), and Auto-UI (Zhang and Zhang, 2023), a multimodal encoder-decoder with T5 and BLIP. Finally, CogAgent (Hong et al., 2023), a 9B-parameter visual LLM with a high-resolution cross module, excels in GUI understanding and achieves top performance on AITW. OmniPhrser (Wan et al., 2024) employs OCR for text extraction and Blip2 for improved multimodal comprehension.

4.4 Main Results

Table 3 presents action accuracy across primary setups, including various task subsets such as overall performance, general tasks, installation tasks, Google Apps, single-action tasks, and web shopping. Notably, LLaVA-Mob demonstrates exceptional efficiency, achieving an overall accuracy of 77.52 percent with only 1 billion parameters. It performs particularly well in the General and Single task subsets, with accuracies of 71.61 percent and 87.15 percent, highlighting its robustness across diverse scenarios. Despite its smaller size, LLaVA-Mob approaches the performance of larger models like SphAgent (Chai et al., 2024) and LLaVA (Ma et al., 2024) and surpasses many in efficiency. Unlike models such as MobileAgent (Wang et al., 2024a) and CogAgent (Hong et al., 2023), which benefits from additional data and long memory, LLaVA-Mob relies solely on end-to-end data to achieve an excellent balance between performance and resource efficiency. This makes it an ideal choice for mobile applications and resource-constrained environments. Its strong performance across all subsets underscores its effectiveness and efficiency in handling GUI-related perception and decision-making tasks.

4.5 Ablation Study

Our ablation study evaluated the contributions of different components of the model, focusing on Pre-Training Vision Encoder and Synthetic Data. All ablation experiments were trained with origin format of AITW dataset and tested on General data with accuracy metric.

Model	Layers	Resolution	Pretrain Task	General
ViT-large	24	336	CLIP	61.51
ViT-bigG	48	224	CLIP	62.85
SeeClick	48	224	Grounding	64.51

Table 4: Comparison of vision encoders within the same structure on action accuracy, using AMEX 500K as Stage 1 data and the Origin format of AITW as Stage 2 data.

Pre-Training we conduct an ablation study on the visual decoder, comparing model performance initialized with bigG and SeeClick. As shown in table 4, comparing the first and second lines, the performance of the model can be further improved by choosing a more powerful visual encoder. Meanwhile, SeeClick, pre-trained on large-scale GUI data, significantly enhances adaptation to GUI action prediction task.

Data	Size	Cost/$	Epoch	Train/h	General
AMEX	500K	0	1	8	64.51
Caption	24K	0	3	2	66.32
Caption	163K	0	3	7	66.99
Mixing	8K+8K+8K	15	3	2	67.25

Table 5: Comparison of alignment datasets in Stage 1 within the same structure using SeeClick as the vision encoder, with the Origin format of AITW as Stage 2 data. The cost reflects the use of LLaMA2-70B through an API, resulting in incurred expenses.

Synthetic Data Table 5 demonstrates the effectiveness of the Synthetic dataset in improving model performance. Despite having significantly fewer samples than AMEX (Chai et al., 2024) , both 24k and 164k caption data can outperform AMEX (Chai et al., 2024) , achieving higher accuracy on General action prediction task. Given that the caption data in the synthetic dataset is much longer and more detailed than the brief content summaries in the AMEX dataset, this demonstrates that in alignment tasks, richer detailed descriptions lead to better alignment outcomes and data quality. The comparison between the third and fourth rows emphasizes that data quality is more important than data size for alignment tasks. The synthetic pipeline's ability to capture detailed ICON information has greatly enhanced data quality. This demonstrates the importance of high-quality, domain-specific data for alignment, with the synthetic pipeline achieving strong and efficient results, even with smaller sample sizes.

5 Conclusion

In this paper, we introduced LLaVA-Mob, a compact and efficient multimodal large language model tailored for smartphone GUI automation tasks. By addressing the unique challenges of mobile environments, LLaVA-Mob demonstrates how lightweight architectures can effectively balance performance and computational efficiency.

Our approach features two main innovations: a specialized model architecture leveraging a 1B-parameter language model and a pre-trained vision encoder optimized for GUI tasks, and a synthetic data generation strategy to enhance visual-textual alignment through high-quality domain-specific datasets. These advancements ensure LLaVA-Mob delivers robust performance while maintaining low resource requirements, making it suitable for deployment on mobile devices.

The experimental results validate the efficacy of our approach, with LLaVA-Mob achieving competitive accuracy compared to larger models on the AITW benchmark, highlighting its ability to manage diverse GUI-related tasks effectively. This work underscores the potential of lightweight MLLMs to serve as practical, scalable solutions for mobile automation, bridging the gap between resource constraints and advanced functionality.

6 Future Work

GUI agents based on instruction fine-tuning only perform basic representation transfer, narrowing the prediction action space within the entire instruction generation task. While still far from real-world application, they serve as cost-effective base models. Recent studies have explored combining reinforcement learning strategies, such as Proximal Policy Optimization (Schulman et al., 2017), with MLLMs, with significant efforts made in recent works Digirl (Bai et al., 2024) and RL4VLM (Zhai et al., 2024). Future research should focus on integrating instruction fine-tuned models with reinforcement learning to build GUI automation agents that can be deployed in real-world environments. Further exploration is needed to develop mobile-friendly reinforcement learning environments that better adapt to MLLMs.

Limitations

Detailed ablation studies across multiple sub-tasks can highlight the differences between methods more effectively. However, due to the extensive size of the AITW test set, conducting these tests is very time-consuming, with some tasks taking over 20 hours. As a result, ablation experiments were only performed on the General task. Future research should focus on acquiring standardized test subsets to speed up inference and testing, which would help optimize further explorations in this area.

References

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*, abs/2305.10403.

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

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 1(2):3.

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

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechu Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *ArXiv preprint*, abs/2310.09478.

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

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.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tinghe Ding. 2024. Mobileagent: enhancing mobile control via human-machine interaction and sop integration. *arXiv preprint arXiv:2401.04124*.

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

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

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. Openclip. If you use this software, please cite it as below.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. Camel: Communicative agents for" mind" exploration of large scale language model society. *ArXiv preprint*, abs/2303.17760.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv preprint*, abs/2301.12597.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023b. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv: 2308.03688*.

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

OpenAI. 2023. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. 2024. Showui: One vision-language-action model for gui visual agent. *arXiv e-prints*, pages arXiv–2411.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy P Lillicrap. 2023. Androidinthewild: A large-scale dataset for android device control. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Toran Bruce Richards. 2023. Auto-gpt: An autonomous gpt-4 experiment. https://github.com/Significant-Gravitas/Auto-GPT.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-LLM: A trainable agent for role-playing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13153–13187, Singapore. Association for Computational Linguistics.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *ArXiv preprint*, abs/2303.17580.

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 *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6699–6712, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Srinivas Sunkara, Maria Wang, Lijuan Liu, Gilles Baechler, Yu-Chung Hsiao, Jindong Chen, Abhanshu Sharma, and James W. W. Stout. 2022. Towards better semantic understanding of mobile interfaces. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5636–5650, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Jianqiang Wan, Sibo Song, Wenwen Yu, Yuliang Liu, Wenqing Cheng, Fei Huang, Xiang Bai, Cong Yao, and Zhibo Yang. 2024. Omniparser: A unified framework for text spotting key information extraction and table recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15641–15653.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. Voyager: An open-ended embodied agent with large language models. *ArXiv preprint*, abs/2305.16291.

Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024a. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. *arXiv preprint arXiv:2401.16158*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2023b. A survey on large language model based autonomous agents. *ArXiv preprint*, abs/2308.11432.

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

Biao Wu, Yanda Li, Meng Fang, Zirui Song, Zhiwei Zhang, Yunchao Wei, and Ling Chen. 2024a. Foundations and recent trends in multimodal mobile agents: A survey. *arXiv preprint arXiv:2411.02006*.

Qinzhuo Wu, Weikai Xu, Wei Liu, Tao Tan, Jianfeng Liu, Ang Li, Jian Luan, Bin Wang, and Shuo Shang. 2024b. Mobilevlm: A vision-language model for better intra-and inter-ui understanding. *arXiv preprint arXiv:2409.14818*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *ArXiv preprint*, abs/2309.07864.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing reasoning and acting in language models. volume abs/2210.03629.

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

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chaoya Jiang, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qi Qian, Ji Zhang, and Fei Huang. 2023. mplug-owl: Modularization empowers large language models with multimodality.

Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. 2024. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *arXiv preprint arXiv:2405.10292*.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023a. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities.

Hang Zhang, Xin Li, and Lidong Bing. 2023b. Video-llama: An instruction-tuned audio-visual language model for video understanding. *ArXiv preprint*, abs/2306.02858.

Xiaoyi Zhang, Lilian de Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, et al. 2021. Screen recognition: Creating accessibility metadata for mobile applications from pixels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Zhuosheng Zhang, Yao Yao, Aston Zhang, Xiangru Tang, Xinbei Ma, Zhiwei He, Yiming Wang, Mark Gerstein, Rui Wang, Gongshen Liu, and Hai Zhao. 2023c. Igniting language intelligence: The hitchhiker's guide from chain-of-thought reasoning to language agents.

Zhuosheng Zhang and Aston Zhang. 2023. You only look at screens: Multimodal chain-of-action agents. *ArXiv preprint*, abs/2309.11436.

Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. 2023. Mmicl: Empowering vision-language model with multi-modal in-context learning. *ArXiv preprint*, abs/2309.07915.

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

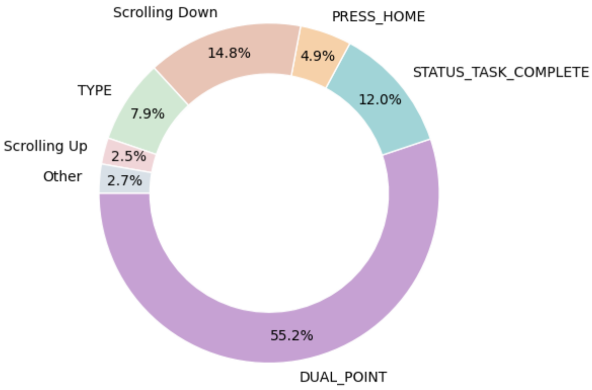

Figure 3: Distribution of Task Types in AiTW dataset: This chart shows the frequency distribution of different task types across the entire training dataset, consisting of approximately 1 million data points.

Origin Instruction Template	**InsCom Middle Template**
Action Decision: action type: PRESS_HOME, touch point: [-1.0, -1.0], lift point: [-1.0, -1.0], typed text: "".	The action is <PRESS_HOME>.
Action Decision: action type: PRESS_BACK, touch point: [-1.0, -1.0], lift point: [-1.0, -1.0], typed text: "".	The action is <PRESS_BACK>.
Action Decision: action type: PRESS_ENTER, touch point: [-1.0, -1.0], lift point: [-1.0, -1.0], typed text: "".	The action is <PRESS_ENTER>.
Action Decision: action type: STATUS_TASK_COMPLETE, touch point: [-1.0, -1.0], lift point: [-1.0, -1.0], typed text: "".	The action is <STATUS_TASK_COMPLETE>.
Action Decision: action type: TYPE, touch point: [-1.0, -1.0], lift point: [-1.0, -1.0], typed text: "{string}".	The action is <TYPE>, "typed_text": "{string}".
Action Decision: action type: Scrolling_Up, touch point: [0.8, 0.5], lift point: [0.2, 0.5], typed text: "".	The action is <Scrolling_Up>.
Action Decision: action type: Scrolling_Down, touch point: [0.2, 0.5], lift point: [08, 0.5], typed text: "".	The action is <Scrolling_Down>.
Action Decision: action type: DUAL_POINT, touch point: {coordinate}, lift point: {coordinate}, typed text: "".	The action is <DUAL_POINT>, "touch_point": "{coordinate}", "lift_point": "{coordinate}".

Table 6: Examples of transformations between origin data format and Our formats for all task types.