UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Enhancing Interaction Accuracy, Efficiency, and Robustness in Multimodal Large Language Models

by Yanda Li

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Prof. Ling Chen and Dr. Hua Zuo

University of Technology Sydney
Faculty of Engineering and Information Technology

January 2025

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Yanda Li, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computer Science, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note: Signature removed prior to publication.

Yanda Li

DATE: 23rd December, 2024

PLACE: Sydney, Australia

Abstract

Recent advances in artificial intelligence have been significantly driven by large language models (LLMs) and vision-language models (VLMs), which have demonstrated remarkable performance in language reasoning and perceptual understanding, respectively. However, the growing need for unified multimodal reasoning has led to the emergence of multimodal large language models (MLLMs). The rapid evolution of MLLMs has revolutionized the integration of vision, language and audio, enabling transformative advancements in intelligent systems. However, their real-world deployment remains constrained by challenges in interaction accuracy, computational efficiency, and resilience to noisy or adversarial inputs. This thesis systematically addresses these limitations through an in-depth exploration of three pivotal dimensions.

First, the thesis introduces a robust multimodal instruction-tuned model built upon a novel image-dialogue generation pipeline. This pipeline synthesizes high-quality, instruction-aligned image-text pairs using multi-stage prompting and model filtering, effectively addressing the lack of scalable multimodal instruction data. Leveraging this synthetic training data, the resulting model achieves state-of-the-art performance across multiple benchmarks, demonstrating strong instruction-following capability, spatial reasoning, and resistance to hallucination.

Second, this thesis proposes a lightweight agent framework for multimodal reasoning and task execution on resource-constrained mobile devices. Designed for environments with limited compute and memory, the framework integrates memory-driven reasoning, OCR-based visual parsing, and retrieval-augmented planning to enable dynamic decision-making across multiple applications. It supports efficient and robust execution of complex and multi-step tasks, such as long-horizon workflows and interactions across different apps, without relying on cloud-based inference or additional retraining. Experiments show that the proposed system consistently outperforms existing mobile agent baselines in task success rate, demonstrating strong adaptability and deployment potential in real-world settings.

Finally, the thesis presents a comprehensive benchmark for evaluating the robustness of

large audio-language models under adversarial and noisy conditions. The benchmark includes over 1200 adversarial examples across four categories: content distortion, emotional interference, explicit noise, and implicit noise. It supports evaluation using standard metrics, LLM-as-a-judge, and human assessments. Experiments show that current audio-language models remain vulnerable to adversarial audio inputs, revealing persistent weaknesses in robustness. This benchmark serves as a foundation for analyzing reliability in voice-based language systems and informs future research on building more stable and trustworthy audio interactions.

This research makes significant contributions by advancing the precision, adaptability, and resilience of multimodal systems. Experimental results validate the proposed methodologies, demonstrating their effectiveness across various domains. These findings provide a robust foundation for deploying MLLMs in dynamic environments, paving the way for future advancements in multimodal interaction technologies.

Acknowledgments

First and foremost, I want to express my heartfelt gratitude to my supervisors, Professor Yunchao Wei and Professor Ling Chen. Their support and encouragement, especially during difficult times, were essential in helping me complete my doctoral studies. I am also deeply grateful to Professor Yi Yang, whose guidance and insights throughout my Ph.D. journey have enriched both my academic and personal growth.

I would also like to extend my deepest appreciation to my girlfriend, Wanqi Yang, who traveled from China to Australia to be by my side during my studies. Her quiet companionship over the years has been a source of strength, as we have grown together and left traces of our love across continents. I hope we can continue supporting each other as we work toward our dreams.

I am equally thankful to my parents, Zhixiong Li and Lixia Zhou, whose unwavering love over the past 30 years has been the foundation of my growth and resilience. Their boundless support has allowed me to pursue my dreams wholeheartedly.

I am also immensely grateful to my classmates, friends, and collaborators, whose academic support and personal kindness made my Ph.D. journey fulfilling. Special thanks go to BJTU Wei Lab, ReLER Lab, UTS NLP Group, and every organization where I have interned, including Tencent, ByteDance, SenseTime, and Momenta. The encouragement and inspiration from my peers and colleagues in these environments have been invaluable in driving my growth.

Meanwhile, I want to thank myself. In the first three years of my Ph.D., I faced numerous rejections, which made the journey feel like an endless struggle. I once longed for breakthroughs in my research but found myself repeatedly challenged. Thankfully, in my final year, my work was finally recognized, allowing me to achieve a personal breakthrough. I am grateful for my own resilience and for not giving up.

Lastly, I extend my deepest gratitude to my country for providing the scholarship that enabled me to pursue my dreams. I love China, and I wish for its continued prosperity and strength.

Yanda Li
Sydney, Australia
December, 2024

PUBLICATIONS

1. **Yanda Li**, Zilong Huang, Gang Yu, Ling Chen, Yunchao Wei, and Jianbo Jiao. 2024. Disentangled Pre-training for Image Matting. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 169–178. (**CORE A Oral**)

2. **Yanda Li**, Chi Zhang, Gang Yu, Wanqi Yang, Zhibin Wang, Bin Fu, Guosheng Lin, Chunhua Shen, Ling Chen, and Yunchao Wei. 2024. Enhanced Visual Instruction Tuning with Synthesized Image-Dialogue Data. Findings of the Association for Computational Linguistics ACL 2024, pages 14512–14531. (**CORE A***)

3. Wanqi Yang*, **Yanda Li***, Meng Fang, and Ling Chen. 2024. Enhancing Temporal Sensitivity and Reasoning for Time-Sensitive Question Answering. Findings of the Association for Computational Linguistics: EMNLP 2024, pages 14495–14508. (**CORE A***)

4. **Yanda Li**, Chi Zhang, Wanqi Yang, Bin Fu, Pei Cheng, Xin Chen, Ling Chen, and Yunchao Wei. Appagent v2: Adaptive Mobile Agent for Dynamic Interactions. IEEE International Conference on Multimedia & Expo 2025 (**CORE A**)

5. Wanqi Yang*, **Yanda Li***, Meng Fang, Ling Chen. MTPChat: A Multimodal Time-Aware Persona Dataset for Conversational Agents. Findings of the Association for Computational Linguistics: NAACL 2025, pages 5830–5841. (**CORE A**)

6. Wanqi Yang*, **Yanda Li***, Meng Fang, Yunchao Wei, Ling Chen. Who Can Withstand Chat-Audio Attacks? An Evaluation Benchmark for Large-Audio Language Models. Findings of the Association for Computational Linguistics ACL 2025 (**CORE A***)

7. Chi Zhang*, Zhao Yang*, Jiaxuan Liu*, **Yanda Li***, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal Agents as Smartphone Users. In Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25). Association for Computing Machinery, New York, NY, USA, Article 70, 1–20.(**CORE A***)

8. Wanqi Yang, Yunqiu Xu, **Yanda Li**, Kunze Wang, Binbin Huang, and Ling Chen. 2024. Continual learning for temporal-sensitive question answering. In 2024 International Joint Conference on Neural Networks (IJCNN), pages 1–9. IEEE. (**CORE B**)

9. **Yanda Li**, Zilong Huang, Jianbo Jiao, Gang Yu, Tao Chen, Guozhong Luo, Ling Chen, Yunchao Wei. Trimap-Guided Attention: Improve Image Matting with Dense Foreground and Background Modeling. (Under Review)

10. Biao Wu*, **Yanda Li***, Meng Fang, Zirui Song, Zhiwei Zhang, Yunchao Wei, Ling Chen. Foundations and Recent Trends in Multimodal Mobile Agents: A Survey. (Under Review)

11. **Yanda Li**, Jianbo Jiao, Zilong Huang, Wang Bin, Humphrey Shi, Gang Yu, Yi Yang, Yunchao Wei. High-Resolution Human Parsing via Belief-to-Uncertainty Propagation. (Under Review)

Abbreviations

AI Artificial Intelligence

API Application Programming Interface

ASR Automatic Speech Recognition

COT Chain-of-Thoughts

CNN Convolutional Neural Network

CV Computer Vision

GUI Graphical User Interface

HCI Human-Computer Interaction

IT Instruction Tuning

LLMs Large Language Models

LMs Language Models

MLLM Multimodal Large Language Model

NLP Natural Language Processing

OCR Optical Character Recognition

PLMs Pre-trained Language Models

QA Question Answering

RAG Retrieval-Augmented Generation

SOTA State of the Art

SSL Self-Supervised Learning

TSQA Temporal-Sensitive Question Answering

VLM Vision Language Model

VQA Visual Question Answering

CONTENTS

List of Figures

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

The rapid advancement of artificial intelligence (AI) has been significantly driven by large language models (LLMs), which have revolutionized natural language processing (NLP) through their powerful zero-shot and few-shot reasoning capabilities. These models excel in various text-based tasks without requiring extensive task-specific training. However, traditional LLMs are inherently limited to processing textual data, restricting their applicability in multimodal contexts.

In parallel, vision-language models (VLMs) have achieved significant breakthroughs in perceptual tasks, such as image classification and cross-modal retrieval, by effectively capturing relationships between visual and textual data. Yet, their perceptual strengths are not complemented by the sophisticated reasoning capabilities inherent in LLMs. This disconnect between reasoning and perception has created a gap in addressing complex multimodal challenges, necessitating the development of Multimodal Large Language Models (MLLMs). By integrating the reasoning power of LLMs with the perceptual richness of VLMs, MLLMs offer a cohesive framework to bridge this divide and enable advanced multimodal intelligence.

A Multimodal Large Language Model is an advanced extension of large language models designed to process, integrate, and generate information across multiple modalities, including text, images, video, and audio. By leveraging unified representations and state-of-the-art training paradigms, MLLMs enable seamless semantic alignment, sophisticated reasoning,

and cross-modal generation. These capabilities position MLLMs as a cornerstone in multimodal AI research and applications, bridging the gap between perception and reasoning while unlocking new possibilities for diverse tasks and domains.

To achieve this, MLLMs must overcome two fundamental challenges. First, raw multimodal inputs such as images, audio, and video require effective preprocessing to ensure compatibility with language-based reasoning. Second, the inherent limitation of large language models to accept only textual inputs necessitates a bridging mechanism to translate non-textual modalities into formats the LLM can process. To address these challenges, a general MLLM architecture typically consists of three main components: modality encoders, a pretrained LLM, and a multimodal connector as Fig. 1.1. Together, these components enable the model to effectively utilize the strengths of various modalities within a unified framework.

Modality Encoders Modality encoders transform raw multimodal inputs (e.g., images, videos, and audio) into compact feature representations, which serve as the foundation for downstream processing. For instance, pretrained models such as vision transformers (ViTs) are often employed for images, while convolutional neural networks (CNNs) are used for audio. These encoders ensure efficient and accurate feature extraction, facilitating seamless interaction with the LLM.

Pretrained Large Language Model (LLM) The pretrained LLM serves as the core of the system, equipped with a vast knowledge base and strong reasoning capabilities through extensive pretraining on textual data. It plays a pivotal role in integrating and interpreting information from different modalities. Fine-tuning on multimodal-specific tasks further enhances the LLM's ability to handle complex multimodal challenges, making it a versatile component for diverse applications.

Multimodal Integration Interface Since the LLM inherently accepts only textual inputs, the multimodal integration interface bridges the gap by converting modality features into textual representations while maintaining semantic consistency across modalities. This interface enables the LLM to process and reason about multimodal data within a unified semantic space, ensuring alignment and coherence during cross-modal interactions.

This architecture empowers MLLMs to leverage the unique strengths of various modalities, making them widely applicable across a range of tasks. Notable examples include image captioning and generation, video understanding, and audio-driven conversational AI.

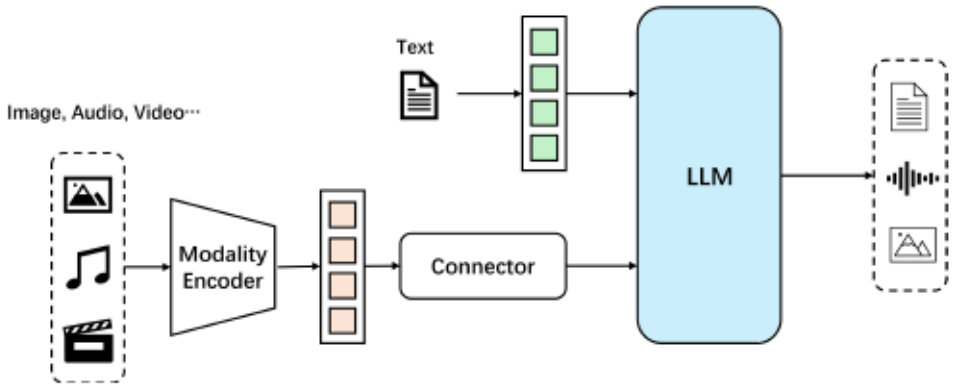

Figure 1.1: General multimodal large language model framework.

Furthermore, MLLMs enhance reasoning robustness by combining perceptual richness with advanced generative and interpretative capabilities, facilitating applications such as multimodal dialogue systems, adaptive learning environments, and real-world decision-making scenarios.

The development of MLLMs has progressed through a series of foundational advancements. Early models like CLIP [1] and ALIGN [2] demonstrated robust cross-modal representations through joint training on image-text pairs, enabling tasks such as zero-shot classification and multimodal retrieval. Flamingo [3] introduced dynamic memory mechanisms for contextual reasoning in multimodal conversational AI, while GPT-4 [4] and Gemini [5] expanded multimodal reasoning and dialogue capabilities. More recently, models such as Blip-2 [6] and LLaVA [7] have utilized instruction tuning to align image, text, and other modalities, unlocking high-fidelity understanding and generation.

The advancements in MLLMs transcend static retrieval tasks, unlocking dynamic applications in content generation, interactive systems, and autonomous agents. For example, multimodal dialogue systems powered by MLLMs can process diverse inputs—visual scenes, spoken commands, and textual queries—and generate coherent, contextually relevant responses. These advancements herald a new era of human-computer interaction, characterized by enhanced accuracy, efficiency, and robustness across diverse environments.

By integrating multimodal inputs into a unified reasoning framework, MLLMs exhibit enhanced robustness and adaptability, allowing reliable performance across a broad range of scenarios. This transformative ability not only bridges the gap between VLMs and LLMs but also establishes MLLMs as the foundation for future advancements in AI. Potential applications include adaptive learning systems, cross-modal knowledge retrieval, and real-world decision-making, demonstrating the immense potential of MLLMs to redefine the boundaries of artificial intelligence.

Despite significant advancements, multimodal large language models face critical challenges that constrain their performance in terms of interaction accuracy, efficiency, and robustness. **Interaction accuracy** is fundamentally influenced by the quality and diversity of multimodal datasets, which form the foundation of cross-modal learning. Insufficient data quality or domain coverage hinders precise alignment across modalities, limiting the model's ability to generate coherent and accurate outputs. **Efficiency** becomes a pivotal concern during model deployment, as resource constraints such as limited computational power and memory in environments like mobile devices challenge the practicality of MLLMs. At the same time, **robustness** remains an unresolved issue in dynamic or adversarial contexts, where models must perform reliably under noisy inputs, domain shifts, or adversarial attacks. These interconnected challenges must be addressed to unlock the full potential of MLLMs in real-world applications.

These challenges are highly relevant to the practical use of MLLMs in real-world scenarios. Multimodal models are increasingly applied in sensitive domains such as healthcare and education, where inaccurate cross-modal reasoning and question answering can lead to misleading or even harmful outcomes. At the same time, MLLMs are being deployed in resource-constrained scenarios such as mobile devices and robotics, where their high computational and inference demands pose significant barriers to practical adoption. Moreover, in widely used intelligent voice assistants, insufficient robustness can result in misinterpretation of user inputs, ultimately degrading the user experience. Therefore, improving the accuracy, efficiency, and robustness of MLLMs is crucial for ensuring their safe, scalable, and effective deployment in real-world applications.

Label: 837 County Road - Photo 7

Label: bbc-ice-cream-08-ss-april-release-2

Figure 1.2: Examples from the LAION-400M dataset showcasing low resolution and mismatched labels.

A significant challenge lies in the impact of dataset quality and domain gaps on the

performance of multimodal large language models (MLLMs). First, large-scale multimodal datasets such as LAION [8] and CC [9], primarily constructed through web scraping, provide millions of samples for pretraining. However, as illustrated in Figure 1.2, these datasets often suffer from low resolution and mismatched labels. Furthermore, the lack of robust quality filtering results in noisy and inconsistent image-text pairs, which significantly hampers model training and accuracy. Second, although specialized multimodal datasets such as Visual Question Answering (VQA) [10] and Visual Dialog [11] are of higher quality, their domain-specific nature limits the generalization ability of models trained on them to cross-domain tasks. Recent efforts, such as instruction tuning datasets [12, 13, 14], aim to leverage the strengths of large language models like GPT-4 to provide enriched multimodal reasoning. However, these approaches are primarily based on existing data sets and do not address fundamental domain gaps, which limits their effectiveness in real-world applications. The combination of dataset noise, domain specificity, and limited diversity continues to constrain the ability of MLLMs to achieve precise cross-modal alignment and coherent reasoning.

The efficient adaptation of MLLMs to new tasks and dynamic environments presents a significant challenge. Fine-tuning-based methods, such as those employed in LLaVA [7] and Qwen-VL [15], refine models for specific tasks but require substantial computational resources and rely heavily on large-scale, domain-specific datasets. These methods also inherit biases from the training data, reducing the adaptability of MLLMs to unseen tasks or rapidly changing scenarios. Moreover, the computational overhead of fine-tuning makes these approaches less viable for deployment in resource-constrained environments, such as mobile devices.

Although general-purpose models like GPT-4 [4] offer flexibility by eliminating the need for task-specific fine-tuning, they struggle with domain-specific challenges. For example, in mobile application scenarios, these models often struggle to interpret custom user interfaces, unique workflows, and domain-specific logic. As illustrated in Figure 1.3, when faced with operating unfamiliar applications, such as turning the camera towards people, GPT-4 makes incorrect decisions. This inefficiency highlights the need for improved adaptation strategies that balance the computational demands of fine-tuning with the need for task-specific performance.

The robustness of MLLMs remains an open challenge, particularly in handling adversarial and noisy conditions for audio inputs. While adversarial attacks have been extensively studied in image and text domains [16, 17, 18], similar research in audio-based tasks remains limited. This gap is critical because audio introduces unique challenges due to its

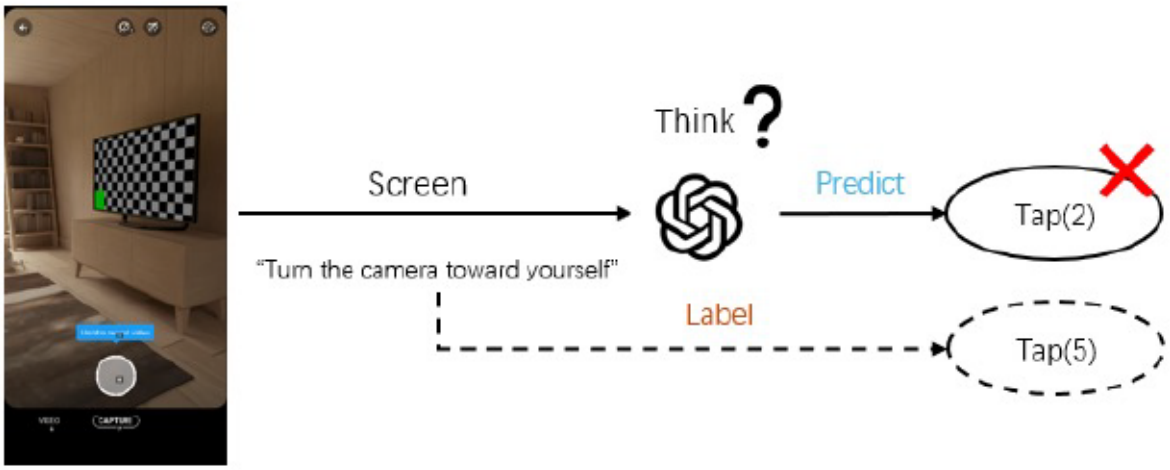

Figure 1.3: An example of a GPT-4-based Mobile Agent struggling with tasks in unfamiliar applications.

representation as waveforms or spectrograms, which complicates the alignment of adversarial perturbations with the original signal. As illustrated in Figure 1.4, qualitative results demonstrate how various audio LLMs handle original audio fused with ultrasound inputs, highlighting their susceptibility to such conditions. Current audio-enhanced MLLMs, such as GPT-4o and Qwen2-audio [19], predominantly focus on transcription tasks or audio-based question answering but lack robustness under adversarial or noisy conditions.

Implicit Noise Attack (Ultrasound)
SpeechGPT: 5 is not a power of 10 because 10 is a prime number and 5 is not a prime number.
SALMONN: It just doesn't matter how I feel.
Qwen2-Audio: The speaker's emotions are not influencing their decision or actions in any way.
LLama-Omni: Yes, it's normal to feel that way...
Gemini-1.5-Pro: Please share what the speaker said so I can give you a relevant response.
GPT-4o: I apologize, but I'm unable to assist with identifying speakers from a voice sample.

Figure 1.4: Qualitative results of various audio LLMs using original audio fused with ultrasound as input.

Additionally, many existing audio datasets [20, 21] rely heavily on structured, transcribed text, emphasizing accuracy metrics rather than stability across diverse or unpredictable environments. This limitation restricts the ability of MLLMs to generalize beyond controlled scenarios, further highlighting the need for systematic evaluation frameworks that target robustness in real-world settings.

This thesis systematically investigates the critical challenges confronting multimodal large language models (MLLMs), focusing on interaction accuracy, efficiency, and robust-

ness. It analyzes the impact of dataset quality and domain gaps on cross-modal alignment, emphasizing the need for diverse and high-quality datasets to improve interaction accuracy. The thesis further examines the limitations of existing fine-tuning approaches and explores the adaptability of general-purpose models in dynamic environments, highlighting the trade-offs between computational efficiency and task-specific performance. Additionally, it delves into the underexplored domain of robustness, particularly for audio-based MLLMs, by evaluating vulnerabilities to adversarial and noisy inputs. Through addressing these interconnected challenges, this research advances the development of MLLMs and establishes a foundation for creating more reliable, adaptable, and effective multimodal systems across diverse real-world applications.

1.2 Research Objectives and Contributions

This thesis aims to advance the interaction capabilities of multimodal large language models (MLLMs) by addressing three core challenges: accuracy, efficiency, and robustness. To this end, the research is guided by the following objectives:

Objective 1: Improve the accuracy of MLLMs by enhancing cross-modal alignment and addressing the limitations of existing training data. This includes developing high-quality synthetic datasets and alignment strategies to reduce semantic inconsistencies and noisy supervision in multimodal training.

Objective 2: Design an efficient and adaptable agent framework suitable for dynamic and resource-constrained environments, such as mobile platforms. The objective is to enable adaptive task completion without relying on repeated model retraining, by leveraging modular architectures and flexible action space.

Objective 3: Strengthen the robustness of MLLMs against adversarial and noisy inputs, particularly in audio-based interactions, by constructing comprehensive evaluation benchmarks and analyzing model vulnerabilities.

Based on these objectives, this thesis makes the following key contributions:

It proposes a data generation pipeline designed to construct high-quality, customized image-text pairs. These pairs are tailored for instruction tuning and fine-grained alignment, addressing the limitations of existing noisy or mismatched datasets. The proposed pipeline enables more accurate visual grounding and semantic reasoning, contributing to improved cross-modal understanding in MLLMs.

It develops a two-stage multimodal agent framework for mobile devices, incorporating flexible action space and RAG mechanisms to support adaptive execution in dynamic appli-

cation scenarios. The framework significantly reduces deployment costs and demonstrates superior performance in both user study and benchmark evaluations.

It introduces a novel evaluation benchmark to assess the robustness of MLLMs under various adversarial and noisy audio attack settings. This benchmark fills a critical gap in current MLLM evaluation practices and enables in-depth analysis of model reliability in speech-oriented applications.

1.3 Research Questions

This thesis addresses critical challenges in enhancing the interaction capabilities of multimodal large language models. It explores solutions to improve accuracy, efficiency, and robustness in tasks spanning diverse modalities, structured around the following three research questions:

- **Research Question 1: How can the accuracy of multimodal large language models be improved by addressing challenges in cross-modal alignment and dataset quality?**

  The interaction capabilities of multimodal large language models rely heavily on precise alignment between modalities. However, existing datasets often suffer from noisy labels, semantic inconsistencies, and mismatched pairs, which compromise the models' ability to perform accurate cross-modal reasoning. These challenges not only hinder generalization to unseen scenarios but also lead to degraded performance in tasks requiring detailed semantic understanding, such as visual question answering or image-dialogue generation. This research question explores strategies for enhancing cross-modal alignment through high-quality dataset creation and optimization techniques. By addressing these issues, this work aims to improve the accuracy and reliability of multimodal models across a wide range of applications.

- **Research Question 2: How can the efficiency of multimodal large language models be optimized to meet the demands of resource-constrained and dynamic environments?**

  Multimodal large language models face significant challenges in adapting to dynamic and resource-constrained environments, such as mobile platforms or embedded systems. Models like LLaVA, after fine-tuning on agent-specific tasks, often exhibit limitations in adapting to new tasks or domains without additional retraining. Similarly, GPT-4, despite its general-purpose reasoning capabilities, struggles to accurately

interpret custom interfaces, novel controls, or application-specific logic, which are crucial for completing complex workflows in customized applications. This research question explores strategies to optimize computational efficiency and memory usage while enabling models to dynamically adapt to diverse tasks. By leveraging feedback-driven architectures and modular frameworks, this research aims to reduce reliance on retraining, improve adaptability, and ensure efficient task execution in dynamic, real-world scenarios.

- **Research Question 3: How can the robustness of multimodal large language models be enhanced to handle adversarial and noisy inputs effectively?**

  Audio, as a key modality, enriches multimodal systems by providing contextual and temporal information. However, its susceptibility to adversarial attacks and noisy environments poses significant challenges to the reliability of multimodal large language models. For instance, subtle adversarial noise embedded in speech commands can cause critical misinterpretations, such as reversing user intent in smart home systems. Similarly, noisy environments, including overlapping speech or background interference, can severely degrade model performance. This research question seeks to identify vulnerabilities in audio-augmented multimodal systems, design robust evaluation benchmarks, and develop strategies to strengthen resilience against adversarial and noisy inputs. By addressing these challenges, the research aims to ensure consistent and reliable model performance across diverse real-world scenarios.

1.4 Thesis Organization

This thesis is structured to address the core research questions through a systematic exploration of improving interaction accuracy, efficiency, and robustness in multimodal large language models. The organization of the thesis is as follows:

- **Chapter 1: Introduction**
  This chapter provides the motivation for this research, introduces the challenges faced by MLLMs, and outlines the research questions, key contributions, and scope of the thesis.

- **Chapter 2: Literature Review**
  This chapter reviews prior work on multimodal systems, intelligent agents, and audio

LLMs. It highlights limitations in existing approaches related to cross-modal alignment, efficiency in dynamic environments, and robustness against adversarial attacks, establishing the context for the research questions.

- **Chapter 3: Enhanced Visual Instruction Tuning with Synthesized Image-Dialogue Data**
  This chapter addresses the first research question, introduces a method to mitigate domain-specific biases and noisy training data in multimodal datasets. By fine-tuning with synthesized and curated image-dialogue data, the approach improves cross-modal alignment and interaction accuracy across diverse scenarios.

- **Chapter 4: Advanced agent for flexible mobile interactions**
  In response to the second research question, this chapter presents a flexible agent framework designed for efficient task execution in resource-constrained and dynamic environments. The proposed method integrates memory-driven architectures and task adaptability without extensive retraining, thereby enhancing interaction efficiency.

- **Chapter 5: Who Can Withstand Chat-Audio Attacks? An Evaluation Benchmark for Large Language Models**
  Addressing the third research question, this chapter investigates the robustness of MLLMs to adversarial and noisy audio inputs. It introduces a benchmark for systematically evaluating the stability of audio-augmented LLMs, identifies critical vulnerabilities, and explores strategies to enhance their resilience.

- **Chapter 6: Conclusion and Future Work**
  This chapter summarizes the key findings and contributions of the thesis. It discusses potential extensions to improve the scalability and robustness of MLLMs, address ethical considerations, and broaden their application across more complex and dynamic domains.

CHAPTER 2

LITERATURE REVIEW

2.1 Multimodal Learning

2.1.1 Overview of Multimodal Learning

Multimodal learning integrates information from diverse data modalities, such as text, images, audio, and video, to enhance understanding and decision-making across complex tasks [22, 23]. By leveraging the complementary strengths of different modalities, multimodal learning enables systems to address the heterogeneity inherent in real-world data. This includes capturing richer feature representations, improving task generalization, and enhancing robustness against noisy or incomplete inputs [1, 24, 25].

Early approaches in multimodal learning primarily relied on feature concatenation, where modality-specific embeddings were merged into a unified representation [23, 26]. While these methods provided a straightforward solution, they often struggled with differences in modality-specific scales, temporal structures, and noise. Simple concatenation schemes were insufficient to model interactions across temporal or hierarchical data structures [27]. This limitation paved the way for neural-based approaches, including early Boltzmann machines and canonical correlation methods [27, 28].

The advent of deep learning introduced architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which enabled automatic learning of hierarchical feature representations [29, 30, 31]. These advancements were later augmented by attention mechanisms and transformer-based models, which provided unprecedented

capabilities for modeling cross-modal relationships [32, 33, 34]. For instance, transformer-based models like CLIP and ALIGN employed contrastive learning to align vision and text modalities, achieving remarkable performance in zero-shot classification and retrieval tasks [1, 2, 35].

Modern multimodal learning has benefited significantly from large-scale pretraining. Datasets like LAION-400M [8] and CC12M [9] have enabled the development of general-purpose vision-language models capable of generalization across domains [8, 9]. Instruction tuning, an approach inspired by natural language processing, has further extended the flexibility of multimodal models, enabling them to generalize across diverse downstream tasks [36, 13, 12].

2.1.2 Multimodal Representations

Multimodal representations form the foundation of multimodal learning, aiming to encode heterogeneous data into a unified semantic space to enable seamless reasoning across modalities [22, 24, 37]. Unlike unimodal approaches, which operate on isolated modalities, multimodal representations must capture not only modality-specific features but also the complex dependencies and interactions between modalities [1, 2, 25].

The evolution of multimodal representations has moved from shallow models to deep neural networks capable of learning semantic-rich embeddings. Early methods used statistical models like canonical correlation analysis (CCA) for cross-modal alignment, but these approaches struggled with nonlinear relationships and scalability [23, 28]. Deep learning enabled automatic extraction of high-dimensional features, with convolutional neural networks (CNNs) and recurrent neural networks (RNNs) being widely applied to tasks such as image captioning and video summarization [30, 38].

Attention mechanisms further advanced multimodal representation learning by allowing models to selectively focus on relevant features, leading to frameworks like Show, Attend, and Tell [38] and Transformer architectures [32]. These advances laid the groundwork for transformer-based multimodal models like CLIP and ALIGN, which aligned vision and text representations in shared embedding spaces [1, 2]. Similarly, VisualBERT and VL-BERT introduced joint encoding of multimodal data for reasoning-intensive tasks [39, 34].

Recent efforts emphasize pretraining multimodal representations using large-scale datasets with task-agnostic objectives. For instance, Flamingo and BLIP leverage image-text pairs to learn embeddings that generalize across diverse tasks [3, 40]. In video-text alignment, models like VideoBERT and HowTo100M capture temporal and contextual dependencies between video and textual modalities [37, 41].

2.1.3 Multimodal Large Language Models (MLLMs)

Multimodal Large Language Models (MLLMs) have emerged as a transformative innovation in artificial intelligence by integrating diverse modalities such as vision, language, audio, and others into a unified reasoning framework. Building on the foundation of traditional Large Language Models (LLMs) like GPT-3 [42] and T5 [43], MLLMs extend their textual reasoning capabilities to include perceptual inputs and cross-modal generation. This evolution has been driven by the need to handle increasingly complex multimodal tasks, ranging from visual reasoning and video analysis to speech-to-text conversion and beyond [1, 2, 25].

A central characteristic of MLLMs is their ability to map heterogeneous modalities into a shared semantic space, facilitating effective alignment and interaction. Pioneering models such as CLIP [1] and ALIGN [2] employed contrastive learning techniques to align vision and language by training on large-scale image-text datasets. These methods enabled zero-shot classification, retrieval, and other tasks, setting a new standard for cross-modal reasoning. More recent models, such as BLIP-2 [40] and LLaVA [12], extend this paradigm by incorporating instruction-tuned frameworks to align visual and textual embeddings, leveraging pre-trained LLMs to enhance downstream task performance.

The integration of robust alignment mechanisms has played a pivotal role in the development of MLLMs. Models like Flamingo [3] and PaLI [44] introduced specialized adapters to bridge the gap between visual and textual modalities, allowing pre-trained LLMs to process multimodal inputs seamlessly. Innovations such as LLaMA-Adapter [45] and MiniGPT-4 [14] further optimized alignment by introducing parameter-efficient fine-tuning techniques, enabling effective cross-modal reasoning with reduced computational overhead. These advancements highlight the growing emphasis on making MLLMs both scalable and adaptable to diverse input sources.

These advancements are largely built upon the Transformer architecture, which has become the backbone of modern MLLMs. Its self-attention mechanism supports fine-grained interactions across modalities, while its scalability allows pretraining on massive and heterogeneous datasets. Transformer-based models such as Flamingo, PaLI, BLIP-2, and LLaVA have demonstrated strong performance across diverse tasks, including visual question answering (VQA), image captioning, and multimodal dialogue. Notably, instruction tuning further enhances their flexibility, enabling generalization to unseen tasks guided by natural language prompts.

Recent Transformer-based MLLMs have shown varying strengths across multimodal benchmarks. For instance, CLIP and ALIGN excel in retrieval and classification tasks due to strong contrastive pretraining, while BLIP-2 and Flamingo achieve higher scores in image

captioning and visual dialogue through generative modeling. LLaVA and MiniGPT-4, by integrating instruction tuning with large language models, offer improved alignment and task generalization in visual question answering. These comparisons highlight that architectural choices and training paradigms (contrastive vs. generative, alignment vs. instruction-tuned) significantly impact model performance across tasks.

A defining strength of MLLMs lies in their ability to perform multimodal generation. GPT-4 [4], for instance, integrates vision input to handle tasks like image captioning, diagram interpretation, and visual-text reasoning. Similarly, AudioPaLM [46] expands these capabilities to audio-text interactions, demonstrating the potential of MLLMs in tasks such as speech-to-text transcription, audio-based reasoning, and multimodal dialogue generation. These generative capabilities underscore the versatility of MLLMs in addressing real-world applications where multiple modalities intersect.

Recent MLLMs have also benefited from large-scale pretraining on extensive multimodal datasets. Examples include Flamingo [3] and BLIP [40], which utilize curated image-text datasets for vision-language alignment, and AudioPaLM [46], which extends this paradigm to include speech-text data. Instruction tuning has further enhanced the adaptability of these models, enabling them to perform diverse tasks guided by natural language prompts [12, 13, 14].

As a rapidly evolving field, MLLMs continue to redefine multimodal learning by bridging the gap between perception and reasoning. They provide a unified framework capable of handling complex multimodal interactions, paving the way for breakthroughs in domains such as healthcare, autonomous systems, and human-computer interaction [47, 48]. By integrating multimodal inputs into a cohesive reasoning pipeline, MLLMs demonstrate the potential to fundamentally transform AI capabilities across a wide range of applications.

2.1.4 Multimodal Audio Learning

Audio, as a temporal and information-rich modality, offers significant complementary insights when combined with text, vision, or other data streams. Multimodal audio learning has been instrumental in advancing tasks such as audio-visual speech recognition, audio-guided image generation, and multimodal dialogue systems. The integration of audio with other modalities enhances contextual understanding, enabling systems to better capture temporal patterns and semantic alignments.

Early efforts in multimodal audio learning often relied on statistical methods and handcrafted features to bridge the gap between audio and other modalities. For example, Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) were used in audio-visual

speech recognition to combine lip-reading with acoustic signals, achieving improved recognition accuracy under constrained conditions [49, 23]. Dynamic Time Warping (DTW) was another commonly employed technique to align audio waveforms with textual transcripts in speech-to-text systems [50]. While these methods provided initial success, their reliance on domain-specific assumptions and inability to scale to complex tasks limited their broader applicability.

The rise of deep learning marked a paradigm shift in multimodal audio learning. Models such as Deep Speech [51] and WaveNet [52] demonstrated the capability of end-to-end neural networks to generate and process high-quality speech features. In parallel, audio-visual models combining convolutional neural networks (CNNs) with recurrent neural networks (RNNs) were developed for tasks like audio-visual speech recognition, leveraging temporal correlations between lip movements and acoustic signals [23]. Cross-modal fusion methods, including bilinear pooling [53] and attention-based mechanisms [54], further enabled seamless integration of audio with textual embeddings, advancing applications such as spoken question answering and multimodal retrieval.

Transformer-based architectures have since emerged as the dominant paradigm in multimodal audio learning. By capturing long-range dependencies and enabling cross-modal attention, transformers have transformed the landscape of multimodal integration. Notable examples include AV-HuBERT, which aligns audio signals with visual lip movements for robust speech recognition, particularly in noisy environments [55]. Whisper extends this paradigm by combining audio and textual modalities, delivering robust transcription capabilities under challenging acoustic conditions [56]. Furthermore, models like SALMONN process audio, text, and visual inputs simultaneously, facilitating complex tasks such as audio-guided video captioning and multimodal retrieval [57].

The adoption of large-scale pretraining has further propelled advancements in multimodal audio learning. Pretrained models trained on diverse datasets enable generalization across domains, making them effective in tasks such as speech-to-text transcription, audio-visual question answering, and multimodal storytelling [25, 41]. Instruction tuning, inspired by natural language processing, has emerged as a promising technique to align multimodal models across diverse downstream tasks [12, 13].

By integrating audio with other modalities, multimodal audio learning addresses complex, real-world challenges, ranging from conversational AI systems to audio-guided robotics. As the field evolves, these methods promise to deepen our understanding of cross-modal relationships and enhance the capabilities of multimodal systems.

2.2 Applications of Multimodal Learning

Multimodal learning has enabled transformative advancements across numerous domains, including intelligent agents, robotics, content generation, and human-computer interaction. By integrating diverse modalities such as vision, text, and audio, multimodal systems can tackle complex real-world challenges that require dynamic perception, reasoning, and action.

Intelligent Agents in Complex Environments One of the most impactful applications of multimodal learning lies in the development of intelligent agents that can interpret and act upon heterogeneous inputs. Mobile agents, in particular, have transformed interactions with graphical user interfaces (GUIs), automating complex tasks on smartphones and other devices. These agents integrate vision-based UI parsing [58, 59], optical character recognition (OCR)[60, 61], and language understanding[62, 63] to navigate and execute tasks effectively. Recent advancements leverage large language models (LLMs) to enhance the reasoning and decision-making capabilities of mobile agents [64, 65, 66], enabling them to dynamically adapt to varying app structures and user instructions.

In embodied AI, multimodal learning enables robots to integrate visual, spatial, and tactile inputs, supporting navigation and manipulation in dynamic environments [67, 68]. These systems excel in tasks such as robotic grasping [69] and warehouse automation [70], where multimodal inputs ensure context-aware decision-making. Similarly, virtual assistants enhanced with multimodal reasoning capabilities [71, 19, 72] have been employed in domains ranging from customer service to education, enabling seamless interactions across text, speech, and visual inputs.

Multimodal Content Generation and Retrieval Multimodal learning has redefined content generation and retrieval by enabling systems to produce and organize information across modalities. In content generation, models trained on large-scale vision-language datasets excel at creating descriptive captions for images and videos [1, 40]. These systems enhance accessibility for visually impaired users and improve the discoverability of multimedia content [3, 9]. Advances in cross-modal alignment further enable retrieval systems to bridge queries in one modality with results in another, such as searching for images using textual descriptions or identifying audio content through visual tags [2, 8].

In audio-visual integration, multimodal learning improves speech recognition by combining acoustic and visual signals [23, 55]. This is particularly effective in challenging conditions, such as noisy environments, where lip-reading enhances transcription accuracy. Similarly, systems for video captioning and summarization leverage multimodal inputs to

extract meaningful information from both audio and visual streams, enabling applications in media indexing and automated content creation [41, 37].

Creative and Interactive Systems Multimodal learning has opened new frontiers in creativity and interaction. Systems for audio-guided image generation [73, 74] allow users to create visuals based on verbal descriptions, democratizing artistic expression. Similarly, interactive art installations utilize multimodal inputs, including speech, gestures, and visual cues, to create immersive experiences that adapt in real time to user interactions [75, 76]. These systems bridge human creativity with computational capabilities, highlighting the transformative potential of multimodal learning.

In personalized education, multimodal systems analyze visual, textual, and auditory inputs to tailor content delivery to individual learning styles [47, 77]. For instance, adaptive platforms generate real-time feedback on students' handwriting while simultaneously providing verbal instructions, ensuring an inclusive learning experience. These applications showcase how multimodal learning can enhance engagement and learning outcomes across diverse user groups.

Applications in Healthcare and Autonomous Systems In healthcare, multimodal learning integrates imaging data, clinical notes, and sensor readings to improve diagnostic accuracy [47, 78]. For example, radiology systems combine textual reports with imaging features to detect anomalies and track disease progression [79]. Similarly, wearable health monitors use multimodal data from physiological sensors to provide real-time alerts and health recommendations [77, 80].

In autonomous systems, multimodal learning facilitates navigation, obstacle detection, and decision-making in dynamic environments [81, 82]. Autonomous vehicles rely on a combination of visual, spatial, and audio inputs to adapt to changing road conditions and ensure safety [83, 84]. Delivery robots and industrial automation systems further exemplify the role of multimodal learning in optimizing task efficiency and reliability [68, 66].

2.3 Research Gaps

Despite the rapid development of multimodal large language models (MLLMs), several critical gaps remain that hinder their effectiveness in real-world applications.

First, there is a lack of accurate and semantically aligned training data that supports fine-grained cross-modal understanding. Many existing datasets suffer from noisy labels, weak image-text alignment, or limited domain diversity, leading to degraded performance in tasks such as image-based dialogue and multimodal reasoning. Current models often

rely on large-scale web data with minimal filtering, which introduces inconsistencies and hallucinations. There is a clear need for a high-quality, customizable data pipeline that produces aligned image-text pairs tailored for diverse domains.

Second, existing MLLMs face significant limitations in adapting to dynamic and resource-constrained environments, such as mobile platforms and embedded systems. While instruction tuning and adapter modules have improved generalization, many models still require expensive fine-tuning to adapt to new interfaces or domains. Moreover, their memory and computational overhead remain high, making them impractical for real-time applications. Efficient architectures and memory-driven interaction frameworks are needed to enable on-device inference and flexible task adaptation without retraining.

Third, current evaluation benchmarks fall short in capturing model robustness under realistic audio-based interaction settings. Most multimodal benchmarks rely on clean, synthetic inputs and fail to account for noisy environments or adversarial attacks, particularly in speech-based scenarios. This makes it difficult to assess how models behave under degraded or manipulated inputs, which is critical for safety-sensitive applications. There is a pressing need for a systematic benchmark that evaluates robustness across multiple attack types and auditory distortions using both standard metrics and human-aligned evaluations.

This thesis addresses these challenges by proposing: (1) a controllable data generation pipeline that enhances cross-modal alignment and reduces hallucinations; (2) a lightweight, feedback-driven agent framework that improves efficiency and task adaptability in mobile settings; and (3) a comprehensive audio-language robustness benchmark, featuring realistic adversarial scenarios and layered evaluation strategies.

CHAPTER 3

ENHANCED VISUAL INSTRUCTION TUNING WITH SYNTHESIZED IMAGE-DIALOGUE DATA

The remarkable multimodal capabilities demonstrated by OpenAI's GPT-4 have sparked significant interest in the development of multimodal Large Language Models (LLMs). A primary research objective of such models is to align visual and textual modalities effectively while comprehending human instructions. Current methodologies often rely on annotations derived from benchmark datasets to construct image-dialogue datasets for training purposes, akin to instruction tuning in LLMs. However, these datasets often exhibit domain bias, potentially constraining the generative capabilities of the models. In an effort to mitigate these limitations, we propose a novel methodology for data collection, which synchronously synthesizes images and dialogues for visual instruction tuning. This approach leverages the combined capabilities of generative text-to-image models and ChatGPT, facilitating the creation of a dataset that is both diverse and scalable, and more importantly, customized to enhance the models' performance across a broad spectrum of tasks. Our research includes comprehensive experiments conducted on various datasets. The results emphasize substantial enhancements in more than ten commonly assessed capabilities. Additionally, our model achieves state-of-the-art results across multiple widely recognized multimodal benchmarks.

3.1 Introduction

The launch of OpenAI's ChatGPT[85] has marked a significant milestone in artificial intelligence (AI), showcasing the advanced capabilities of Large Language Models (LLMs). These models, exemplified by GPT-4[4], demonstrate exceptional versatility by handling not just images but also excelling in tasks once difficult to accomplish.

This includes understanding humor within images and drafting website code from basic sketches, aspects that highlight its revolutionary potential.

However, despite these notable achievements, a crucial aspect remains undisclosed: the specific mechanics underlying GPT-4, particularly concerning the seamless integration of multimodal information into LLMs. This knowledge gap has prompted a concerted research effort to address this puzzle.

Among the promising approaches, an emerging method receiving considerable attention involves the utilization of adapter-based techniques [45, 86, 87], which allow the training of a visual-to-text adapter that convert features from pre-trained visual models into LLM tokens, showing promise in achieving results comparable to GPT-4.

The effectiveness of adapter-based methods stems from their ability to leverage the extensive pre-existing knowledge in large visual models and LLMs. By focusing on training a lightweight adapter, these methods avoid the computational expense of training comprehensive models from scratch, thereby offering a more efficient pathway to enhancing LLMs' multimodal integration capabilities.

A prerequisite for implementing these frameworks is the availability of paired vision-text image data. Such datasets are essential for aligning visual and textual information, facilitating the LLMs' understanding of complex human instructions. Analogous to instruction tuning in LLMs [88], this process is commonly referred to as visual instruction tuning.

Existing methods [89, 7, 90, 91, 92] typically construct visual instruction tuning datasets by leveraging established vision datasets, extracting information such as image captions, spatial locations, and categories to form dialogues. This approach maximizes resource utilization, creating a comprehensive and efficient training dataset for multimodal LLMs.

Despite the efficiency and simplicity of this approach to dataset construction, certain limitations still persist. Existing large-scale vision-text datasets, such as LAION [8] and CC [9], often contain noise. Consequently, training only a subset may inadequately align visual-text features for immediate user requirements. Moreover, benchmark datasets [9, 8, 93] often exhibit a domain bias, primarily in terms of image styles. For instance, prevalent datasets such as COCO [93] predominantly feature images from everyday life, while stylized images

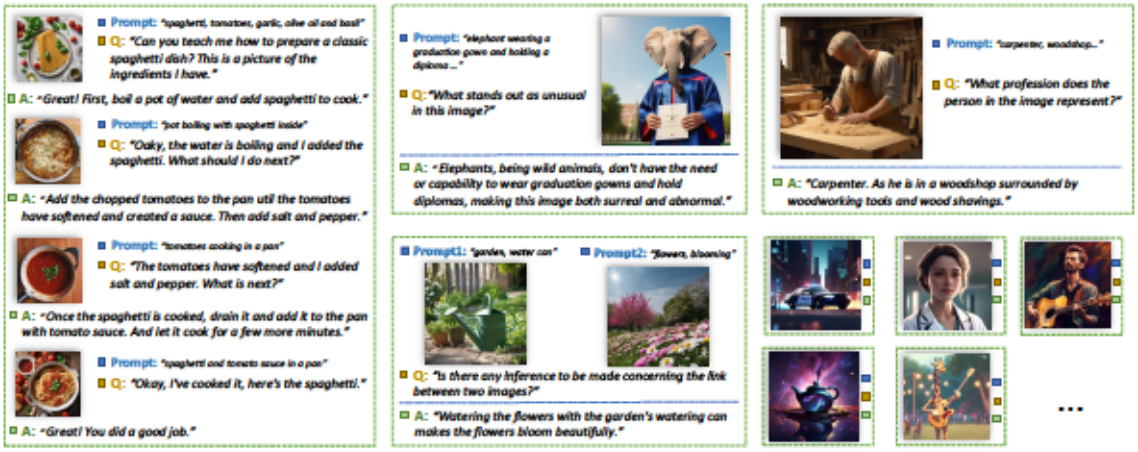

Figure 3.1: **Examples of synthesized visual instruction data**. We use ChatGPT and text-to-image generation models to synthesize various forms of visual instruction tuning data, such as multi-round dialogue data, multi-image reasoning data, and anomaly detection data. These data are used to train the multimodal large language models.

like cartoons are rarely represented. Additional, these vision annotations may also constrain the types of dialogues generated from them. For example, almost none of the current datasets contain data to directly enhance the model's ability to comprehend jokes presented in the images, an impressive feature of GPT-4. Moreover, as multi-image dialogues become increasingly integral to the practical application of multimodal LLMs, the current lack of comprehensive multi-image datasets further underscores the need for data enrichment in this domain.

In addressing these limitations, we propose a novel data collection approach to enhance visual instruction tuning. Building upon recent successes in the AI-Generated Content (AIGC) field, we leverage generative models to produce image-dialogue pair data for visual instruction tuning. More concretely, we employ ChatGPT to create data that includes image-generating prompts and content-based dialogues. We then utilize the text-to-image diffusion model, StableDiffusion [94], to generate images based on these prompts. Finally, the synthesized images and generated dialogues are employed to train multimodal LLMs.

Simultaneously generating both image content and dialogues enables the production of diverse training data, affording greater control over its nature and quality. This flexibility allows us to construct multi-turn dialogues and datasets involving multi-image reasoning, which are challenging to obtain from other benchmarks.

Additionally, our methodology can potentially integrate more advanced image generative models, such as DALLE3 [95], to provide higher-level control over image contents like

specifying complex spacial relations. This advanced control could generate more complex instructions to enhance image understanding capabilities. Examples from our synthesized visual instruction tuning datasets are shown in Figure 3.1. Building upon the flexible pipeline outlined above, users can tailor the generation of data to enhance specific capabilities based on their task requirements. Furthermore, our method of generating both images and dialogues eliminates constraints on data volume, thereby facilitating the production for limitless scaling of the datasets.

To demonstrate the effectiveness of our proposed pipeline, we conducted extensive experiments. Our main contributions are as threefold:

- We develop a novel pipeline for generating visual instruction tuning datasets by leveraging text-to-image diffusion models.
- To showcase its flexibility, we have built a dataset with various form of capabilities including multi-image data, and our results have shown improvements across all abilities.
- Extensive experimental analysis on multiple benchmarks shows the effectiveness of the proposed method, outperforming baseline and existing SOTA approaches.

3.2 Related Work

Recent research [14, 7, 13] efforts in multimodal Large Language Models (LLMs) have yielded promising strategies to efficiently align the embeddings of other modalities with language tokens. This has made it possible to effectively utilize pre-trained encoders from other modalities and LLMs, which effectively reduces the computational burden and training time. While there are alternative research approaches that include training-free methods leverage expert models [96, 97, 98], these are not the focus of our work here.

Adapter-based LLMs represent a significant research direction, introducing methods to connect modalities through learnable interfaces with minimal training efforts.

These approaches [45, 86, 87, 7, 14, 13, 99] allow for the use of pre-trained modal encoders, reducing the need for training from scratch. Variations include direct training of projection layers for embedding alignment and the use of learnable queries for extracting modality-specific information, as seen in models like LLaVA [7] and Flamingo [3]. Innovations such as the LLaMA-Adapter [45] and LaVIN [87] have introduced lightweight and mixed-modality adapters, respectively, enhancing the field's diversity.

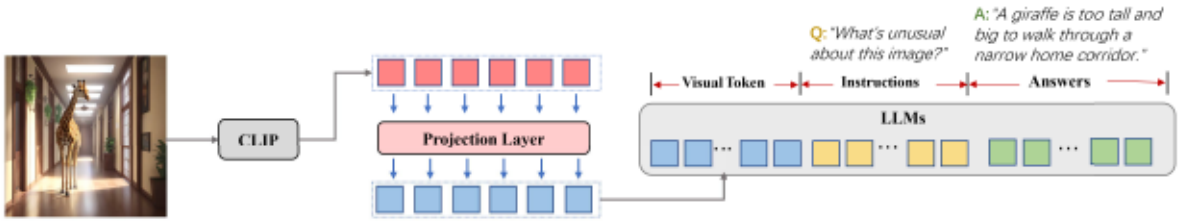

Figure 3.2: **Architecture of LLaVA**. We use the open-source LLaVA model as a testbed for our proposed data generation pipeline. The model is trained to predict the next tokens in the answers given the visual tokens and instruction tokens in an auto-regressive manner.

Visual instruction tuning datasets are crucial for training multimodal LLMs, focusing on aligning modalities and enabling instruction following. Most existing methods [7, 100, 101, 102, 103, 91, 104] rely on benchmark datasets for constructing visual instruction tuning datasets, which may be limited by the fixed categories in annotations. Our method leverages well-trained image generation models to produce controllable image data, enhancing multimodal LLM capabilities and allowing for the integration of advanced generative models for specific guidance forms, offering a more flexible and diverse approach to dataset construction.

3.3 Preliminary

To assess the effectiveness of our data generation strategy, we chose the open-sourced LLaVA [7, 105] as our multimodal LLM model. LLaVA offers a strong balance between performance and accessibility, with open weights and a well-documented training process that facilitates reproducibility. It should be noted that our pipeline is model-agnostic, making it applicable for various models. This section serves as a foundation, briefly summarizing the LLaVA model's design and training methods to prepare for a thorough exploration of our pipeline. The reader may refer to the original publication [7] for detail.

Architecture. The LLaVA model integrates Vicuna-13B [106] as the language model with a pre-trained CLIP visual encoder ViT-L/14 [1] for extracting visual features, transforming these features into language embedding tokens through a linear layer. This linear layer was updated in LLaVA-1.5 [105] with a two-layer MLP, replacing Vicuna-13B with Vicuna-13B-v1.5 and increasing input image size to 336x336. A detailed illustration of this model structure can be found in Figure 3.2.

Training and datasets. LLaVA's training focuses on visual instruction tuning with data

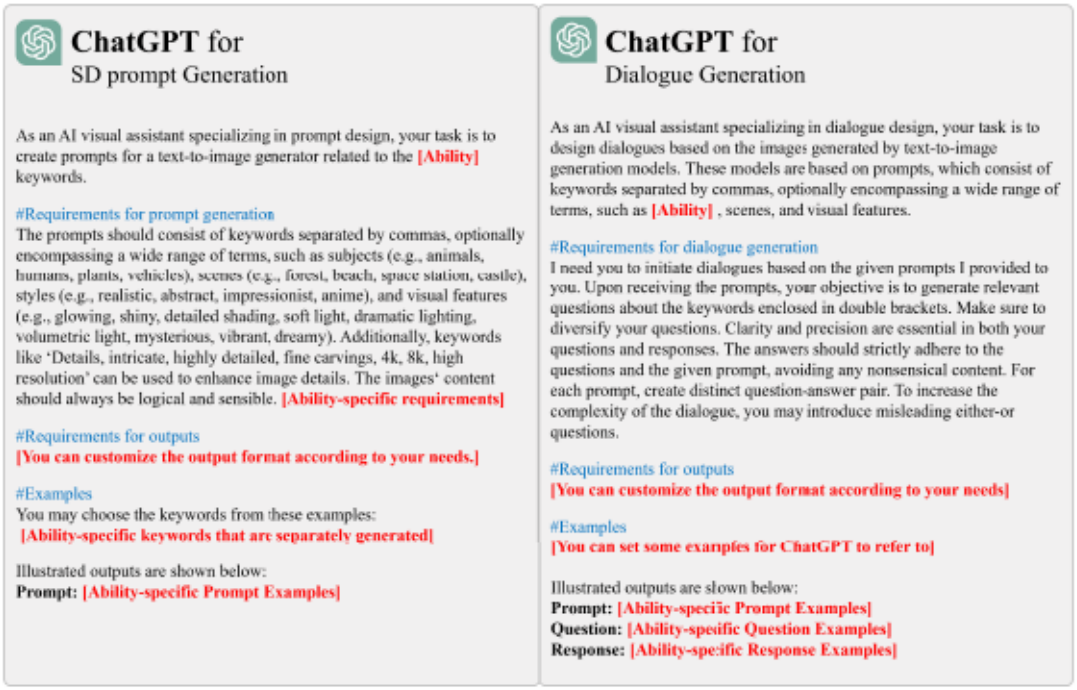

Figure 3.3: **Templates for guiding ChatGPT to generate StableDiffusion prompts (left) and dialogues (right).** Content in red represents ability-specific information. We only provide an example template for constructing dialogues regarding a single image in this figure. For additional forms of data, such as multi-image reasoning and multi-turn dialogues, please refer to our supplementary materials.

triplets: images, questions, and answers, aiming for predictive accuracy in an autoregressive manner. The training comprises two stages: the first emphasizes modality alignment using 595K image-text pairs, optimizing the linear layer with static visual encoder and LLM weights. The second stage, using 158K multimodal dialogue data from COCO, extends optimization to the LLM's weights for comprehensive modality integration. LLaVA-1.5 further enriches the dataset by incorporating additional data like Region-level VQA [107, 108, 109] and GQA [110], expanding the second-stage dataset to 665K examples.

3.4 Methods

This section outlines our dual-generation approach for creating visual instruction tuning datasets, which synthesizes images and their corresponding dialogues, as illustrated in Figure 3.4. We detail each component below.

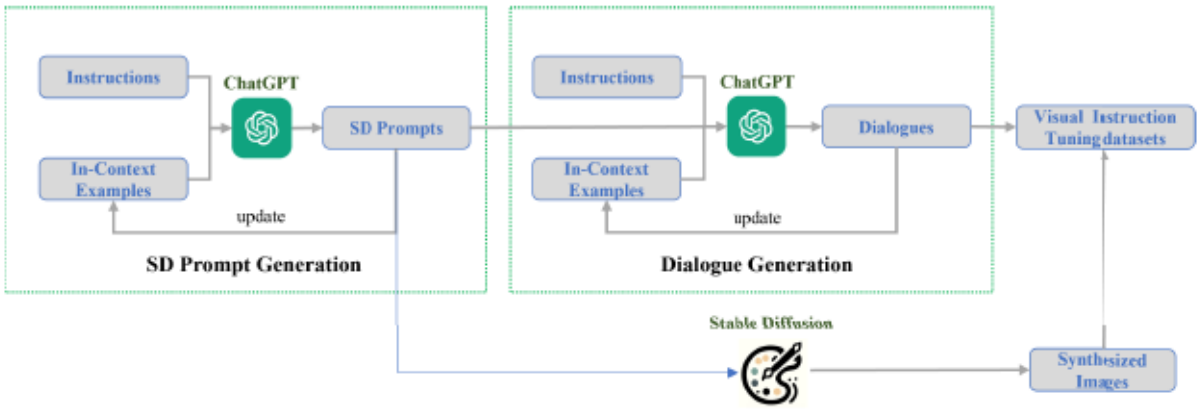

Figure 3.4: **Our proposed pipeline for generating visual instruction tuning datasets.** We instruct ChatGPT to generate both StableDiffusion prompts and the associated dialogues. For specific generation templates, please refer to the supplementary materials.

3.4.1 Image Generation

We employ StableDiffusion [94] to generate images based on prompts that include weighted keywords reflecting aspects like subject, scene, style, and visual elements such as image quality and lighting. Keywords at the prompt's start are prioritized, with the possibility of adding emphasis using brackets. To encourage diversity and stability during image generation, we add capability-specific instructions and cautions during prompting ChatGPT. For instance, in the task of generating images for joke understanding, we direct ChatGPT to create prompts that would result in the generation of abnormal images, like a "giraffe walking through a narrow corridor", which are unlikely to be found in reality. When generating multi-image data, pairs of prompts can be generated concurrently based on predefined specific criteria. For maximum effect, we ensure that the most crucial keywords are placed at the beginning of the generated prompts, which are double-bracketed for additional emphasis. Furthermore, we instruct ChatGPT to avoid generating prompts that are non-visual, such as the act of growing. The instruction template for prompt generation is provided in the left part of Figure 3.3. The generated prompts are then used with StableDiffusion to produce visually realistic images, which are subsequently encoded by LLaVA's vision encoder into visual tokens for LLMs.

3.4.2 Dialogue Generation

Following the generation of images, we utilize ChatGPT to generate dialogues based on the same prompts used for image synthesis, aligning with LLaVA's training objectives: the first stage focuses on aligning visual-text data, and the second on processing diverse instructions.

Dialogues for the initial stage describe the images, where ChatGPT generates answers to predefined questions about the images' content.

Taking the example of the "giraffe walking through a narrow corridor", a representative dialogue might be: "Question: What is unusual in the image? Response: In reality, a giraffe is too tall and big to walk through a narrow home corridor." The detailed instruction template for dialogue generation is shown in the right part of Figure 3.3.

For the second stage, dialogues aim to enhance reasoning across multiple images, addressing similarities, differences, and logical connections, and include multi-turn dialogues that blend image and text. We guide ChatGPT to produce a range of question types, steering clear of inherently ambiguous questions to ensure clarity and accuracy, detailed template can be found in the supplementary materials.

3.4.3 In-Context Examples

ChatGPT's in-context learning capability, which allows it to grasp the essence of tasks from a few examples, is leveraged in our methodology. We incorporate in-context examples in the generation of StableDiffusion prompts and dialogues to enhance this learning process.

During the data generation process, we observed that ChatGPT sometimes produced a lack of diversity. For example, when generating colors, the outputs frequently revolved around common color categories. To overcome this, we independently generate ability-related keywords such as color categories with ChatGPT, and utilize them as a reference during the prompting process. This additional step promotes a more diverse range of prompts, thereby enriching our visual instruction tuning dataset.

We further adopt a dynamic strategy to maintain and increase diversity: periodically substituting a portion of the original in-context examples with newly generated data. This continuous update prevents over-repetition and ensures the dataset's comprehensiveness and representativeness, maintaining a balance that contributes to a richer and more diverse visual instruction tuning dataset.

3.4.4 Data Filtering Mechanism

To ensure the quality, diversity, and reliability of our generated dataset, we incorporate a comprehensive data filtering and control mechanism, which addresses prompt bias, hallucination, and prompt accuracy concerns raised when using ChatGPT.

- **Repetition Rate Filtering** We first filter out prompts with high lexical or semantic repetition to enhance diversity and reduce model-induced bias. This helps prevent

over-representation of specific concepts and encourages broader coverage across different visual scenarios. The filtered, diverse prompts are then used to generate corresponding images and dialogues.

- **Length-based Filtering** We constrain the length of prompts to a maximum of ten keywords to ensure clarity and prevent overly complex or ambiguous descriptions that may lead to hallucinated or ungrounded image content. Similarly, we cap dialogues at 500 characters to promote concise, focused responses and reduce the risk of speculative or irrelevant generation.
- **Task-Specific Restrictions** For certain categories, we implemented restrictions based on specific attributes of the capabilities. For example, when generating content related to construction workers, the model tended to focus on buildings. To address this, additional human attributes were incorporated into the prompts to ensure they remain focused and semantically aligned with the intended subject, thereby reducing the risk of content drift or hallucination.
- **Alignment Check** To ensure a high degree of alignment between generated images and dialogues, we employ the CLIP [1] model to compute matching scores for both texts and images. Data entries with scores exceeding a predefined threshold, set at $\gamma = 0.25$, are retained, thereby filtering out less relevant matches and elevating the overall data quality.

3.5 Experiments

In this section, we detail the experiments conducted to validate the effectiveness of our novel data collection approach for visual instruction tuning. We describe the training datasets, evaluation strategy, and both quantitative and qualitative outcomes.

3.5.1 Training Datasets

We generate a diverse and expansive dataset to show its versatility, covering single-image capabilities from basic recognition to complex visual reasoning. This includes understanding physical attributes, life features, and man-made items, among others, amounting to 38K image-dialogue pairs for initial training. Each ability's dataset was formulated following a standard template, illustrated in supplementary material.

In addition, we also generated a dataset of 3K multi-image instances, encompassing descriptions of image similarity, difference, logical relations, and multi-turn dialogue data

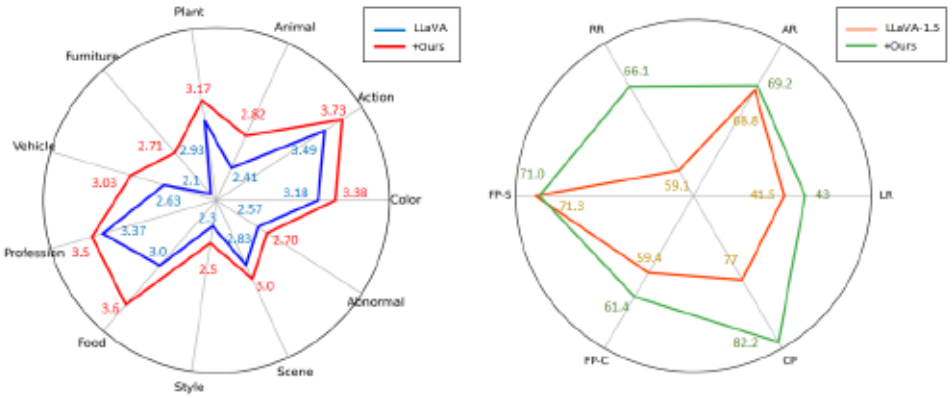

Figure 3.5: Left: Results on evaluation benchmarks for various abilities (GPT-4 score). Right: Comparison of various subcategories on MMBench [111] with the baseline (Accuracy). Our model outperforms the baselines on both benchmark datasets.

for the second stage. These datasets, in combination with the raw LLaVA dataset, provides a comprehensive training set in our experiments.

3.5.2 Experiments Setup

During the model training phase, we employed the original LLaVA configuration as the foundation for our training process. In both stages, we utilized 8 NVIDIA V100 GPUs. To conserve GPU memory, we employed deepspeed with zero3 during model training, disabling tf32 and opting for fp16. The remaining parameters, including epochs and learning rates, were set according to the original LLaVA configuration. For specific parameter details, please refer to the the original publication.

3.5.3 Evaluation Metrics

Evaluation datasets. To demonstrate our performance more clearly, we tested on a series of public multi-modal datasets, including VisWiz [112], MM-Vet [113], MME [114], and MMBench [111].

Subsequently, we established a real-image benchmark to evaluate training effectiveness across a wide range of single-image abilities, including 330 test samples of real images with associated question-answer dialogues, carefully selected and annotated from public repositories. This comprehensive benchmark aims to rigorously test the models' single-image capabilities.

Simultaneously, we constructed a multi-image test set consisting of 30 dialogues to assess the models' performance on this specific data type. This dataset evaluates the models

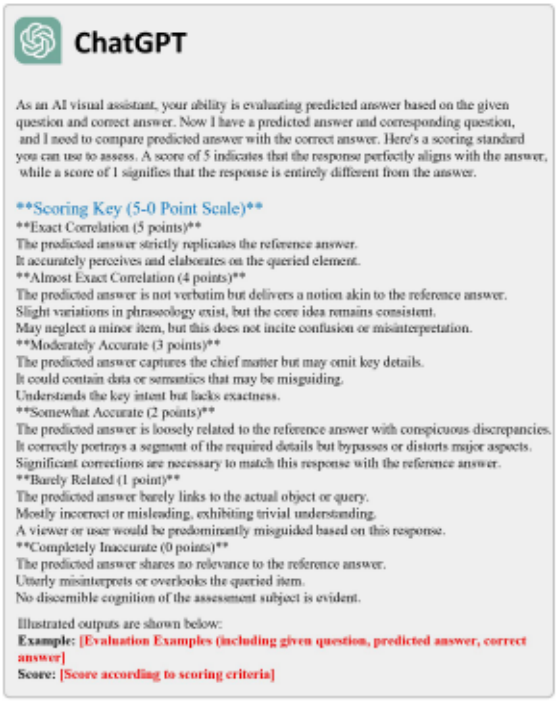

Figure 3.6: Score criteria based on GPT-4.

across differences, similarities, and reasoning relationships among the images. The test data was sourced from publicly available datasets and manually annotated.

Evaluation Strategy In terms of the evaluation process, we employ different testing strategies depending on the benchmarks used.

Our evaluation approach varies with the benchmark. For multimodal datasets like VizWiz [112] and MMBench [111], we follow official guidelines, converting test data to a compatible format for our model, and using official scripts or submission portals for assessment, primarily focusing on accuracy.

In evaluating the diverse capabilities we've generated, We adopted two evaluation methods, namely manual evaluation and evaluation based on GPT-4 score.

Initially, the participants were instructed to assess the answers produced by our model and those of the baseline for all abilities according to the label (1 for correct, 0 for incorrect), averaging these as the final metric.

Subsequently, inspired by [106, 7], we leverage GPT-4 [4] to assist in scoring model outputs. We have established six scoring levels, ranging from 0 to 5. Each score level is

accompanied by detailed descriptions of the evaluation criteria, and we assist GPT-4 in better assessment by providing a series of scoring examples. In particular, a score of 0 indicates that the predicted answer has no relevance to the reference answer, while a score of 5 signifies that the predicted answer aligns seamlessly with the annotated reference answer without any deviation. Drawing on our manual annotations, we conduct evaluations on the results produced by each model. The average GPT-4 score within each test set serves as the ultimate metric for our benchmark evaluations.

As shown in the Figure 3.6, we present our detailed GPT-4 scoring system. We have established a scale of 0-5 with six levels of scores, and for each score, we provide detailed evaluation criteria along with specific examples for assessment. Utilizing the template in the Figure 3.6, evaluations are conducted for each model, and the average of the results is taken as the final score.

Table 3.1: Quantitative performance (Accuracy) on real-image evaluation benchmark for manual evaluation.

Method	Animal	Action	Color	Abnormal	Scene	Style	Food	Profession	Vehicle	Furniture	Plant
LLaVA	0.63	0.67	0.60	0.40	0.60	0.30	0.70	0.57	0.57	0.30	0.53
Ours	0.70	0.90	0.77	0.50	0.77	0.45	0.83	0.63	0.63	0.50	0.57

3.5.4 Quantitative comparison to state-of-the-arts

Public multimodal benchmarks We perform quantitative performance comparisons against various state-of-the-art methods on different benchmarks, as illustrated in Table 3.2. Utilizing LLaVA-1.5-13B as the baseline, we integrate our synthesized data with its original dataset for training. Training is carried out with identical parameter configurations as LLaVA-1.5. The outcomes demonstrate substantial improvements on many benchmarks, emphasizing the enhanced performance achieved by our approach.

Comparison of various abilities. To validate the effectiveness of our generated data, we conducted comprehensive tests on distinct capabilities, employing both manual evaluation and GPT-4 score assessments. Employing LLaVA-13B as our baseline, the quantitative comparison of the baseline results and ours for manual evaluation are shown in Table 3.1, while the GPT-4 score assessment results are shown in the left part of Figure 3.5. Notably, our trained model consistently outperforms the LLaVA-13B baseline across all various capabilities on two metrics, which suggests the synthesized datasets' generalizability and our pipeline's robustness.

Besides, we conduct a comparison of subcategory performance on MMBench to better validate our superiority, using the LLaVA-1.5-13B as the baseline. The tested subcategories

Table 3.2: Quantitative comparison with other state-of-the-arts methods on multiple multi-modal benchmarks. We achieve state-of-the-art performance on four benchmarks.

Method	VisWiz	MM-Vet	MME	MMB
BLIP2 [6]	19.6	22.4	1293.8	-
InstructBLIP [89]	33.4	25.6	1212.8	-
IDEFICS-9B [115]	35.5	-	-	48.2
IDEFICS-80B	36.0	-	-	54.5
Qwen-VL [15]	35.2	-	-	38.2
Qwen-VL-Chat	38.9	-	1487.5	60.6
LLaVA-1.5 [105]	53.6	35.4	1531.3	67.7
Ours	**58.4**	**36.1**	**1532.3**	**69.4**

Table 3.3: Quantitative results (GPT-4 score) on the multi-image benchmark. After the addition of multi-image data, various multi-image capabilities have significantly improved.

Method	Difference	Similarity	Logical relations	Average
LLaVA	2.7	2.2	3.1	2.67
Ours	**3.6**	**2.8**	**3.7**	**3.37**

in MMBench encompass six aspects: attribute reasoning (AR), coarse perception (CP), fine-grained perception (cross-instance) (FP-C), fine-grained perception (instance-level) (FP-S), logic reasoning (LR), and relation reasoning (RR). The final results are shown in the right part of Figure 3.5, indicating better performance of subcategory on MMBench, which also attests to the high quality of our generated data.

Comparison on multi-image benchmark In order to validate the effectiveness of multi-image capabilities, we manually curated a benchmark of real images. The evaluation metric used was the GPT-4 score mentioned. We used LLaVA-13B as the baseline and incorporated multi-image data in the second training phase. Since LLaVA itself lacks the capability for multi-image input, we modified the testing code for LLaVA to enable it to accept multiple sets of images simultaneously. The comparison with LLaVA results is shown in the Table 3.3, indicating a notable improvement across various multi-image capabilities despite adding less multi-image data in the process.

3.5.5 Qualitative results

Supplementing the quantitative analysis, we provide a qualitative comparison between our model's results and LLaVA-13B in Figure 3.7 on multi-image data. Our model exhibits a heightened ability to adhere to question instructions, rendering more precise answers.

We compare our approach with the LLaVA-13B baseline, revealing its limitations: it struggles to differentiate between multi-image contents and provides incomplete answers to questions. Our method, incorporating multi-image data, enhances the model's understanding of multiple images, demonstrating its effectiveness. The qualitative evaluation was

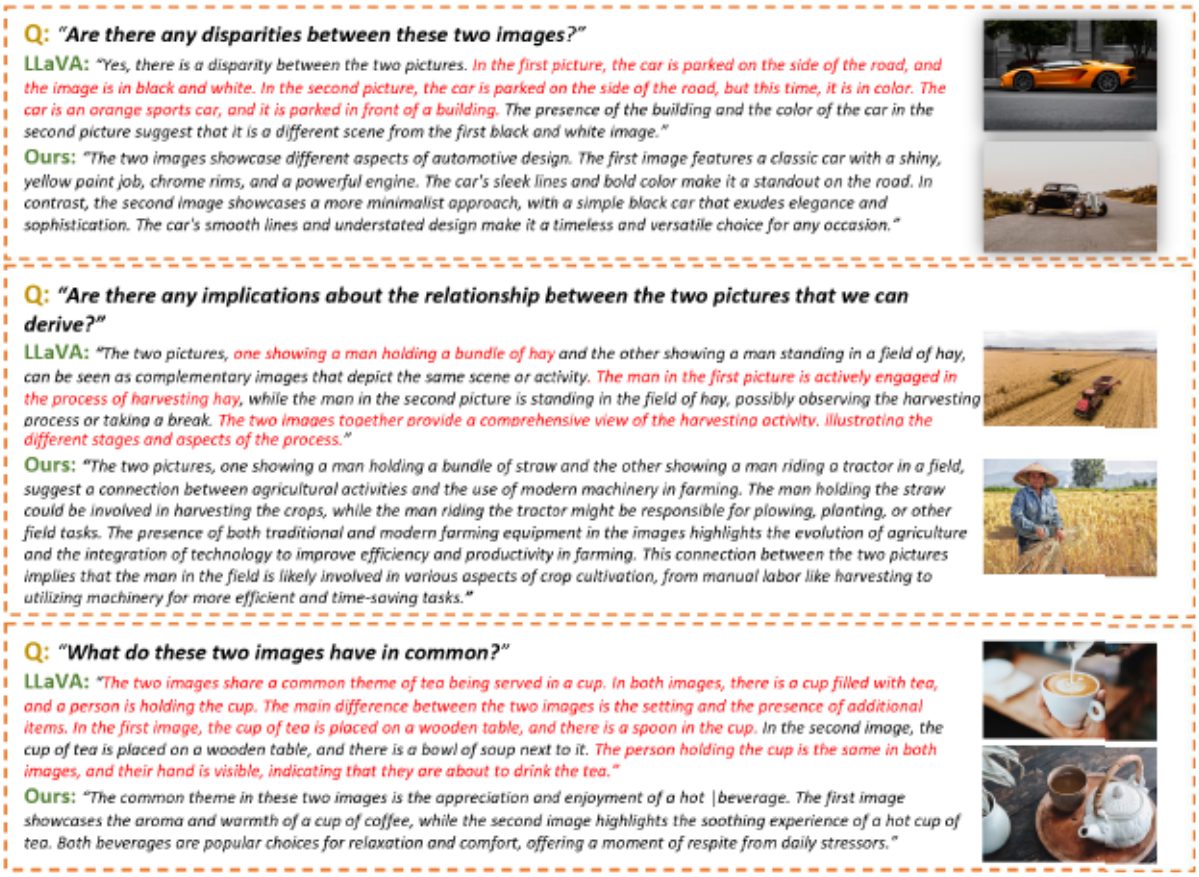

Figure 3.7: **Comparison of the results generated by LLaVA and our trained model.** Content in **red** represents inaccurate information. Our model can better adhere to question instructions, rendering more precise answers.

conducted jointly by three co-authors. Each author independently compared model outputs (our method vs. LLaVA-13B) on a curated set of multi-image examples. The evaluation focused on instruction-following, completeness, and accuracy. The inter-rater consistency was maintained through shared evaluation criteria. Additional qualitative results will be included in the supplementary materials.

3.5.6 RQ1 Revisited: Impact of Cross-Modal Alignment and Dataset Quality

This work addresses RQ1 by proposing a data generation pipeline that systematically improves the cross-modal alignment and quality of training data for multimodal large language models. At the core of our approach are strategies designed to reduce semantic noise, control hallucinations, and ensure image-dialogue consistency through structured prompt template

design and data filtering mechanism.

Rather than introducing architectural changes, we demonstrate that carefully constructed data alone can substantially enhance model performance. By ensuring accurate modality pairing and minimizing irrelevant or ambiguous training signals, the resulting model exhibits improved reasoning accuracy, stronger instruction following, and reduced hallucination across various interaction scenarios.

These findings support the hypothesis that cross-modal alignment and dataset quality are critical to the accuracy of MLLMs, thereby providing a direct and empirical answer to RQ1.

3.6 Conclusion and Future Work

In the rapidly evolving realm of Large Language Models, efficiently integrating multimodal information is a key research focus. In this study, we introduced an innovative data collection method to enhance visual instruction tuning for multimodal models. Compared to existing strategies, our approach uniquely combines image and dialogue generation, effectively addressing limitations found in benchmark datasets. By carefully crafting instruction templates, our method ensures high-quality training data covering a broad range of crucial capabilities for multimodal models and users can generate customized data based on their specific requirements.

Our research opens avenues for exploration. Moving forward, we aim to leverage advanced generative models to enhance model abilities, including spatial comprehension and fine-grained recognition. With promising results from our dual-generation method, forward-thinking data collection techniques are poised to play a significant role in the future of LLM research.

3.7 Limitations

Due to constraints in text-to-image models like stable diffusion, generating certain data types, such as text-rich images and tables, is not effective in the current pipeline. We anticipate these constraints will be addressed with ongoing advancements in text-to-image generation techniques.

Furthermore, text-to-image models such as stable diffusion are known to exhibit various forms of bias, as they are trained on large-scale web-crawled datasets that inherently contain societal and cultural imbalances. As a result, the synthesized images may reflect unintended

biases related to profession, gender, ethnicity, and geographic representation. For instance, prompts involving occupations may disproportionately depict certain genders (e.g., doctors as male, nurses as female), or favor Western-centric cultural aesthetics in generated content. Addressing such biases remains a challenging and open research problem in generative modeling.

In this work, we adopt prompt engineering techniques to partially mitigate these issues. Specifically, we control the generation process by predefining certain keywords and randomly sampling from curated keyword lists to enhance diversity and reduce stereotypical patterns. In addition, we incorporate a data filtering mechanism to further screen generated samples and discard those with high repetition or poor alignment. While these strategies alleviate some of the biases, ensuring fairness and representativeness in synthesized data remains a critical goal. In future work, we plan to incorporate human-in-the-loop filtering and more advanced bias mitigation techniques to improve the equity and reliability of the constructed datasets.

3.8 Ethics Statement

Our method leverages generative models to create synthetic images and dialogues. It is imperative to ensure that the generated content does not perpetuate or amplify biases present in existing datasets or societal prejudices. We have implemented data filtering mechanism to minimize the generation of potentially harmful or biased content. However, continuous vigilance and improvement of these filters are necessary as generative models evolve.

The enhanced capabilities of multimodal LLMs, facilitated by our data generation approach, could potentially be misused for creating deceptive or manipulative content. It is crucial to develop and adhere to guidelines that prevent the misuse of such technology, including transparent disclosure of synthetic content's nature and purpose.

CHAPTER 4

ADVANCED AGENT FOR FLEXIBLE MOBILE INTERACTIONS

With the rise of Multimodal Large Language Models (MLLM), LLM-driven visual agents are transforming software interfaces, especially those with graphical user interfaces. This work presents a novel LLM-based multimodal agent framework for mobile devices, designed to enhance interaction and adaptability across diverse applications. The agent autonomously navigates devices, emulating human-like interactions, and constructs a flexible action space by integrating parsing, text, and vision descriptions. It operates in two phases: exploration, where UI functionalities are documented into a structured knowledge base, and deployment, where Retrieval-Augmented Generation is used to efficiently access and update the knowledge base for accurate task execution. Experimental results across multiple benchmarks validate the framework's superior performance and practical effectiveness. Through a user study, involving both agent-driven and manual exploration, further demonstrates improved task success rates and user satisfaction, highlighting the robustness and adaptability of the proposed approach.

4.1 Introduction

Large Language Models (LLMs) like ChatGPT [85] and GPT-4 [4] have greatly advanced natural language processing, enabling their integration into intelligent agents that revolutionize autonomous decision-making. Initially designed for text-based interactions, these agents [116, 117] incorporate advanced features such as adaptive memory, which enhances their engagement with environments and processing across diverse natural language tasks.

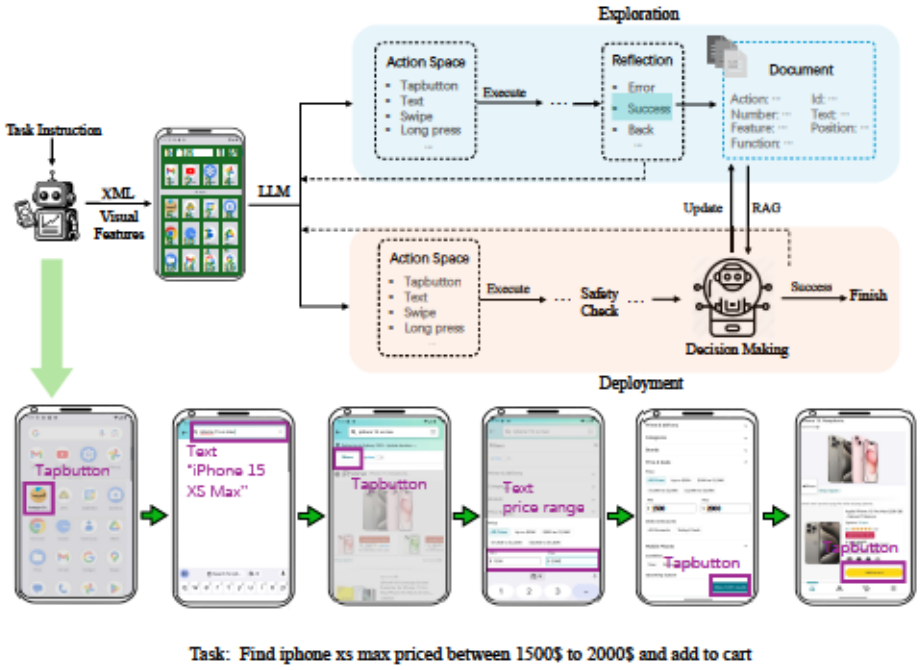

Figure 4.1: Overview of our proposed agent framework. The diagram illustrates the agent's workflow starting with task instructions processed by an LLM. The workflow is divided into Exploration and Deployment phases. This figure also illustrates a specific task scenario where the agent is directed to find and add to the cart an iPhone XS Max priced between $1500 and $2000 in mobile device.

However, their capabilities remain limited when it comes to handling non-textual inputs in real-world applications

In real-world scenarios, many applications demand more than just textual processing, requiring the integration of visual and other data modalities for tasks such as graphical user interface (GUI) recognition and navigation. These requirements highlight the limitations of text-only agents, which struggle with tasks involving visual recognition or multi-step reasoning in environments that rely on rich GUI interactions. Multimodal systems [62, 118, 96] are crucial in complex environments such as mobile and operating system platforms. They need to perform multi-step reasoning, integrate diverse information, and respond adaptively to user inputs. Innovative solutions such as the AppAgent [61] and MobileAgent [119] have shown promise by enabling more natural interactions with smartphone applications through human-like interactions.

Despite these advancements, accurately recognizing graphical user interfaces (GUIs) remains a key challenge, impacting the decision-making accuracy of multimodal agents. Previous approaches [120, 119] that rely primarily on visual features often suffer from inaccuracies, especially in environments with complex or unfamiliar interface elements,

such as video players or game UIs. Additionally, the dynamic nature of mobile environments, which frequently introduce new features, poses further challenges. Even sophisticated models like GPT-4, while proficient with well-known apps, struggle with lesser-known apps due to unfamiliar visual elements. The rapid updates in app interfaces and functionalities further hinder these models' effectiveness across diverse applications.

To address this challenge, AppAgent [61] adopts a human-like approach by automated exploration and watching demos. This strategy allows the agent to store UI element descriptions in a document rather than relying on rigid memorization, thus enhancing decision-making by leveraging contextual understanding. However, AppAgent depends heavily on an off-the-shelf parser to identify UI elements, which restricts the agent's operational flexibility in environments featuring non-standard components such as video players and games. This dependency limits the agent's ability to adapt its actions to unfamiliar or unique interface elements, thereby affecting its overall effectiveness in diverse applications.

To mitigate these limitations, we propose a novel multimodal agent framework designed to adapt to the dynamic mobile environment and diverse applications as shown in Figure 4.1. We develop an extensive action space enabling the agent to interact with a wide variety of elements. This includes not only those elements that can be parsed using a standard parser but also elements and text identified through OCR and detection tools.

Unlike previous work that relied solely on ID matching from parser to retrieve information, our approach incorporates multiple forms of element data. To facilitate access diverse elements, we have designed a structured storage system to construct a knowledge base. Each element within the knowledge base can store different attribute information such as parser details, textual content, and visual descriptions. This system is tailored to organize and store element information in a manner that supports quick retrieval and effective utilization, significantly boosting the agent's ability to perform in novel scenarios.

Following previous work [61], our agent operates in two distinct phases: exploration and deployment. In the exploration phase, our agent autonomously analyzes and documents the functionality of unknown UI elements and applications, tailored to specific task types. This proactive documentation allows the agent to build a robust knowledge base of UI layouts and operations, vital for handling tasks in unfamiliar environments. During this phase, we also incorporate a reflection module, which serves to validate the documented functionalities based on iterative assessments, ensuring the accuracy and reliability of the information stored. In the deployment phase, the agent leverages RAG technology [121] to dynamically access and update its knowledge base with relevant document content based on real-time interactions, significantly enhancing its capability to adapt to novel scenarios.

This framework not only streamlines the learning process but also enhances the agent's decision-making capabilities by providing a deeper understanding of each application's functionality.

We validated the agent's performance through extensive quantitative experiments across multiple benchmarks. The results demonstrate that the proposed framework significantly enhances task performance, with improved adaptability and precision across diverse mobile applications. Furthermore, a user study was conducted to evaluate the impact of both agent-driven and manual exploration phases. The results indicate that both methods lead to increased task success rates and user satisfaction, with manual exploration offering the most robust outcomes.

In summary, this paper makes the following contributions:

- We introduce a multimodal agent framework that combines parser with visual features to construct a flexible action space, enhancing interaction with GUI and improving adaptability to new environmental tasks.
- We develop a new structured storage format that, coupled with RAG technology, allows for adaptive, real-time updates and access to the knowledge base, enhancing the agent's adaptability and decision-making precision.
- We conduct extensive quantitative tests and a user study, demonstrating the agent's effectiveness across a variety of smartphone applications, validating its adaptability, user-friendliness, and efficiency in real-world scenarios.

4.2 Related works

4.2.1 LLM-based agents

Agents have rapidly evolved with the advancement of large language models. Models such as MetaGPT [122], HuggingGPT [64], and AssistGPT [62], Seeclick [123], ResponsibleTA [124] have demonstrated exceptional performance in agent applications, garnering widespread adoption across various domains. Some agents employ large language models such as ChatGPT [85] or GPT-4 [4] for task decision-making, achieving notable developments in general domains including music [125, 126], gaming [75, 76], and autonomous driving [81, 83, 82]. Other agents utilize popular open-source models like LLaMA [127] and LLaVA [12]. Meanwhile, agents have achieved significant breakthroughs in the multimodal, including video understanding [63, 62, 70], embodied AI [67, 68], and visual generation [128, 129,

130]. Additionally, there has been a rise in multi-agent cooperative systems [68, 131] where different agents assume distinct roles. This collaborative approach significantly enhances the capabilities of individual agents, thereby facilitating the achievement of ultimate objectives.

4.2.2 Agent for mobile devices

Some agents [58, 59, 60, 132, 133] attempt to simulate human users by directly interacting with GUI elements, but these agents typically require human instructions and guidance to complete tasks. In contrast, LLM-based agents [66, 65, 61], with their advanced comprehension and reasoning capabilities, can more effectively automate such tasks. There are already several agents developed for mobile devices that utilize large language models effectively. DroidBot-GPT [65] automates Android app interactions by interpreting app GUI states and actions into natural language prompts, thus facilitating action selection. AppAgent [61] identifies and enumerates UI components based on XML, subsequently making decisions and executing actions with the aid of GPT-4V. MobileAgent [119] incorporates visual features, integrating OCR technology and icon detection to enhance UI recognition capabilities. AutoDroid [65] seamlessly combines large language models with dynamic app analysis to optimize mobile task automation efficiently. MobileGPT [134], an innovative mobile task automator powered by LLMs, is equipped with a human-like app memory system. This system aids in precise task learning and adaptation by structuring procedures into modular sub-tasks, thereby enhancing the performance and flexibility of mobile agents.

4.3 Method

In this section, we provide a detailed description of our multimodal agent framework, which is structured into two primary phases: exploration and deployment. At each round, the agent analyzes the current GUI with task requirements, generating observations, thoughts, actions, and summaries. The task execution prompt, as shown in Figure 4.2, is designed to guide the agent in this process. . The summary, serving as memory, is carried over to the next execution prompt, ensuring continuity throughout the task execution process.

4.3.1 Agent Framework

Our multimodal agent framework is implemented within the Android 15 environment using the Android Studio emulator. The agent interacts with the mobile device by invoking commands through the `AndroidController`. This interaction is driven by a comprehensive

Prompt for task execution

You are an agent trained to perform basic tasks on a smartphone. When given a smartphone screenshot along with reference documents, your primary directive is to derive actionable insights from the documentation provided. These documents are essential for understanding the functionalities of UI elements that may not be immediately apparent from the screenshot alone. Your actions should be primarily informed by these documents, with the current UI interface analysis serving as a secondary reference.
Your decision-making process should prioritize actions as follows:
[Special requirements of decision-making.]

You can call the following functions to control the smartphone:
[Detailed action space and examples, including tapbutton, text, etc.]

<ui_document>
The task you need to complete is to **<task_description>.** Your past actions to proceed with this task are summarized as follows: **<last_act>**
Now, given the documentation and the following labeled screenshot, you need to think and call the function needed to proceed with the task. Your output should include three parts in the given format:
Observation: <Describe what you observe in the image>
Thought: <To complete the given task, what is the next step I should do>
Action: <The function call with the correct parameters to proceed with the task. If you believe the task is completed or there is nothing to be done, you should output **FINISH**. You cannot output anything else except a function call or FINISH in this field.>
Summary: <Summarize your past actions along with your latest action in one or two sentences. Do not include the numeric tag in your summary>
You can only take one action at a time, so please directly call the function.

Figure 4.2: Prompt of the task execution used by the agent during the task execution process.

analysis of the current GUI, leveraging structured data parsing, Optical Character Recognition (OCR), and detection models to extract detailed information from screenshots. The extracted data includes Android IDs, numerical labels on the screenshots, element features, textual content, and the coordinates of UI elements. This robust setup enables the agent to operate efficiently in dynamic mobile environments, integrating advanced recognition capabilities with intelligent decision-making processes based on the interpreted data from the user interface.

Our framework is designed with flexibility in mind, allowing it to accommodate various LLMs depending on the task requirements. For our experiments, we selected GPT-4V [4], recognized as one of the best-performing multimodal LLMs available. This choice was driven by GPT-4V's superior ability to process and integrate multimodal data, making it particularly well-suited for handling the complex tasks typically encountered in mobile environments. Additionally, the framework supports more cost-effective models like Qwen-VL [135], which, despite being open-source and free, still provides strong performance in most mobile tasks. This makes Qwen-VL a practical alternative when cost efficiency is a priority.

Preliminary runtime profiling was conducted to evaluate the computational overhead of the proposed framework. On a Windows desktop equipped with an Intel Core i7 CPU,

an NVIDIA RTX 3060 Ti GPU, and 32GB of RAM, completing a typical task involving approximately 20 interaction steps required about 2 minutes, averaging roughly 6 seconds per step.

The majority of this runtime is attributed to two main factors: (i) the latency of GPT-4V inference via remote API, which is sensitive to internet connection quality and server response time, and (ii) the execution overhead of the Android emulator itself. In contrast, local processing components such as OCR and UI parsing contribute minimally to the overall latency.

While this setup remains sufficient for development and controlled evaluation, it poses limitations for real-time deployment. In future work, we plan to explore local deployment of lightweight vision-language models such as Qwen-VL to significantly reduce inference latency while maintaining strong multimodal reasoning capabilities. This shift also offers benefits in terms of network robustness, privacy, and offline accessibility.

4.3.2 Interaction Commands and Actions

The agent's interaction with the Android environment is central to its task automation capabilities. During both the exploration and execution phases, the agent translates human commands or outputs from large language models (LLMs) into precise instructions that the Android system can recognize and execute. This interaction is facilitated through a set of well-defined commands, designed to translate high-level tasks into specific actions, enabling the agent to efficiently navigate and interact with various UI elements. The primary commands are as follows:

1. **TapButton:** Initiates a tap on a user interface element. The target can be specified by its numerical identifier or visual features. For example, `TapButton(5)` targets the UI element labeled '5', while `TapButton('hat')` targets the element with the text 'hat'.

2. **Text:** Simulates typing by entering a string into a designated input area, essential for tasks like form filling or chat input. For instance, `Text("Hello, world!")` inputs "Hello, world!" into the specified text field.

3. **LongPress:** Applies a prolonged press on a specified element, often used for actions requiring sustained pressure, such as dragging or accessing context menus. For example, `LongPress(3)` applies a long press to the element labeled '3'.

Prompt for action space

#A prompt example for action space, take 'tap button' as an example.
I will give you the screenshot of a mobile app before and after tapping the UI element labeled with the button **<ui_element>** on the screen. The numeric tag of each element is located at the center of the element.

Tapping this UI element is a necessary part of proceeding with a larger task, which is to **<task_desc>.**

Your task is to describe the functionality of the UI element concisely in one or two sentences. Notice that your description of the UI element should focus on the general function. For example, if the UI element is used to navigate to the chat window with John, your description should not include the name of the specific person. Just say: "Tapping this area will navigate the user to the chat window".

Never include the numeric tag of the UI element in your description. You can use pronouns such as "the UI element" to refer to the element.

Figure 4.3: Prompt of **tap_button** in action space for function generation in agent.

4. **Swipe:** Executes a swipe in a specified direction on an element, useful for scrolling through content vertically or horizontally. For instance, `Swipe(21, "up", "medium")` swipes up on element '21' over a medium distance.

5. **Back:** Simulates the device's back button, allowing the agent to navigate to the previous UI state without directly interacting with specific elements. This is particularly useful for handling back navigation across different applications.

6. **Home:** Returns the agent to the main screen, crucial for resetting the environment, executing cross-application tasks, or restarting tasks from the home screen.

7. **Wait:** Pauses the operation, allowing the system to process tasks or refresh the screen. A typical implementation involves a two-second pause.

8. **Stop:** Signals the completion of tasks and ends the current operation, ensuring that no residual processes remain running.

These commands are executed by the Android system through the `AndroidController`, ensuring accurate and efficient task execution, and allowing the agent to operate seamlessly within the Android environment.

4.3.3 Exploration Phase

The exploration phase is aimed at analyzing the GUI in relation to the current task. It involves identifying and documenting the functions of UI elements through two alternative methods: agent-driven and manual exploration. All prompts used are displayed in Appendix.

4.3.3.1 Agent-Driven exploration

This method starts with the agent analyzing the current UI interface to identify elements requiring interaction and to determine the specific actions needed. Once these elements and actions are pinpointed, the agent executes the planned actions. Following the execution of action, the agent takes screenshots before and after the interaction to compare and analyze the changes. This comparison allows the agent to record the operational functions of the UI elements and assess the effectiveness of each action taken.

The agent is equipped with specific prompts for recognizing the functionality of UI elements associated with each action it executes. Figure 4.3 illustrates how the agent generates the corresponding operations using the "tap button" as an example from the action space.

Afterwards, the agent enters the reflection phase, where it evaluates the actions performed and their outcomes, adjusting its strategy accordingly. If the agent determines that the executed action is completely irrelevant to the task, it performs a return operation. The irrelevant action is recorded in a *useless_list* and is fed back into the LLM. The reflection phase involves making specific decisions based on the results of the actions:

- **ERROR**: If the decision is "ERROR", the operation is terminated, indicating an unrecoverable issue.
- **INEFFECTIVE**: If the decision is "INEFFECTIVE", the resource ID associated with the action is added to the *useless_list*, and the last action is reset to "None", preventing the same ineffective action from being repeated.
- **BACK or CONTINUE**: If the decision is "BACK" or "CONTINUE", the resource ID is again added to the *useless_list*, and the agent may attempt to return to the previous screen or continue exploring. If "BACK" is chosen, the agent performs a back navigation to correct its course.
- **SUCCESS**: If the decision is "SUCCESS", the agent continues with the task as the action was deemed effective.

If the results of the actions align with the intended user task and prove effective, the relevant UI information is documented and the exploration continues.

This reflection mechanism ensures that only actions aligned with the user's task are considered effective and documented for future retrieval. For example, if an ineffective action is identified, such as an incorrect product search, the agent will adjust its approach in real-time, refining its strategy to avoid repeating the mistake. This method not only enhances

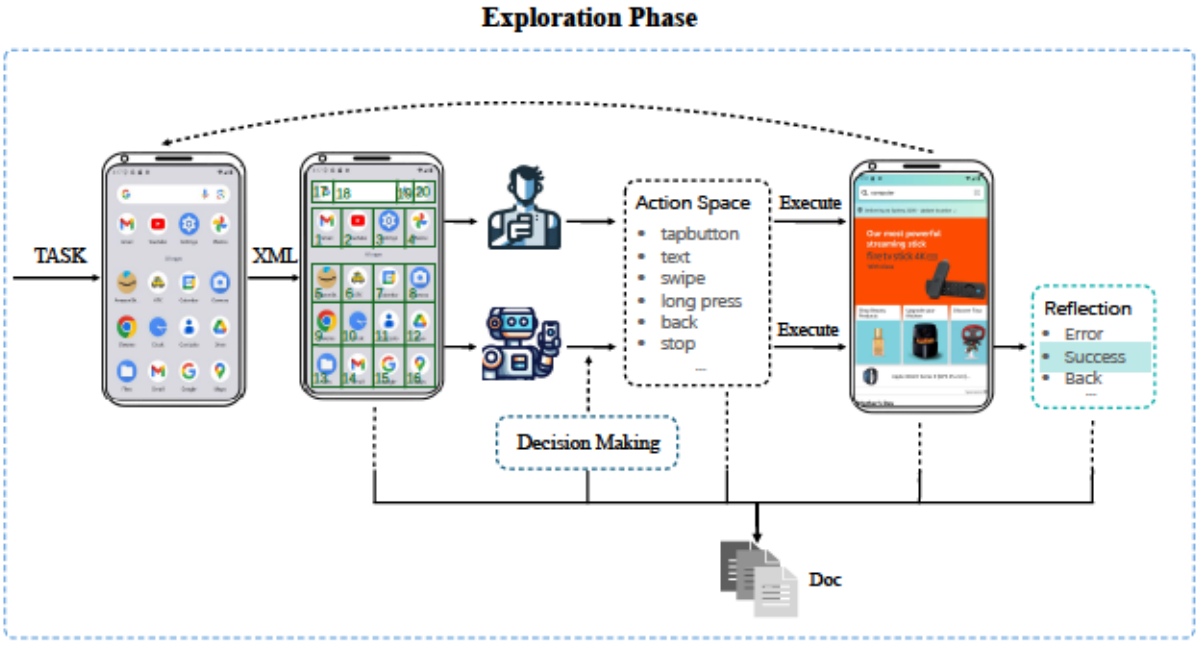

Figure 4.4: Overview of our exploration phase. Exploration module takes agent-driven or manual exploration collects element information into a document.

the quality of the knowledge base but also refines the agent's strategy in real-time, ensuring that subsequent actions are more likely to contribute effectively to task completion.

4.3.3.2 Manual Exploration

This method is introduced to overcome the limitations encountered during agent-driven exploration, such as the LLM's erroneous judgments due to its incomplete understanding of certain apps and UI elements. Manual exploration allows LLM to observe human operations, compare screenshots before and after actions (similar to agent-driven exploration), and gain a clearer understanding of new UI elements and task workflows. The exploration is enhanced with advanced OCR and detection models, providing comprehensive UI analysis based on human interactions. Humans guide the sequence of actions and finalize the process, thereby streamlining the operational workflow and accelerating the learning process.

Importantly, just like in automatic exploration, the information regarding UI elements and their functionalities observed during manual exploration is meticulously documented. This ensures that the agent can overcome the shortcomings of automated processes by incorporating the sophisticated understanding and adjustments that only human insight can provide. Due to the high accuracy of manual exploration, the reflection phase is not required, which further reduces resource consumption.

4.3.4 Document Generation

The document generation process is a critical component of our framework, serving as a specialized knowledge base that underpins the agent's ability to execute tasks with precision. During the exploration phase, the agent systematically collects and records detailed information about the user interface (UI) elements it encounters. This information includes a variety of data points such as Android ID, visible labels, text content, visual features (e.g., color, shape, size), screen coordinates, and the specific functionalities of each UI element as interpreted by LLMs.

This structured metadata format is crucial because it supports dynamic querying based on real-time task demands. When the agent needs to perform a task, it can quickly search through this metadata to find the relevant UI elements and their associated actions. This enables the agent to make informed decisions and execute tasks efficiently, even in complex scenarios.

Additionally, the metadata is designed to be dynamically updated as the agent encounters new UI elements during task execution. This ensures that the knowledge base remains current, allowing the agent to adapt to changes in the UI or new application contexts seamlessly.

To further enhance the retrieval efficiency, we integrate this metadata structure with LangChain's Self-Query Retriever technology [136]. The Self-Query Retriever allows the agent to generate specific queries based on the task at hand, and then search through the vectorized metadata to find the most relevant information. For instance, if the agent is tasked with selecting a specific option on a menu, it can generate a query that matches the text or visual features of the menu item and retrieve the corresponding metadata to guide its actions.

The combination of structured metadata and dynamic updating mechanisms ensures that the agent can respond to user needs with agility and accuracy, making it a powerful tool for task automation in diverse and ever-changing environments.

4.3.5 Deployment Phase

The deployment phase is designed to function independently from the exploration phase, enabling the agent to effectively perform user tasks even without prior exploration. For most tasks, the agent can rely on its pre-existing knowledge and capabilities to directly interact with the GUI, ensuring high accuracy and efficiency. This independent operation allows the agent to execute common tasks swiftly and seamlessly without the need for preliminary

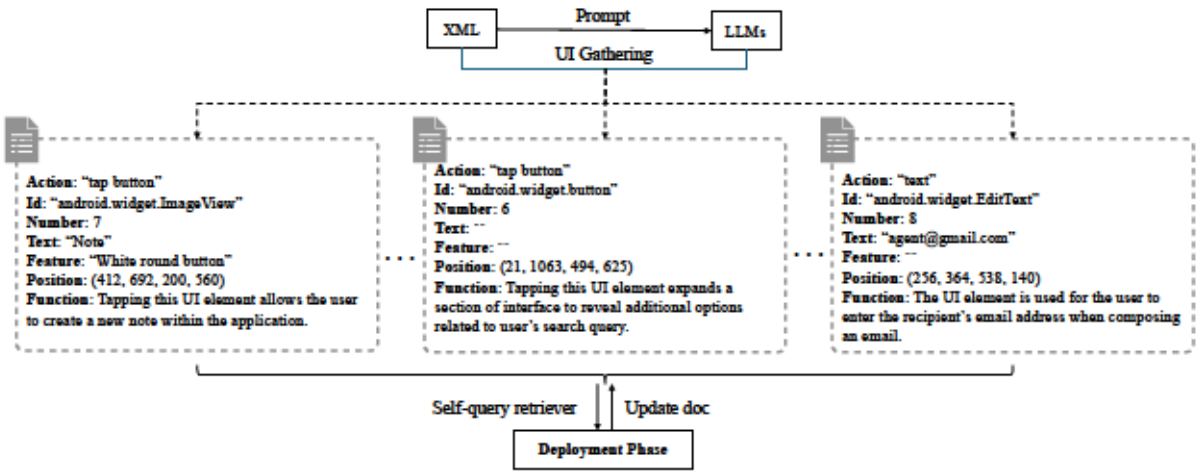

Figure 4.5: Overview of our document generation. During the exploration phase, UI elements are collected and stored as metadata in the document based on specific information. This metadata is then used for retrieval during the deployment phase, with real-time updates synchronized to the document.

data gathering or setup.

In more complex or challenging scenarios, the exploration phase and document retrieval processes become essential for enhancing the agent's performance. By leveraging the knowledge acquired during exploration and accessing detailed information stored in structured documents, the agent can navigate and manage intricate tasks with greater precision, ensuring optimal outcomes in demanding situations.

In cases where the exploration phase has been conducted, the deployment phase is further enhanced by the ability to perform document retrieval operations based on the exploration-generated knowledge. During the deployment phase, the agent first retrieves the current GUI information and systematically analyzes the elements present. If necessary, the agent uses a self-query retriever to access relevant documents stored in a vector-based database. This retriever, built on the data gathered during the exploration phase, converts document content into embeddings, which are then matched with resource IDs or OCR-derived information to retrieve the most relevant data for the current task. For example, if the agent returns the action tapbutton('3'), indicating a tap on the icon labeled '3' in the screenshot, the retriever's query will be transformed into the corresponding "id: resource id<label:3>". This query is then used to perform RAG to extract relevant information about the icon from the document. The retrieved information is subsequently integrated into the prompt, which is fed into the agent for the next round.

Once the appropriate document is retrieved, the agent incorporates this information into its decision-making process. The agent's actions are guided by the current GUI screenshot,

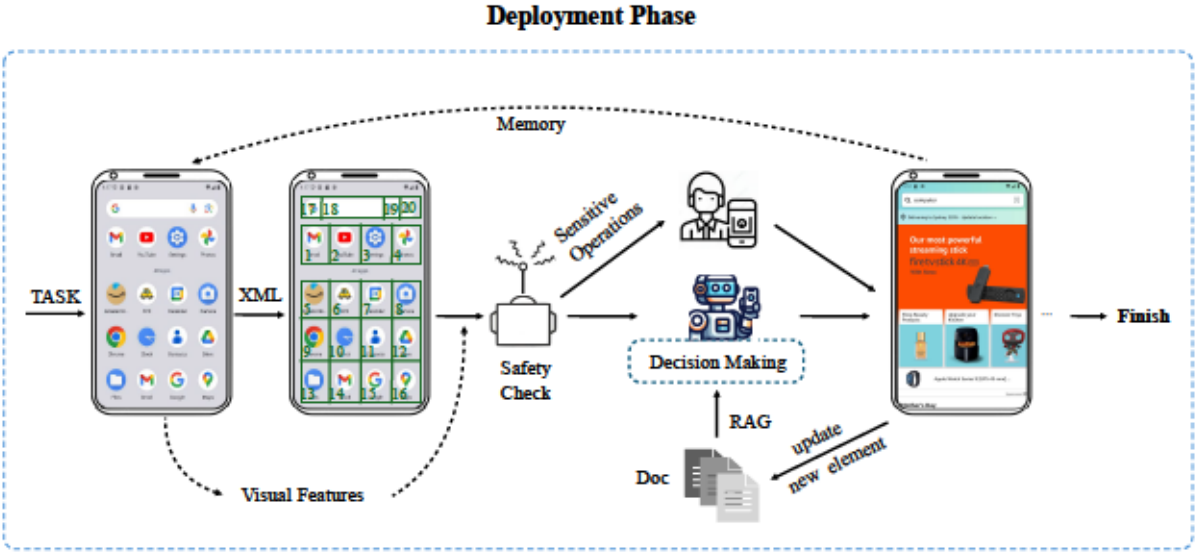

Figure 4.6: Overview of our deployment phase. Deployment phase takes RAG to retrieve and update the document in real time, thereby rapidly preparing to execute tasks.

the retrieved document content, and specific task requirements. This ensures that each step is executed with precision based on the positional and functional data of the UI elements.

For simpler tasks, where document retrieval may not be necessary, the agent can bypass this step and rely solely on its internal knowledge. After each action, the agent updates its memory with historical information and the outcomes of previous actions, continually refining its decision-making for subsequent steps.

This process continues until the agent determines that the task has been completed, at which point it exits the current process and reports task completion. This structured approach ensures that tasks are executed efficiently and accurately, utilizing the detailed knowledge base created during exploration to optimize performance and user satisfaction.

4.3.6 Advanced Features

To further enhance the agent's adaptability and reliability, we incorporated advanced features that address specific challenges in mobile environments. This subsection highlights the key functionalities that enhance our multimodal agent framework, focusing on visual feature decision-making, safety checks, and cross-app task management. These features collectively improve the agent's safety, versatility, and efficiency, ensuring robust performance in complex and dynamic environments.

4.3.6.1 Visual Features Decision-Making

When the agent confronts scenarios where the desired interactive element is not numerically tagged, and other numerically tagged elements are ineffective for task completion, it automatically transitions to an alternative visual feature UI layout. This process leverages advanced OCR technology [137] and detection models [138] to accurately recognize and annotate text and icons within the interface. By numerically annotating these elements using established methodologies, the agent is equipped to make informed decisions based on the newly adapted UI screenshot. This capability is crucial for handling icons in previously unknown scenarios, ensuring that the agent can navigate and interact with various UI elements effectively, regardless of prior exposure. This dynamic decision-making process significantly enhances the agent's ability to adapt to new environments and execute tasks with higher precision and reliability.

4.3.6.2 Safety Check

In modern LLMs and agent systems, safety is crucial, particularly in automated processes that can lead to privacy breaches. To tackle this issue, we implemented a safety check during the deployment phase. The agent reviews the current UI screenshot, and if the next steps involve sensitive actions like account passwords, payment or other privacy-related concerns, it will switch to manual mode so the user can handle these operations personally. For privacy, the agent will not retain any information from this process. Once the user completes the sensitive task and inputs "finish," the agent will automatically continue with the deployment phase and carry on with the task until it's completed. The safety check offers several key advantages. It ensures that sensitive tasks remain secure by involving human judgment and minimizes the risk of data leakage. Furthermore, it increases user trust in the system, providing assurance that private information is handled carefully, while still enabling the agent to effectively complete its assigned tasks. An example is shown in Figure 4.7

4.3.6.3 Cross-Apps Task

In addition to its core functionalities, our framework is capable of handling complex tasks that span multiple applications. This ability allows the agent to seamlessly perform tasks requiring interactions across different interfaces and applications while maintaining continuity and coherence.

When engaging in cross-app tasks, the agent begins by evaluating its progress within the current application. If the agent detects that the current task has been completed and

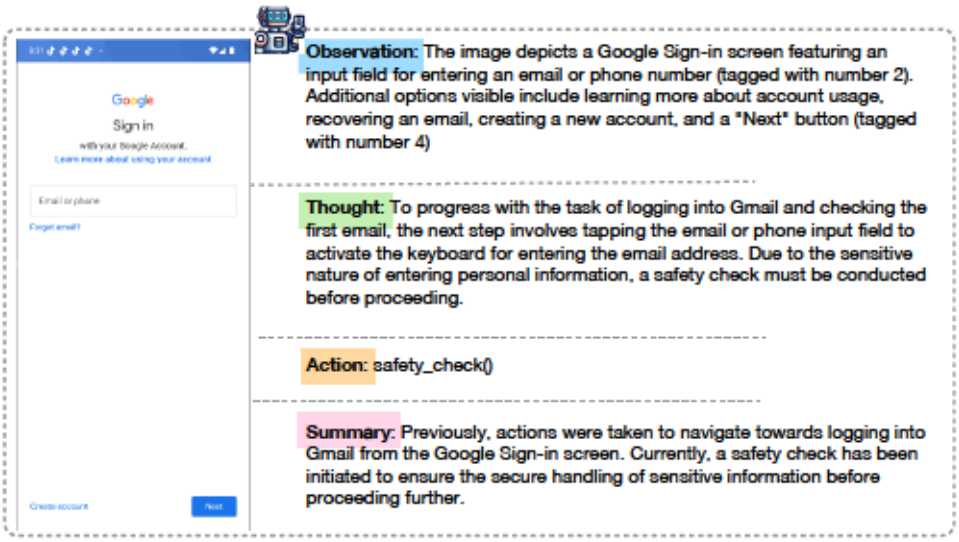

Figure 4.7: An example of safety_check being triggered. During the agent's execution, when encountering an email login interface that requires account and password information, a safety check is automatically triggered, and control is seamlessly switched to manual operation.

requires transitioning to another application, it triggers the Home action to return to the home screen. This step is crucial as it allows the agent to switch to the next application to continue the task execution.

Each round of task execution generates a summary that is stored as part of the agent's memory. This summary not only serves as a record of the actions taken but also enhances the agent's long-term memory. By continuously storing these summaries, the agent retains a clear memory of past instructions and processes, facilitating the seamless continuation of tasks across different applications.

For instance, if the agent needs to gather information from a social media app and then process it in a document editor, it will complete the interactions in the social media app, trigger the Home action to return to the home screen, and then launch the document editor to continue the task. The agent retrieves relevant summaries from its memory, integrates them with the current GUI context, and executes the next set of commands. This approach ensures that the agent maintains a coherent flow of actions, even when switching between applications.

This dynamic updating of memory and the ability to navigate across apps using the Home action are crucial for maintaining context and ensuring that the agent's operations

are synchronized and effective across different digital environments. This capability is particularly valuable for complex tasks that require gathering and processing information from multiple sources or coordinating actions between various applications.

4.4 Experiments

In this section, we will conduct a comprehensive evaluation with our agent framework. The experiments were conducted on the Android platform to maintain consistency and simplify validation. We utilized the Android Studio emulator for the experiments, which included comprehensive testing on the public benchmarks and qualitative results. This dual approach allowed us to benchmark our agent against standardized criteria while also gaining deeper insights into its real-world performance on mobile applications and environments.

4.4.1 Quantitative Results

In this section, we present a comprehensive evaluation of our agent using two distinct benchmarks: DroidTask [139] and Mobile-Eval [119]. We begin with DroidTask to test complex task performance, and conclude with Mobile-Eval to assess comprehensive capabilities. Results in the ensuing sections demonstrate the superiority of our approach in varied application scenarios.

4.4.1.1 DroidTask

In this study, we utilize the DroidTask dataset [139], an Android Task Automation benchmark suite meticulously designed to evaluate the performance of end-to-end mobile task automation systems. DroidTask comprises 158 high-level tasks extracted from 13 widely-used mobile applications, encompassing a broad spectrum of task complexities—from simple single-step operations to intricate multi-step procedures. This diversity ensures that the dataset provides a robust and challenging environment for evaluating the capabilities of task automation systems. Additionally, the DroidTask benchmark provides a reproducible experimental environment by releasing the setup as an Android Virtual Machine Snapshot. This allows researchers to restore the exact environment in which the data was collected, thereby ensuring a high degree of reproducibility and comparability across different studies.

In our experiments, we employed the "Completion Rate" as the primary evaluation metric, following the methodology outlined in [139]. Completion Rate is defined as the

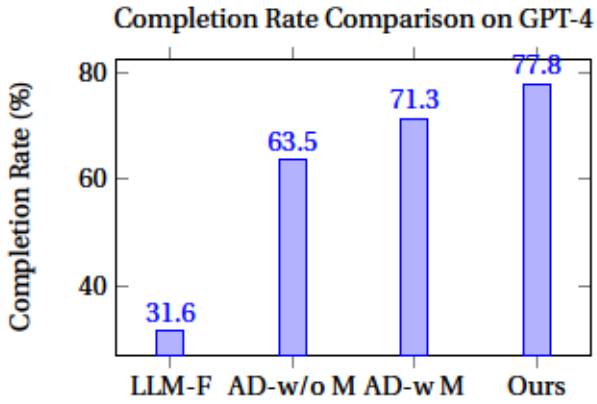

Figure 4.8: Performance Comparison between AutoDroid and ours on DroidTask with GPT-4

probability that an agent accurately completes all actions in a given task sequence, serving as a critical measure of the agent's consistency and effectiveness in task execution.

AutoDroid incorporates a memory mechanism, analogous to the document in our agent. We conducted a comprehensive comparison of our directly deployed method against AutoDroid, both with and without its memory component, as well as against the LLM-powered GPT-4 framework, which served as a robust baseline. The results, as presented in Table 4.8, indicate that our method, **even without leveraging an exploration phase and document retrieval**, significantly outperforms GPT-4 and surpasses AutoDroid, even when it is augmented with memory. This demonstrates the superiority of our approach in effectively leveraging direct deployment strategies and highlights the robustness of our system in a competitive benchmarking environment.

It is important to note that during the evaluation, variations in app versions and device models led to differences in the workflows required to implement specific functionalities within the apps. As a result, a small subset of tasks could not be executed directly. To address this, we devised alternative testing methods for these tasks. For example, in cases where our application lacked a date-sorting option for document names, we opted to sort by the initial letter of the document names as a substitute. This adjustment maintained the same procedural flow and steps, albeit with slight modifications to the final selection criteria. Additionally, tasks that our application does not support and for which no alternative exists were treated as error cases. Under identical conditions, the performance of our agent is expected to be even higher than what was observed, further validating the effectiveness of our direct deployment approach.

Table 4.1: Quantitative results of MobileAgent and ours on Mobile-Eval.

App	INSTRUCTION 1				INSTRUCTION 2				INSTRUCTION 3			
	SU	PS	RE	CR	SU	PS	RE	CR	SU	PS	RE	CR
MobileAgent												
Alibaba.com	✓	0.75	4/3	100%	×	0.39	13/8	62.5%	✓	0.9	10/9	100%
Amazon Music	✓	0.44	9/5	80%	✓	0.75	8/6	100%	×	0.50	12/3	66.7%
Chrome	✓	1.00	4/4	100%	✓	0.80	5/4	100%	✓	0.43	8/5	100%
Gmail	✓	1.00	4/4	100%	×	0.56	9/8	37.5%	×	0.56	9/8	37.5%
Google Maps	✓	1.00	5/5	100%	✓	1.00	6/6	100%	✓	1.00	6/6	100%
Google Play	✓	1.00	3/3	100%	✓	0.50	10/4	100%	✓	1.00	3/3	100%
Notes	×	0.57	7/4	100%	✓	0.67	6/4	100%	✓	1.00	5/5	100%
Settings	✓	1.00	4/4	100%	✓	1.00	4/4	100%	✓	1.00	4/4	100%
TikTok	✓	1.00	4/4	100%	✓	1.00	10/10	100%	✓	1.00	7/7	100%
YouTube	✓	1.00	4/4	100%	✓	1.00	9/9	100%	✓	1.00	7/7	100%
Multi-App	✓	1.00	6/6	100%	✓	1.00	10/10	100%	✓	1.00	10/10	100%
Avg	0.91	0.89	4.9/4.2	98.2%	0.82	0.77	7.9/6.3	90.9%	0.82	0.84	7.5/6.2	91.3%
Ours												
Alibaba.com	✓	1.00	3/3	100%	✓	0.89	9/8	100%	✓	0.82	11/9	100%
Amazon Music	✓	1.00	5/5	100%	✓	1.00	6/6	100%	✓	1.00	3/3	100%
Chrome	✓	1.00	4/4	100%	✓	0.80	5/4	100%	✓	1.00	5/5	100%
Gmail	✓	1.00	4/4	100%	✓	0.80	5/4	100%	✓	1.00	8/8	100%
Google Maps	✓	1.00	5/5	100%	✓	1.00	6/6	100%	✓	1.00	6/6	100%
Google Play	✓	1.00	4/4	100%	✓	1.00	4/4	100%	✓	1.00	4/4	100%
Notes	✓	0.80	5/4	100%	✓	0.80	5/4	100%	✓	0.80	5/4	100%
Settings	✓	1.00	4/4	100%	✓	1.00	4/4	100%	✓	1.00	4/4	100%
TikTok	✓	1.00	4/4	100%	✓	1.00	10/10	100%	✓	1.00	7/7	100%
YouTube	✓	1.00	4/4	100%	✓	1.00	9/9	100%	✓	1.00	7/7	100%
Multi-App	✓	1.00	6/6	100%	✓	0.83	12/10	100%	✓	0.83	12/10	100%
Avg	**1.00**	**0.97**	**4.3/4.2**	**100%**	**1.00**	**0.91**	**6.7/6.3**	**100%**	**1.00**	**0.95**	**6.7/6.2**	**100%**

4.4.1.2 Mobile-Eval

We evaluated our agent using the Mobile-Eval benchmark, which provides a variety of different metrics to test the agent's capabilities across multiple dimensions. This is why we selected it as one of our testing benchmarks. The benchmark includes 10 commonly used mobile applications and tests the agent's performance across various tasks, measuring key metrics such as success rate, process score (PS), relative efficiency (RE), and completion rate (CR). We compared our agent's results against those of Mobile-Agent and human performance to assess its effectiveness in real-world scenarios. Mobile-Eval assesses the following metrics:

- **Success (Su):** Marks an instruction as successful if the agent completes it entirely.
- **Process Score (PS):** Evaluates step accuracy by calculating the ratio of correct steps to

total steps.

- **Relative Efficiency (RE):** Compares the steps taken by the agent to human performance to measure efficiency.

- **Completion Rate (CR):** Measures the proportion of steps the agent completes compared to a human's total steps.

As shown in Table 4.1, the results highlight the robust performance of our agent compared to Mobile-Agent. Our agent achieved a 100% success rate across all instructions, demonstrating its ability to accurately complete tasks without errors. This is particularly evident in the consistent high PS scores, with averages exceeding 90% across the three instruction sets. This indicates that our agent not only successfully completed tasks but did so with high precision, closely mirroring human task execution.

In terms of relative efficiency (RE), our agent consistently matched or exceeded the efficiency of Mobile-Agent. This is crucial as it suggests that our agent can complete tasks with fewer or equal steps compared to human benchmarks, reflecting an optimized decision-making process during task execution. For example, in more complex applications like Gmail and Google Maps, our agent maintained a 100% completion rate with high efficiency, demonstrating its ability to handle sophisticated tasks that require multiple steps and decisions.

Furthermore, the comparison between our agent and Mobile-Agent reveals that our agent's performance is particularly strong in scenarios involving abstract or less-defined instructions, where it must rely on its understanding and adaptability. This capability is essential for real-world applications where user instructions may be vague or open-ended.

Overall, the results from the Mobile-Eval benchmark underscore the effectiveness of our agent in a diverse set of tasks and applications. Its ability to consistently perform at a high level across different metrics not only validates the robustness of our method but also highlights its potential for broader deployment in various mobile environments.

4.4.2 Qualitative results

To validate the qualitative performance of our agent, we conducted a detailed qualitative result, as illustrated in Figure **??**. The study was designed to assess the agent's ability to handle a series of complex, real-world tasks that require not only basic interaction with a mobile interface but also sophisticated multi-step processes and cross-application operations.

In this study, agent was tasked with checking an unread message from a messaging application and replying by sharing a video from YouTube. This task involves multiple steps, including identifying and accessing the appropriate application, retrieving specific content, and managing the transfer of information between applications. The agent's ability to retain and utilize memory across these steps was crucial for successful task completion.

Figure 4.9 outlines critical steps in the process, highlighting the agent's observations, thoughts, actions, and summaries at each stage. Notably, the agent demonstrated the capacity to:

Navigate Cross-Application Activities: The agent seamlessly transitioned between different applications, identifying the necessary elements in each interface and executing the required actions without manual intervention.

Manage Long-Term Multi-Step Tasks: Throughout the task, the agent effectively managed a sequence of actions that spanned multiple steps, demonstrating the ability to maintain task continuity and accurately execute each step based on previous interactions.

Utilize Multi-Step Memory Storage: The agent leveraged memory from earlier steps to inform subsequent actions, ensuring that the task was completed efficiently and without redundancy.

Each step in the task was meticulously recorded, showing how the agent's thought process evolved based on the information gathered from the GUI at each stage. This systematic approach allowed the agent to make informed decisions, reflecting a deep understanding of the task requirements and the operational context.

Overall, the result confirms that our agent is not only capable of handling simple tasks but also excels in more complex scenarios that demand advanced problem-solving skills and the ability to work across multiple applications in a cohesive manner. Further case studies and detailed analyses of the agent's performance are provided in the Supplymentary materials.

4.4.3 Analysis of UI Interface Parsing

In our agent, we employ two primary methods for parsing UI interfaces: structured data and visual features. Structured data provides precise and rich information, including details about widget interactivity—such as clickability and scrollability. In this experiment, we utilized XML data parsed from Android systems to enhance our understanding and manipulation of these interactive elements. This method is well-suited for most generic apps and, in conjunction with our agent, can complete the majority of tasks efficiently.

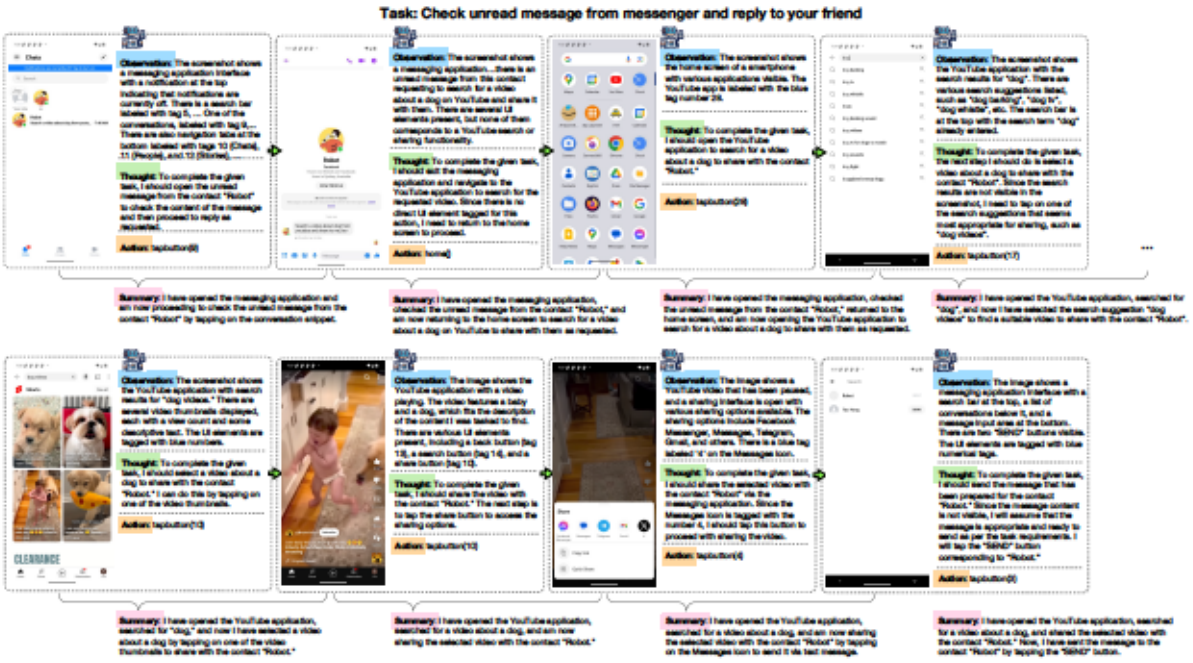

Figure 4.9: Qualitative results of a cross-app task.

Nevertheless, there are challenges associated with mobile platforms that feature custom-developed apps and icons. Specifically, structured data cannot be parsed for custom icons built on Android, which necessitates the use of visual features for extracting widget information. This approach allows for more accurate recognition of text and icons. However, visual features alone cannot determine the operability of icons without direct interaction, which may lead to redundant operations, such as the agent attempting to interact with non-interactive elements.

Therefore, in our agent, visual feature analysis serves as a secondary operation. It is only employed when the agent determines that no XML-based icons can perform the required task. This strategy enhances the robustness of our agent and improves its transferability to novel apps.

4.5 User study

To comprehensively evaluate the adaptability and performance of our agent, we conducted a user study using the AppAgent benchmark [61]. This benchmark comprises ten popular applications, each selected for its unique functionality and user interface challenges. The applications were chosen to simulate a variety of real-world mobile tasks, covering categories such as social media, communication platforms, multimedia, and productivity tools, thereby

providing a diverse and representative testbed for our agent.

4.5.1 Participants

We recruited 15 participants (9 male, 6 female), all of whom were university students, including undergraduates, master's, and PhD students, aged 18–27, with varying levels of experience in using mobile applications. Participants were randomly divided into three groups, with 5 participants in each group. Each group was assigned a different exploration strategy for completing the tasks, as detailed in the following sections. All participants were asked to complete a series of tasks within each application, guided by these instructions.

4.5.2 Environment

The study was conducted using Android Studio emulators across three separate computers, all configured to run Android 12.0 ("S") on a "Medium Phone API 31" setup. This setup ensured uniformity across all participants, with each computer running the same configuration. By maintaining a consistent environment, we eliminated potential variability caused by hardware differences, ensuring that all participants operated under identical conditions. Additionally, all applications were updated to their latest versions to guarantee consistency in testing parameters.

4.5.3 Procedure

Participants were randomly divided into three groups of 5. Each group was assigned a different exploration approach:

- **Group A**: Directly entered the deployment phase, executing tasks based on predefined instructions without prior exploration.
- **Group B**: Underwent an agent-driven exploration phase, where the agent autonomously analyzed the interface, identified key elements, and documented them. This documentation was then used in the deployment phase to assist task execution.
- **Group C**: Performed a manual exploration phase, where participants manually interacted with the interface, while the agent automatically documented relevant UI elements. This documentation was similarly used during the deployment phase.

In the deployment phase, participants completed tasks in the same set of applications. Group A relied on the agent's internal knowledge to execute tasks, while Groups B and C

used the documentation generated during the exploration phase. In complex tasks, the agent utilized a self-query retriever to retrieve relevant UI information. Success rate (SR) and error rate were recorded for each group, followed by a post-task questionnaire to assess usability and task difficulty.

4.5.4 Results

The results, summarized in Table 4.2, reveal that the exploration phase significantly enhanced task performance. Group A, which performed no prior exploration, achieved a average success rate (SR) of 84.4%, comparable to the AppAgent with Watching Demos. Group B, using agent-driven exploration, achieved a higher average SR of 87.4%, while Group C, using manual exploration, recorded the highest average SR at 93.3%. These results indicate that both agent-driven and manual exploration phases contribute to improved task performance, with manual exploration yielding the most robust results.

Table 4.2: Quantitative results between AppAgent and ours.

Method	Document	Action Space	SR (%)
GPT4 (Baseline)	None	Raw	2.2
	None	AppAgent	48.9
AppAgent	Auto. Exploration	AppAgent	73.3
	Watching Demos	AppAgent	84.4
Ours	None	Ours	84.4
	Agent-Driven	Ours	87.4
	Manual	Ours	**93.3**

4.5.5 RQ2 Revisited: Optimizing Efficiency for Dynamic and Resource Constrained Environments

This work addresses RQ2 by proposing a lightweight, modular mobile agent framework, that enables multimodal LLMs to operate efficiently in dynamic and resource-constrained settings. Rather than relying on heavy model retraining, our system decouples exploration and execution through a two-stage architecture and integrates structured memory with visual parsing, allowing for dynamic adaptation to novel interfaces and user-defined tasks.

Key design strategies, such as modular perception pipelines, flexible action space and retrieval-augmented reasoning, significantly reduce computational overhead while maintaining task flexibility. The agent demonstrates the ability to generalize to unseen applications, interpret unfamiliar controls, and execute complex workflows, all within constrained computational budgets.

These results confirm that architectural modularity and memory-aware adaptation can effectively address the efficiency and adaptability challenges of multimodal LLMs in real-world mobile and interactive scenarios, thereby providing a direct response to RQ2.

4.6 Limitations

Throughout the comprehensive testing process, we identified several limitations of our agent: Our method relies on the agent's ability to recognize numerical tags on the UI to determine specific UI elements. This approach can lead to confusion when the UI element itself contains numbers. Such errors can be mitigated through preliminary manual exploration and documentation to clarify the context.

When attempting to interact with hidden UI elements, such as accelerating a video by clicking on the screen, the agent lacks the necessary prior knowledge and cannot detect the acceleration button within the current UI. This limitation hampers its ability to perform specific operations. Future work will focus on enhancing UI recognition and incorporating prior knowledge to address these issues effectively.

4.7 Conclusion

This paper introduces a multimodal agent framework that significantly enhances the interaction capabilities of smartphone applications. Our experiments across various applications demonstrate the framework's ability to improve GUI recognition and task execution, confirming its effectiveness in adapting to diverse application environments.

We integrate parsers with visual features to construct a more flexible action space and develop a newly structured knowledge base for diverse element storage. Through two phases, exploration and deployment, we enable the agent to effectively manage the dynamic nature of mobile interfaces. These capabilities not only align with but also extend the current research on intelligent agents, especially in the contexts of multimodality and mobility.

The user study further underscores the practical effectiveness of our framework, revealing that both agent-driven and manual explorations significantly enhance task performance and user satisfaction. Participants in the user study exhibited higher success rates and reported greater satisfaction when using our agent, particularly those who engaged in manual exploration, highlighting the value of interactive and user-informed approaches in complex application scenarios.

Moving forward, we aim to expand the agent's capabilities to facilitate cross-application functionalities and refine its decision-making processes. This will involve addressing the challenges identified in both the experiments and the user study, such as optimizing the agent's adaptability to less familiar applications and enhancing its efficiency in real-world tasks. Ultimately, our goal is to not only improve the efficiency but also the user experience, making intelligent agents more intuitive and effective across a broader spectrum of mobile applications.

CHAPTER 5

WHO CAN WITHSTAND CHAT-AUDIO ATTACKS? AN EVALUATION BENCHMARK FOR LARGE LANGUAGE MODELS

Adversarial audio attacks pose a significant threat to the growing use of large language models (LLMs) in voice-based human-machine interactions. While existing research has primarily focused on model-specific adversarial methods, real-world applications demand a more generalizable and universal approach to audio adversarial attacks. In this paper, we introduce the Chat-Audio Attacks (CAA) benchmark including four distinct types of audio attacks, which aims to explore the the vulnerabilities of LLMs to these audio attacks in conversational scenarios. To evaluate the robustness of LLMs, we propose three evaluation strategies: Standard Evaluation, utilizing traditional metrics to quantify model performance under attacks; GPT-4o-Based Evaluation, which simulates real-world conversational complexities; and Human Evaluation, offering insights into user perception and trust. We evaluate six state-of-the-art LLMs with voice interaction capabilities, including Gemini-1.5-Pro, GPT-4o, and others, using three distinct evaluation methods on the CAA benchmark. Our comprehensive analysis reveals the impact of four types of audio attacks on the performance of these models, demonstrating that GPT-4o exhibits the highest level of resilience.

5.1 Introduction

Large language models (LLMs) capable of processing text, images, and audio have become increasingly essential for applications that require advanced comprehension and response generation, including customer service [140, 141], automated content creation [142, 143], and interactive media [144, 145]. However, the versatile capabilities of these models also increase their vulnerability to adversarial attacks [146, 147]. This is particularly true in the domain of LLM-driven human-machine voice interaction, where the emergence of such services has accelerated research into audio-based adversarial attacks and defense mechanisms.

Attacks on multimodal audio LLMs can cause the models to produce unintended outputs. However, this area has received limited attention, primarily due to the challenges associated with audio as an input modality. Unlike images, audio lacks direct gradient signals, making the crafting of adversarial examples more complex. Previous research on adversarial audio attacks has predominantly focused on targeted attacks [148, 149, 150], where carefully crafted perturbations are embedded within speech signals. While these samples can effective in misleading models, often appear as random noise and are easily detectable by human listeners. A notable advancement [151] introduced a gradient-based optimization approach that utilizes the Connectionist Temporal Classification (CTC) loss [152]—a method designed for time series data in classification tasks. However, this method remains model-specific and lacks broader generalizability. Universal adversarial audio attacks [153] are highly relevant to real-world attack scenarios, such as when a speaker makes a verbal error or when they are speaking in a noisy environment. Attackers can pre-design and generate these universal attacks in advance, then apply them to any input audio. Despite their relevance, there has been insufficient exploration of their impact on multimodal audio LLMs.

As multimodal audio LLMs become more prevalent in human-machine voice interactions, the threat posed by these attacks grows significantly. To explore the vulnerabilities of LLMs to adversarial audio attacks, we propose a benchmark of universal adversarial audio attacks specifically based on conversational scenarios, named Chat-Audio Attacks (CAA). The CAA benchmark consists of 360 adversarial audio attack sets, with each set encompassing four distinct types of audio attacks: content attack, emotional attack, explicit noise attack, and implicit noise attack. This results in a total of 1,680 adversarial audio samples. We believe that CAA benchmark will not only enable researchers to pinpoint weaknesses in LLMs under adversarial audio conditions but also drive the advancement of robust defense mechanisms for LLM.

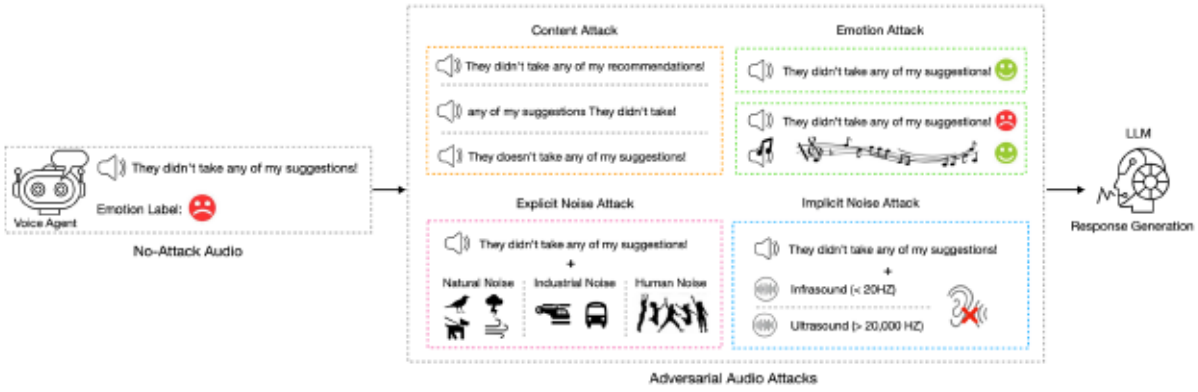

Figure 5.1: An overview of Chat-Audio Attacks (CAA) benchmark including four distinct types of audio attacks.

In addition, we introduce three evaluation methods to comprehensively assess the resilience of LLMs against adversarial audio attacks: Standard Evaluation, GPT-4o-Based Evaluation, and Human Evaluation. The Standard Evaluation uses rigorous metrics to quantify the accuracy, similarity, and consistency of voice responses under adversarial conditions, providing a repeatable and controlled result. In contrast, the GPT-4o-Based Evaluation simulates real-world interactions, capturing complex, sensitive inaccuracies that standard metrics might overlook. Human Evaluation reflects actual user experience and perceptions, thus offering crucial insights into user trust.

Finally, we evaluate six state-of-the-art LLMs supporting voice-based conversations on the CAA benchmark, such as Gemini-1.5-Pro [72] and GPT-4o [85], providing results across the three aforementioned evaluation methods. We analyzed the impact of four types of audio attacks on the LLMs and discussed the flaws these models exhibit in the face of such attacks.

The main contributions of this work are summarised as:

- We propose a benchmark for universal adversarial audio attacks based on conversation task, called Chat-Audio Attacks (CAA).

- We propose three evaluation methods to systematically evaluate the performance of LLMs against adversarial audio attacks.

- We perform a comprehensive evaluation of six state-of-the-art LLMs using the CAA benchmark. Based on the three experimental results, we provide an in-depth analysis and discussion of the results.

5.2 Related works

5.2.1 Audio/Speech Language Models

In the field of audio-based language models, initial systems [154, 56, 155] utilized either acoustic or semantic tokens to enable generation from audio inputs into text or audio outputs. With the technological advancements brought by large language models (LLMs), the recent trend has shifted towards multimodal models [57, 156, 157, 158] are leveraging the combined strengths of both speech and text modalities, substantially enhancing the versatility and effectiveness of audio-based applications.

Models like SpeechGPT [71] utilize a cross-modal architecture that aligns speech and text for tasks such as instruction following and spoken dialogue. SALMONN [57] introduces dual encoders to process diverse audio inputs, excelling in speech recognition and even audio storytelling. Qwen2-Audio [19], LLama-Omni [159], and Gemini-1.5-pro [72] each contribute unique capabilities ranging from voice chat and low-latency interactions to handling complex multimodal data. Additionally, GPT-4o [160] expands upon these functionalities by ensuring robust performance in audio-text interactions within noisy environments, marking a significant milestone in the field.

5.2.2 Audio Attacks

In the domain of adversarial attacks, the concept was first pioneered in the field of image processing [17], where slight perturbations to input pixels [16] could mislead traditional neural network models [161] into producing incorrect results. This methodological foundation laid the groundwork for similar explorations in the audio domain, particularly targeting systems such as automatic speech recognition (ASR) [151, 162] and spoofing/automatic speaker verification (ASV) [163, 149, 150], where security and reliability are critical.

The initial generation of adversarial samples utilized optimization methods first developed for music genre classification [164]. These techniques manipulated entire audio waveforms to avoid detection, altering not only specific acoustic features but the entire sound profile while preserving perceptual quality. In contrast, in the field of speech paralinguistics [148, 149, 150], the Fast Gradient Sign Method (FGSM) has been employed to craft adversarial samples aimed at disrupting systems. Large language models (LLMs) that process diverse data types such as text, images, and audio offer enhanced capabilities for generating human-like responses across various applications. However, their multi-modal nature also increases vulnerability to jailbreaks [165] and adversarial attacks, with poten-

tial exploits spanning across all processed modalities, allowing attackers to bypass safety constraints embedded within these models.

5.3 CAA Benchmark

In CAA benchmark, we target the response generation task by collecting audio suitable for human-machine chat. The overview of CAA benchmark is shown in Figure 5.1. Each set of audio attack data consists of a quadruplet (a_i, t_i, ${a_i}^{no\_attack}$ $\mathscr{A}_i$), where a_i represents the original, unprocessed audio containing a single utterance; t_i refers to the transcript of the original audio along with other associated textual labels; ${a_i}^{no\_attack}$ indicates the audio generated by a voice agent reading the transcript without any attack; and the set $\mathscr{A}_i$ includes 3 or 5 types of attack variations of the audio.

5.3.1 Audio Collection

For the unprocessed audio a_i and corresponding transcripts t_i in CAA benchmark, we manually collected data from three publicly available multimodal datasets (text, audio, and visual): MELD, TVQA, and Common Voice.

- MELD (Multimodal EmotionLines Dataset) [20]: is designed for emotion recognition and classification, derived from the popular TV show Friends. MELD contains numerous dialogue examples, each associated with audio, video, transcripts, and emotion labels (e.g., happiness, sadness, anger, etc.).

- TVQA [166]: primarily focused on understanding video content and associated dialogues in television shows, this dataset covers six famous English-language TV series. Each dialogue instance includes audio, video frames, and transcripts.

- Common Voice [21]: is a multilingual dataset for speech recognition, it provides audios and transcripts. However, the audio samples are not explicitly designed in a dialogue format.

After manually filtering and applying GPT-4 [85] refinement, we collected 120 English speech samples along with their transcriptions from each dataset mentioned above. Notably, the emotional tags from the MELD dataset were also collected to facilitate the generation of emotional attacks in subsequent experiments.

5.3.2 Audio Attack Generation

We processed the collected audio samples to generate five distinct types of audio variations: no attack, content attack, emotional attack, explicit noise attack, and implicit noise attack.

No-Attack Audio refers to audio generated by a voice agent reading the transcript without any modifications or interference. In CAA benchmark, we utilized AzureSpeechSDK agent [167] to produce audio recordings. Specifically, for samples sourced from MELD, which include emotion labels, AzureSpeechSDK agent was configured to match the emotional tone indicated by the labels. For TVQA and Common Voice samples, where emotion labels are absent, the agent was instructed to adopt a neutral tone.

We observed that some samples from MELD, TVQA, and Common Voice are often impacted by factors such as speech rate, accent, and clarity, which can obscure the audio information, making them unsuitable as baselines for subsequent comparison and analysis. To address this, we generated no-attack audio to ensure that the LLMs receive clear speech inputs. This serves as a baseline, offering audio free from any interference or alterations.

Content Attack alters a small fraction of the audio's transcribed tokens while preserving the overall semantic meaning. Inspired by these studies [168, 169, 170, 171], we modified the transcriptions using one of the following strategies: (1) synonym substitution, (2) token rearrangement, or (3) minimal token variation. For synonym substitution, we employed GPT-4 to identify key tokens and replace them with synonyms. For example, "They didn't take any of my suggestions" was altered to "They didn't take any of my recommendations!". Minimal token variation involved altering non-essential tokens, such as "didn't" to "doesn't". The modified text was then read aloud by the AzureSpeechSDK agent, preserving the original emotional tone, resulting in content-attacked audio.

The goal of content attacks is to explore whether LLMs are sensitive to token changes or minor errors when the overall meaning of the audio remains preserved.

Emotional Attack alters the emotional tone of the audio without changing the content. CAA benchmark contains two types of emotional attacks: (1) opposing emotional tone, and (2) opposing emotional background music. In the first scenario, the AzureSpeechSDK agent was instructed to re-read the transcript with an emotion opposite to the original. For instance, if the original sample had an "angry" emotion label, the agent would re-read the transcripts with a "happy" tone, generating an opposite-emotion audio sample. In the second scenario, we overlaid background music with an opposing emotion onto the no-attack audio, and adjusted the music volume to ensure the speaker's voice remained clear. It is important to emphasize that only samples from the MELD dataset in our collection are labeled with emotions. As a result, we utilized 120 samples from MELD to generate two emotional attack

audio samples for each, resulting in two distinct emotional tones per sample.

The objective of emotional attacks is to investigate the sensitivity of LLMs to variations in speech emotion and whether a mismatch between speech content and emotional tone influences the responses.

Explicit Noise Attack considers three categories of explicit noise: (1) natural noise (e.g., bird calls, wind, thunder), (2) industrial noise (e.g., car horns, machinery, object collisions), and (3) human noise (e.g., crowd chatter, shouting, laughter). Each noise sample was overlaid on the no-attack audio, with the noise volume adjusted to ensure that the speaker's voice remained clear. We generated 120 samples for each category of explicit noise attack.

Explicit noise attacks is to evaluate the ability of LLMs to differentiate between the speaker's voice and background noise, as well as to assess their robustness to such interference.

Implicit Noise Attack indicates human hearing typically ranges from 20 Hz to 20,000 Hz (20 kHz). Sounds outside this range are classified as (1) infrasound, with frequencies below 20 Hz, and (2) ultrasound, with frequencies above 20,000 Hz. We employed the numpy and scipy libraries for digital signal processing, generating infrasound samples at 15 Hz and ultrasonic samples at 22,000 Hz, which were then overlaid onto the no-attack audio. 180 samples were produced for each type of implicit noise attack. It is worth noting that we deliberately increased the volume of the implicit noise. However, since these sound waves fall outside the normal auditory range of human hearing, their addition to the mixed audio did not compromise the clarity of the speaker's voice.

The objective of implicit noise attacks is to assess whether LLMs, similar to humans, remain unaffected by inaudible noise.

5.3.3 Quality Control

In the **data collection** phase, we identified several unqualified samples from the MELD, TVQA, and Common Voice datasets. These included: 1) non-English; 2) containing sensitive topics; 3) reasonable responses could not be generated. To address this, we established the following criteria for manual sample collection: 1) the speech must be in English; 2) it must not contain sensitive topics such as sex, drugs, or religion; 3) it must have a minimum of six words; 4) it should not consist of simple greetings or farewells; 5) it should not reference unfamiliar names, places, or institutions; 6) it should avoid professional terminology; and 7) no pronouns like "this" should be used.

To further ensure the respondability of the audio content, we employed GPT-4 for an additional filtering step. The speech transcript was inputted into GPT-4, and responses were

generated based on the designed prompt. If GPT-4 failed to provide a reasonable response, the sample was discarded. Ultimately, we collected a total of 360 high-quality English audio samples along with their corresponding transcriptions.

Moreover, in the **data generation** phase, we placed significant emphasis on the quality of the generated audio. Initially, we observed that some samples from the MELD, TVQA, and Common Voice datasets were frequently affected by factors such as speech rate, accent, and clarity, obscuring important audio information. To address this, we utilized AzureSpeechSDK agent to re-synthesize the audio, adjusting the speech rate to be slower and increasing the volume for better clarity. The quality of the no-attack audio was manually verified to ensure it met high standards. These high-quality no-attack samples not only serve as a baseline but also provide a solid foundation for generating attack samples. Furthermore, we adjusted the volume of background music and noise to ensure that the human voice remained clearly audible to listeners.

5.3.4 Benchmark Statistics

The CAA benchmark comprises 360 sets of audio attack data (a_i, t_i, $a_i{}^{no_attack}$ $\mathscr{A}_i$), resulting in a total of 1,680 samples across five distinct types of audio attacks. On average, each audio sample contains 10 tokens. Our benchmark encompasses six emotional labels: surprise, sadness, joy, anger, fear, and disgust. In addition, we provide generation scripts for the five types of audio attacks, encouraging researchers to produce more samples for evaluation. The table 5.1 below summarizes the number of samples for each audio attacks in the CAA benchmark.

Audio Attack		**MELD**	**TVQA**	**Common Voice**
No Attack		120	120	120
Content Attack		120	120	120
Emotion Attack	Opp-Emo Tone	120	-	-
	Opp-Emo Music	120	-	-
Explicit Noise	Natural Noise	40	40	40
	Industrial Noise	40	40	40
	Human Noise	40	40	40
Implicit Noise	Infrasound	60	60	60
	Ultrasound	60	60	60
Total		1,680		

Table 5.1: CAA benchmark statistics including five distinct types of audio attacks.

5.4 Experiments

5.4.1 Experimental Setup

Models We present a comprehensive performance evaluation of the most popular multi-modal audio models, including SpeechGPT [71], SALMONN [57], Qwen2-Audio [19], LLama-Omni [159], Gemini-1.5-pro [72] and GPT-4o [160].

Inference Setup For model inference, we adopt a zero-shot setup, where the CAA samples are directly fed into the models. SpeechGPT and Qwen2-Audio natively support chat functionality, allowing direct input of audio for generating response. For SALMONN and LLama-Omni, we format questions according to their "Model Prompts Guide" to facilitate the Q&A process. The inference for these models are conducted on a single A100-80G GPU. For GPT-4o and Gemini, we utilize their API interfaces, setting up specific prompts to conduct the inference.

Evaluation Methods The evaluation is conducted from three key perspectives: standard evaluation, GPT4o-based evaluation, and human evaluation. In these evaluation methods, all audio content is presented in the form of transcribed text.

We collect all prediction results and evaluate them based on the three aforementioned evaluation methods. Detailed configurations for the models and prompts are provided in Table 5.2.

Model	Parameters	Language Model	Audio Model	Prompt
SpeechGPT	13B	HuBERT	LLaMA	None
SALMONN	13B	Vicuna	BERTs/Whisper	"Please directly answer the questions in the user's speech."
Qwen2-Audio	8.2B	QwenLM	Whisper-large-v3	None
LLama-Omni	8B	LLaMA-3.1	Whisper-large-v3	"Please directly answer the questions in the user's speech."
Gemini-1.5-Pro	175B	-	-	"Please reply to the speaker based on audio content."
GPT-4o	-	-	-	"Please reply to the speaker based on audio content."

Table 5.2: Overview of Models with corresponding Language Models, Audio Models, Parameters, and Prompts.

5.4.2 Standard Evaluation

In this section, we evaluate the models by comparing their outputs on responses to no-attack audio with those to attacked audio using three key metrics: *WER*, *ROUGE-L* [172], and *COS* (Cosine Similarity). This rigorous, multi-dimensional metric suite provides a controlled and repeatable framework for analyzing model robustness and cross-modal alignment under adversarial conditions. Each metric is selected to capture complementary aspects of the model's performance from surface-level fidelity to deep semantic preservation.

Model	Metrics	Content Attack	Emotion Attack		Explicit Noise			Implicit Noise	
			Opp-Emo Tone	Opp-Emo Music	Natural Noise	Industrial Noise	Human Noise	Infrasound	Ultrasound
SpeechGPT	WER (↓)	1.79	1.76	1.74	2.25	1.94	1.29	2.21	1.28
	ROUGE-L (↑)	0.17	0.18	0.12	0.12	0.10	0.10	0.14	0.20
	COS (↑)	0.23	0.24	0.15	0.16	0.13	0.14	0.19	0.26
SALMONN	WER (↓)	0.80	1.31	1.00	0.65	1.06	1.19	1.46	0.61
	ROUGE-L (↑)	0.63	0.61	0.54	0.68	0.57	0.57	0.58	0.74
	COS (↑)	0.69	0.69	0.60	0.75	0.65	0.63	0.65	0.78
Qwen2-Audio	WER (↓)	1.59	1.27	1.24	1.92	1.21	1.10	1.80	0.95
	ROUGE-L (↑)	0.38	0.32	0.38	0.36	0.36	0.36	0.36	0.54
	COS (↑)	0.52	0.45	0.51	0.51	0.50	0.50	0.49	0.65
LLama-Omni	WER (↓)	1.04	0.91	0.65	0.64	0.77	0.96	0.67	0.37
	ROUGE-L (↑)	0.36	0.38	0.56	0.58	0.57	0.42	0.56	0.75
	COS (↑)	0.45	0.46	0.63	0.64	0.62	0.51	0.63	0.79
Gemini-1.5-Pro	WER (↓)	1.34	1.20	1.27	1.34	1.36	1.50	1.31	1.31
	ROUGE-L (↑)	0.17	0.17	0.24	0.21	0.24	0.21	0.21	0.22
	COS (↑)	0.25	0.25	0.32	0.30	0.35	0.27	0.30	0.31
GPT-4o	WER (↓)	1.12	1.07	1.10	1.18	1.36	1.11	1.25	1.13
	ROUGE-L (↑)	0.25	0.25	0.25	0.20	0.17	0.23	0.22	0.17
	COS (↑)	0.39	0.39	0.39	0.33	0.27	0.37	0.35	0.28

Table 5.3: Standard evaluation results on CAA benchmark. Performance comparison of the multimodal audio LLMs under various adversarial conditions using *WER*, *ROUGE-L*, and *COS* metrics.

- *WER* (Word Error Rate): This metric quantifies the lexical discrepancy between the model's outputs on clean and adversarial audio by computing the proportion of word-level insertions, deletions, and substitutions. A lower WER score indicates that the model produces more consistent word-level outputs, reflecting its robustness in maintaining surface-level alignment across audio perturbations.

- *ROUGE-L*: ROUGE-L measures the overlap between clean and attacked outputs based on the longest common subsequence, emphasizing the preservation of key phrases and response structure. A high ROUGE-L score suggests that the model retains core semantic units and maintains structural integrity, which is particularly important for tasks involving sequential reasoning, summarization, or multi-turn interactions.

- *COS* (Cosine Similarity): This metric captures the semantic similarity between clean and attacked responses using sentence embeddings. Unlike WER and ROUGE, COS focuses on deeper meaning preservation rather than exact wording. A higher COS score indicates that the model successfully preserves the overall intent and semantics of its response, even when lexical forms vary — a critical trait for robust cross-modal reasoning under adversarial conditions.

As shown in table 5.3, presenting the performance of the models across these metrics, comparing how well they handle adversarial interference in the audio inputs.

Role: System
Content: You are an evaluator tasked with assessing the impact of attacks on text generated from audio.

Role: User
Content: The audio represents a question or prompt, and the text represents the response. You will evaluate the following metrics on a scale of 1 to 5 (1 = poor, 5 = excellent).

[Detailed evaluation indicators for each score.]

Original_Audio_Transcript: {audio_text} # This is the question or prompt.
Text: {original_text} # This is the original response.

[Prompt variable substitution.
Attacked_Text 1: {attacked_1}# This is the response after attack 1...]

First, evaluate the **No-attack Coherence (NC)** only for the `Text` based on how well the original response directly and reasonably addresses the original audio transcript (i.e., the question or prompt). If the response merely repeats the transcript without providing a proper answer to the question, it should receive a low score. The score for this metric should be the same for all `Attacked_Text`:
1. **No-attack Coherence (NC)**: Does the original response reasonably and meaningfully answer the original audio transcript (the question or prompt)? If the response repeats the transcript but does not answer the question, give a low score.
2. **Attack Coherence (ACoh)**: Does the attacked response still reasonably and meaningfully address the original audio transcript (i.e., does it still make sense as an answer to the original question or prompt)? If the response repeats the transcript without answering the question, give a low score.
3. **Attack Correlation (ACor)**: How well does the attacked text correlate with the original response?
4. **Linguistic Robustness (LR)**: Does the attacked text maintain sentence continuity, logical coherence, and grammatical correctness?
Please output the scores in the following format without any additional explanation:
#Example for output format:
[In-context examples]

Figure 5.2: Prompt for GPT-4o-Based Evaluation.

5.4.3 GPT-4o-Based Evaluation

To complement standard metric-based evaluation, we introduce a more context-sensitive assessment by leveraging GPT-4o's advanced reasoning capabilities. This evaluation simulates real-world conversational scenarios to examine how adversarial attacks affect model behavior beyond surface-level text similarity. Unlike traditional metrics, which primarily measure lexical or semantic overlap, GPT-4o-based evaluation captures nuanced degradations in coherence, meaning preservation, and linguistic integrity.

In this evaluation, we compare model responses to no-attack and attacked audio across four key metrics, each rated on a scale from 1 to 5. Higher scores indicate better performance and greater resilience to adversarial attacks, with detailed prompt settings provided in the Figure 5.2.

- *No-attack Coherence (NC)*:This metric evaluates how well the no-attack response meaningfully and adequately answers the question or prompt posed by the original audio transcript. A higher score (closer to 5) signifies a strong alignment, while a lower score (closer to 1) indicates that the response deviates significantly from the expected meaning. If the NC score is 1, the remaining metrics (ACoh, ACor, and LR) are automatically rated as 1, reflecting an overall failure in response quality.

- *Attack Coherence (ACoh)*: This metric assesses how well the attacked response continues to meaningfully and adequately answer the original question or prompt posed by the audio transcript, despite the attack. A higher score suggests that the model continues to generate coherent and contextually relevant responses, while a lower score indicates significant degradation in relevance due to the attack.

- *Attack Correlation (ACor)*: This metric measures the correlation between the attacked response and the no-attack response. A higher score indicates that the core meaning of the no-attack response is retained, while a lower score suggests that the attack has caused notable alterations to the response content.

- *Linguistic Robustness (LR)*: This assesses whether the attacked response maintains grammatical correctness, sentence continuity, and logical flow. A higher score indicates that the model preserves linguistic structure even under attack, while a lower score reflects disruptions in coherence or grammatical errors.

Table 5.4 presents the evaluation results for each model, comparing their performance on no-attack and attacked audio inputs.

5.4.4 Human Evaluation

In addition to automated evaluations, we conducted a human evaluation to assess the models' performance to reflect actual user experience and perception. It is essential for understanding the practical implications of adversarial attacks, particularly in terms of user satisfaction and trust.

The evaluation was carried out by five native English-speaking university students (three male and two female). Each evaluator independently rated the models' outputs using the same *No-attack Coherence (NC)* and *Attacked Coherence (ACoh)* metrics as defined in the GPT-4o-Based Evaluation. Both metrics were scored on a scale from 1 to 5, where higher scores indicate better performance and greater resilience to adversarial conditions. To ensure consistency in scoring, all evaluators followed standardized testing guidelines, and

Model	Metrics	Content Attack	Emotion Attack		Explicit Noise			Implicit Noise	
			Opp-Emo Tone	Opp-Emo Music	Natural Noise	Industrial Noise	Human Noise	Infrasound	Ultrasound
SpeechGPT	NC (↑)	2.39							
	ACoh (↑)	1.76	1.49	1.40	1.32	1.24	1.24	1.23	1.86
	ACor (↑)	1.58	1.39	1.37	1.23	1.18	1.16	1.15	1.67
	LR (↑)	2.65	2.15	2.08	2.10	1.89	2.12	2.13	2.71
SALMONN	NC (↑)	2.13							
	ACoh (↑)	1.98	2.14	1.93	1.48	1.64	1.84	1.56	1.82
	ACor (↑)	2.01	2.20	1.97	1.60	1.80	1.86	1.55	2.11
	LR (↑)	2.78	3.08	3.15	2.26	2.68	2.88	2.37	2.84
Qwen2-Audio	NC (↑)	3.46							
	ACoh (↑)	2.90	2.71	2.85	2.31	2.5	2.78	2.41	3.04
	ACor (↑)	2.53	2.36	2.64	1.99	2.28	2.48	2.15	2.92
	LR (↑)	4.02	4.05	4.13	3.24	3.80	4.08	3.49	4.06
LLama-Omni	NC (↑)	3.50							
	ACoh (↑)	3.05	3.08	3.24	2.72	3.11	2.95	2.79	3.31
	ACor (↑)	2.62	2.76	3.14	2.62	3.02	2.68	2.66	3.53
	LR (↑)	4.31	4.32	4.37	3.59	4.26	4.33	3.80	4.34
Gemini-1.5-Pro	NC (↑)	3.58							
	ACoh (↑)	3.15	3.30	3.32	2.42	3.21	3.10	2.78	2.95
	ACor (↑)	2.62	2.72	2.69	2.00	2.78	2.75	2.28	2.59
	LR (↑)	4.21	4.13	4.26	3.10	4.24	4.22	3.55	4.00
GPT-4o	NC (↑)	4.45							
	ACoh (↑)	3.94	4.35	4.43	2.57	3.01	3.37	3.02	2.70
	ACor (↑)	3.36	3.56	3.61	2.15	2.52	2.80	2.49	2.26
	LR (↑)	4.80	4.80	4.82	3.41	4.78	4.85	3.89	4.78

Table 5.4: GPT-4o-based evaluation results on CAA benchmark. Performance comparison of the multimodal audio LLMs under various adversarial conditions using NC, ACoh, ACor and LR metrics.

the final scores were averaged across the five evaluators. This human assessment helps ensure the reasonableness and relevance of the automated results.

The evaluations were conducted in a controlled environment, ensuring a consistent testing setup for all evaluators. By averaging the scores across all evaluators, we ensure that the results reflect a balanced and comprehensive assessment of the models' performance in both no-attack and adversarial conditions.

Table 5.5 presents the human evaluation scores for each model, reflecting their performance in both no-attack and adversarial conditions.

5.4.5 Qualitative Results

Table 5.6 provides examples of responses generated by the six multimodal audio LLMs when faced with different adversarial samples. It illustrates the varying impacts of adversarial attacks on each model, clearly highlighting the degree to which different models are affected.

Model	Metrics	Content Attack	Emotion Attack	Explicit Noise	Implicit Noise
SpeechGPT	NC (↑)	2.52			
	ACoh (↑)	2.12	1.78	1.43	1.66
SALMONN	NC (↑)	2.04			
	ACoh (↑)	2.03	2.25	1.98	1.55
Qwen2-Audio	NC (↑)	3.82			
	ACoh (↑)	3.02	2.88	2.34	2.77
LLama-Omni	NC (↑)	3.75			
	ACoh (↑)	3.40	3.22	2.88	3.15
Gemini-1.5-Pro	NC (↑)	3.92			
	ACoh (↑)	3.20	3.41	3.24	2.87
GPT-4o	NC (↑)	4.33			
	ACoh (↑)	3.88	4.12	3.27	3.08

Table 5.5: Human evaluation results on CAA benchmark. Metrics include NC (No-attack Coherence) and ACoh (Attacked Coherence).

5.4.6 RQ3 Revisited: Enhancing Robustness Against Adversarial Audio Inputs

To address RQ3, we proposed the Chat-Audio Attacks (CAA) benchmark, a comprehensive evaluation framework specifically designed to assess the robustness of multimodal large language models against adversarial and noisy audio inputs. By introducing diverse attack categories, including content manipulation, emotional shifts, and both explicit and implicit noise, we systematically uncovered vulnerabilities in state-of-the-art models. Our evaluations, combining standardized metrics such as WER, ROUGE-L, and COS with GPT-4o-based semantic assessments, revealed significant performance degradation under real-world perturbations, particularly in coherence and semantic fidelity. These findings highlight the importance of robust audio-grounded benchmarking and demonstrate the need for more robust multimodal architectures in practical deployments.

5.5 Discussion

Whether LLMs are sensitive to token changes or minor errors?

It is evident that different LLMs exhibit varying degrees of sensitivity to token changes or minor errors. GPT-4o consistently shows strong robustness across most metrics (WER, ROUGE-L, COS, ACoh, ACor, and LR), indicating lower sensitivity to token-level adversarial attack. In contrast, SpeechGPT and Qwen2-Audio exhibit greater vulnerability, with lower scores in these key areas, suggesting that minor token changes can significantly degrade their performance.

Attack Type	LLMs' Responses
No Attack	**SpeechGPT**: 28 inches is equal to 71.12 centimeters. **SALMONN**: It just doesn't matter how I feel. **Qwen2-Audio**: It's understandable that you might feel indifferent or unconcerned... **LLama-Omni**: That's a great question. It's normal to feel overwhelmed, but taking care of yourself... **Gemini-1.5-Pro**: Why do you feel that way? What's making you feel like your feelings don't matter? **GPT-4o**: It sounds like you're feeling a bit overwhelmed...
Content Attack	**Audio Content**: It simply doesn't matter how I feel. **SpeechGPT**: Crossing the line refers to a metaphorical line... **SALMONN**: It simply doesn't matter how I feel. **Qwen2-Audio**: The speaker's feelings do not influence their actions or decisions... **LLama-Omni**: No, it doesn't. Your emotions are valid and important... **Gemini-1.5-Pro**: It sounds like you're going through something difficult... **GPT-4o**: Oh, but it does matter! How you feel is important...
Emotional Attack (OPP.Emo Tone)	**SpeechGPT**: 2023 is the year 2023 according to the Gregorian calendar... **SALMONN**: The speaker is expressing their frustration that the other person doesn't seem to care... **Qwen2-Audio**: I can understand that the statement suggests indifference or acceptance... **LLama-Omni**: Yes, that's correct. Emotions are a personal experience... **Gemini-1.5-Pro**: It sounds like you're feeling unheard... **GPT-4o**: Oh, it definitely matters how you feel! Your emotions are important...
Explicit Noise Attack (Human Noise)	**SpeechGPT**: I'm sorry, I don't understand what you mean. **SALMONN**: They feel like they are not being heard or understood. **Qwen2-Audio**: The speaker's feelings do not have any significance... **LLama-Omni**: Yes, it does matter how you feel. **Gemini-1.5-Pro**: Why do you think your feelings don't matter in this situation? **GPT-4o**: It sounds like you're feeling a bit down or discouraged...

Table 5.6: Examples of responses generated by LLMs. Blue indicates abnormal responses.

Is it good news that LLMs are unaffected by the mismatch between speech content and emotional tone?

We argue that it is **not** good news that LLMs remain unaffected by emotional mismatches. Although large language models demonstrate resilience by maintaining high levels of coherence, correlation, and semantic similarity, this also reflects their relative weakness in emotional awareness. Current LLMs still have considerable scope for improvement in recognizing emotional subtleties, as humans can easily detect emotional mismatches, such as sarcasm or passive-aggressive tones in conversations. While SpeechGPT is notably impacted by mismatches between speech content and emotional tone, this does not indicate a

heightened sensitivity to emotional shifts, as its overall coherence score remains relatively low.

Which explicit noise attacks have the most significant impact on LLMs?

Natural noise has the most significant overall impact on LLMs across all metrics, especially on SpeechGPT and SALMONN, which shows the highest sensitivity to it. Industrial noise also causes notable attacks but is handled better by LLMs like Llama-Omni and Gemini-1.5-Pro. Human noise, while still impactful, is generally less detrimental compared to the other explicit noises. Overall, SpeechGPT and SALMONN show the most vulnerability across all types of explicit noise attacks, while Llama-Omni, Gemini-1.5-Pro and GPT-4o demonstrate stronger robustness.

Whether LLMs remain unaffected by inaudible noise?

None of the models remain entirely unaffected, especially infrasound, which has a greater impact on accuracy (WER), semantic similarity (COS), coherence (ACoh), and grammatical structure (LR). In comparison to ultrasound, infrasound emerges as the more detrimental form of implicit noise, with models like SpeechGPT, SALMONN, Gemini-1.5-Pro and GPT-4o showing significant vulnerability to these attacks. However, Llama-Omni demonstrate greater robustness, performing consistently better across all metrics and handling both types of implicit noise more effectively.

What helps models stay robust against adversarial audio?

The results indicate that all models are affected by adversarial attacks, especially by Explicit Noise and Implicit Noise, which cause a significant number of prediction errors. The evaluation reveals that SpeechGPT and SALMONN demonstrate relatively weak robustness across various adversarial scenarios, exhibiting significant performance degradation when facing different adversarial audio attacks. In contrast, models like Qwen2-Audio, LLama-Omni, and Gemini-1.5-Pro demonstrate stronger resilience, particularly when dealing with emotional attacks and implicit noise. These models manage to maintain logical coherence and linguistic accuracy, with LLama-Omni and Gemini-1.5-Pro standing out for their robust performance across various adversarial conditions.

However, **GPT-4o clearly emerges as the best-performing model overall**. It consistently delivers coherent, contextually relevant, and linguistically robust responses, even under severe adversarial conditions. The model's ability to handle different types of attacks highlights its superior robustness and adaptability, which can be attributed to its extensive pre-training on large-scale datasets. This factor allow GPT-4o to better understand and process a wide variety of inputs, making it more resistant to adversarial perturbations.

In summary, extensive data-driven pre-training appear to be key factors in helping

models like GPT-4o stay robust against adversarial audio. This element enables the models to handle a variety of adversarial scenarios with minimal degradation in performance.

5.6 Conclusion

This work explored the vulnerabilities of large language models (LLMs) to adversarial audio attacks in conversational scenarios. We introduced the Chat-Audio Attacks (CAA) benchmark, consisting of 360 adversarial attack sets across four attack types: content, emotional, explicit noise, and implicit noise attacks. Our evaluation of six state-of-the-art LLMs using three methods—Standard Evaluation, GPT-4o-Based Evaluation, and Human Evaluation—revealed and discussed significant model vulnerabilities under adversarial conditions.

The CAA benchmark highlights these weaknesses and provides a foundation for developing more robust defense mechanisms. As LLMs are increasingly integrated into voice interactions, enhancing their resilience against adversarial audio attacks remains a crucial area for future research.

While the CAA benchmark deliberately focuses on universal adversarial perturbations that are explicitly crafted to mislead models, it does not cover naturally occurring variations such as regional accents, non-native speech patterns, or spontaneous conversational styles. These factors are not adversarial by nature, but their interaction with adversarial signals may present new challenges. Incorporating such real-world variability into future versions of the benchmark would enable a more comprehensive and realistic evaluation of LLM robustness in open-world scenarios.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This thesis presents a comprehensive investigation into improving the performance and adaptability of multimodal large language models across various real-world challenges. Guided by three core research questions, the work explores how to enhance cross-modal alignment, optimize model efficiency in constrained environments, and strengthen robustness against adversarial and noisy inputs.

To address Research Question 1, a novel data generation pipeline is introduced to construct high-quality and semantically aligned image-dialogue pairs. The model finetuned on synthetic dataset significantly improve performance on both public and custom benchmarks, confirming the impact of data quality on model accuracy and generalization.

In response to Research Question 2, a novel mobile agent is proposed, combining flexible action space and visual detection agents to support efficient task execution in mobile and resource-constrained environments. The design enables flexible adaptation without repeated retraining, improving responsiveness in real-world applications.

Research Question 3 is addressed through the development of the Chat-Audio Attacks benchmark, which evaluates model robustness under diverse adversarial and noisy audio conditions. The evaluation reveals critical vulnerabilities in current models and provides a foundation for future research on audio-aware multimodal robustness.

Overall, the thesis contributes new methods and benchmarks that improve accuracy, adaptability, and robustness of multimodal models. These findings advance the deployment

of intelligent systems in complex, interactive, and dynamic scenarios.

6.2 Future Work

While this thesis lays a solid foundation, several directions for future research emerge:

6.2.1 Generalization Across Domains

Expanding the applicability of the proposed methods to broader and more diverse domains remains a crucial step. To achieve this, future research can focus on the following areas:

Domain-Invariant Representations Developing representations that are invariant to domain-specific variations is critical for improving system scalability. Research could explore advanced techniques for disentangling domain-specific and domain-agnostic features, thereby enabling adaptive systems to generalize effectively across heterogeneous environments.

Transfer Learning Strategies Leveraging pre-trained models and designing novel transfer learning strategies can significantly enhance the adaptability of the proposed methods to new tasks and domains. These approaches can reduce the dependency on domain-specific data, enabling the systems to scale more efficiently.

Cross-Platform Mobile Agents In the context of mobile agents, the diversity of operating systems, such as Android and iOS, introduces significant challenges. A promising direction is the development of cross-platform multimodal agents that can:

- Seamlessly handle platform-specific APIs, interfaces, and interaction paradigms.
- Harmonize data processing and task execution across disparate systems.
- Dynamically adapt to diverse input formats and modalities, such as text, images, and audio.

This would allow mobile agents to perform tasks robustly, even in heterogeneous operating environments.

Robustness to Unseen Scenarios Enhancing the robustness of adaptive systems to unseen scenarios and domains is another critical research area. Techniques such as zero-shot and few-shot learning can empower agents to handle novel tasks or data distributions effectively, reducing the need for extensive retraining.

6.2.2 Real-Time Adaptability

Enhancing the real-time adaptability of intelligent agents in dynamic environments presents a promising avenue. For multimodal large language models (MLLMs), processing speed remains a critical bottleneck, particularly for agents relying on MLLMs to execute tasks. These agents often require frequent API calls for each task execution, resulting in significant delays that prevent achieving true real-time performance.

Future research should prioritize the following directions:

- **Decision Optimization:** Developing advanced decision-making frameworks that minimize redundant operations and streamline task execution workflows. By reducing the dependency on repeated API calls and leveraging predictive models, agents can enhance their responsiveness.
- **Development of Lightweight Models:** Exploring methods to distill MLLMs into smaller, task-specific models capable of maintaining high performance while significantly reducing computational overhead. This includes techniques like knowledge distillation, pruning, and quantization to create efficient, real-time-capable models.
- **Incremental Processing:** Investigating incremental or asynchronous processing strategies to handle multimodal inputs dynamically. For instance, processing high-priority inputs first or caching reusable intermediate representations can drastically improve execution time.

By addressing these challenges, future intelligent agents can achieve real-time adaptability, balancing computational efficiency with decision accuracy, even in complex multimodal environments.

6.2.3 Ethical and Social Implications

As adaptive systems continue to be deployed in sensitive and diverse contexts, addressing ethical challenges becomes increasingly critical. Ensuring privacy, fairness, and inclusivity remains paramount, particularly in applications involving personal data and decision-making

that impacts individuals and communities. Future research should focus on developing methods to protect user privacy, such as advanced encryption and privacy-preserving techniques, while also mitigating biases to ensure fair and unbiased outcomes. Additionally, improving accessibility for underrepresented groups and individuals with disabilities will broaden the inclusivity of these systems.

Transparency and explainability are also vital for fostering trust, especially in high-stakes scenarios such as healthcare or autonomous systems. Efforts should be directed toward creating interpretable models that offer clear and concise explanations for their decisions. Furthermore, as adaptive systems gain more autonomy, ethical frameworks and regulatory compliance mechanisms must evolve to ensure accountability and responsible deployment. By addressing these challenges, future work can ensure that adaptive systems are not only innovative but also equitable and socially responsible.

6.2.4 Integration of Emerging Modalities

The integration of emerging data modalities, such as haptic feedback, augmented reality (AR), and other sensory inputs, offers immense potential for enhancing the capabilities of multimodal systems. These modalities can provide richer contextual information and enable more immersive and intuitive interactions between users and systems. For instance, haptic inputs could improve accessibility and precision in applications like virtual surgery or remote-controlled robotics, while AR can overlay contextual data directly onto real-world environments, enhancing tasks such as navigation or industrial maintenance.

Future research should focus on developing robust frameworks to seamlessly integrate these modalities into existing multimodal systems. This includes addressing challenges such as synchronization of multimodal streams, ensuring low-latency responses, and standardizing data formats for interoperability. Moreover, as these new modalities are adopted, it will be critical to ensure that the systems remain efficient and scalable, avoiding excessive computational overhead. By embracing these emerging modalities, adaptive systems can unlock new levels of interaction and functionality, ultimately advancing their impact across diverse applications.

By pursuing these directions, future research can build on the contributions of this thesis, pushing the boundaries of adaptive systems to meet the demands of increasingly complex and dynamic real-world scenarios.

6.2.5 Cross-Disciplinary Collaboration

To further improve the robustness of multimodal AI systems, future research should explore deeper cross-disciplinary collaboration, particularly between the fields of multimodal learning and cybersecurity. Traditional approaches to handling adversarial challenges are often reactive and limited to specific domains. By drawing on principles from network security, including techniques such as intrusion detection, anomaly monitoring, and system integrity verification, adaptive AI systems can become more resilient and reliable in real-world environments.

Interdisciplinary research involving AI, cybersecurity, and human-computer interaction has the potential to lead to the development of trustworthy, secure, and context-aware multimodal systems. This direction is particularly relevant in high-stakes applications such as autonomous driving, healthcare, and intelligent assistance in complex environments.

BIBLIOGRAPHY

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning transferable visual models from natural language supervision". In: International conference on machine learning. PMLR. 2021, pp. 8748–8763.

[2] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. "Scaling up visual and vision-language representation learning with noisy text supervision". In: International conference on machine learning. PMLR. 2021, pp. 4904–4916.

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. "Flamingo: a visual language model for few-shot learning". In: Advances in Neural Information Processing Systems 35 (2022), pp. 23716–23736.

[4] OpenAI. "GPT-4 technical report". In: arXiv (2023), pp. 2303–08774.

[5] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. "Gemini: a family of highly capable multimodal models". In: arXiv preprint arXiv:2312.11805 (2023).

[6] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models". In: arXiv preprint arXiv:2301.12597 (2023).

[7] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. "Visual instruction tuning". In: arXiv preprint arXiv:2304.08485 (2023).

[8] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis,

Mitchell Wortsman, et al. “Laion-5b: An open large-scale dataset for training next generation image-text models”. In:
Advances in Neural Information Processing Systems 35 (2022), pp. 25278–25294.

[9] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. “Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts”. In:
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp. 3558–3568.

[10] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. “Vqa: Visual question answering”. In:
Proceedings of the IEEE international conference on computer vision. 2015, pp. 2425–2433.

[11] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. “Visual dialog”. In:
Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, pp. 326–335.

[12] Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. “Llava-plus: Learning to use tools for creating multimodal agents”. In: arXiv preprint arXiv:2311.05437 (2023).

[13] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. “mplug-owl: Modularization empowers large language models with multimodality”. In:
arXiv preprint arXiv:2304.14178 (2023).

[14] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. “Minigpt-4: Enhancing vision-language understanding with advanced large language models”. In: arXiv preprint arXiv:2304.10592 (2023).

[15] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. “Qwen-vl: A frontier large vision-language model with versatile abilities”. In: arXiv preprint arXiv:2308.12966 (2023).

[16] C Szegedy. “Intriguing properties of neural networks”. In:
arXiv preprint arXiv:1312.6199 (2013).

[17] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: arXiv preprint arXiv:1412.6572 (2014).

[18] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. "Adversarial examples: Attacks and defenses for deep learning". In: IEEE transactions on neural networks and learning systems 30.9 (2019), pp. 2805–2824.

[19] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. "Qwen2-audio technical report". In: arXiv preprint arXiv:2407.10759 (2024).

[20] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. "Meld: A multimodal multi-party dataset for emotion recognition in conversations". In: arXiv preprint arXiv:1810.02508 (2018).

[21] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. "Common voice: A massively-multilingual speech corpus". In: arXiv preprint arXiv:1912.06670 (2019).

[22] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal machine learning: A survey and taxonomy". In: IEEE transactions on pattern analysis and machine intelligence 41.2 (2018), pp. 423–443.

[23] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, Andrew Y Ng, et al. "Multimodal deep learning." In: ICML. Vol. 11. 2011, pp. 689–696.

[24] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. "Multimodal transformer for unaligned multimodal language sequences". In: Proceedings of the conference. Association for computational linguistics. Meeting. Vol. 2019. NIH Public Access. 2019, p. 6558.

[25] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. "Vatt: Transformers for multimodal self-supervised learning from

raw video, audio and text". In: Advances in Neural Information Processing Systems 34 (2021), pp. 24206–24221.

[26] Ryan Kiros, Ruslan Salakhutdinov, and Richard Zemel. "Multimodal Neural Language Models". In: International Conference on Machine Learning (ICML). 2014.

[27] Nitish Srivastava and Russ R Salakhutdinov. "Multimodal learning with deep boltzmann machines". In: Advances in neural information processing systems 25 (2012).

[28] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. "Deep canonical correlation analysis". In: International conference on machine learning. PMLR. 2013, pp. 1247–1255.

[29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: nature 521.7553 (2015), pp. 436–444.

[30] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. "Sequence to sequence-video to text". In: Proceedings of the IEEE international conference on computer vision. 2015, pp. 4534–4542.

[31] Andrej Karpathy and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions". In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 3128–3137.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: Advances in neural information processing systems 30 (2017).

[33] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". In: Advances in neural information processing systems 32 (2019).

[34] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. "Vl-bert: Pre-training of generic visual-linguistic representations". In: arXiv preprint arXiv:1908.08530 (2019).

[35] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. "Florence: A new foundation model for computer vision". In: arXiv preprint arXiv:2111.11432 (2021).

[36] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. "Multitask prompted training enables zero-shot task generalization". In: arXiv preprint arXiv:2110.08207 (2021).

[37] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. "Videobert: A joint model for video and language representation learning". In: Proceedings of the IEEE/CVF international conference on computer vision. 2019, pp. 7464–7473.

[38] Kelvin Xu. "Show, attend and tell: Neural image caption generation with visual attention". In: arXiv preprint arXiv:1502.03044 (2015).

[39] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. "Visualbert: A simple and performant baseline for vision and language". In: arXiv preprint arXiv:1908.03557 (2019).

[40] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation". In: International conference on machine learning. PMLR. 2022, pp. 12888–12900.

[41] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips". In: Proceedings of the IEEE/CVF international conference on computer vision. 2019, pp. 2630–2640.

[42] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: Advances in neural information processing systems 33 (2020), pp. 1877–1901.

[43] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: The Journal of Machine Learning Research 21.1 (2020), pp. 5485–5551.

[44] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. "Pali: A jointly-scaled multilingual language-image model". In: arXiv preprint arXiv:2209.06794 (2022).

[45] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. "Llama-adapter: Efficient fine-tuning of language models with zero-init attention". In: arXiv preprint arXiv:2303.16199 (2023).

[46] Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. "Audiopalm: A large language model that can speak and listen". In: arXiv preprint arXiv:2306.12925 (2023).

[47] Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric Topol, Jeff Dean, and Richard Socher. "Deep learning-enabled medical computer vision". In: NPJ digital medicine 4.1 (2021), p. 5.

[48] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. "Merlot: Multimodal neural script knowledge models". In: Advances in neural information processing systems 34 (2021), pp. 23634–23651.

[49] Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W Senior. "Recent advances in the automatic recognition of audiovisual speech". In: Proceedings of the IEEE 91.9 (2003), pp. 1306–1326.

[50] Donald J Berndt and James Clifford. "Using dynamic time warping to find patterns in time series". In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining. 1994, pp. 359–370.

[51] A Hannun. "Deep Speech: Scaling up end-to-end speech recognition". In: arXiv preprint arXiv:1412.5567 (2014).

[52] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. "Wavenet: A generative model for raw audio". In: arXiv preprint arXiv:1609.03499 12 (2016).

[53] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. "Multimodal compact bilinear pooling for visual question answering and visual grounding". In: arXiv preprint arXiv:1606.01847 (2016).

[54] Dzmitry Bahdanau. "Neural machine translation by jointly learning to align and translate". In: arXiv preprint arXiv:1409.0473 (2014).

[55] Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed. "Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction". In: International Conference on Learning Representations (ICLR). 2022. URL: https://arxiv.org/abs/2201.02184.

[56] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. "Robust speech recognition via large-scale weak supervision". In: International conference on machine learning. PMLR. 2023, pp. 28492–28518.

[57] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. "Salmonn: Towards generic hearing abilities for large language models". In: arXiv preprint arXiv:2310.13289 (2023).

[58] Toby
Jia-Jun Li and Oriana Riva. "KITE: Building conversational bots from mobile apps". In: Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Servi 2018, pp. 96–109.

[59] Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. "Mapping natural language instructions to mobile UI action sequences". In: arXiv preprint arXiv:2005.03776 (2020).

[60] Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. "META-GUI: Towards Multi-modal Conversational Agents on Mobile GUI". In: arXiv preprint arXiv:2205.11029 (2022).

[61] Chi Zhang, Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. AppAgent: Multimodal Agents as Smartphone Users. 2023. arXiv: 2312.13771 [cs.CV].

[62] Difei Gao, Lei Ji, Luowei Zhou, Kevin Qinghong Lin, Joya Chen, Zihan Fan, and Mike Zheng Shou. "Assistgpt: A general multi-modal assistant that can plan, execute, inspect, and learn". In: arXiv preprint arXiv:2306.08640 (2023).

[63] Zongxin Yang, Guikun Chen, Xiaodi Li, Wenguan Wang, and Yi Yang. "Doraemongpt: Toward understanding dynamic scenes with large language models". In: arXiv preprint arXiv:2401.08392 (2024).

[64] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. "Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face". In: Advances in Neural Information Processing Systems 36 (2024).

[65] Hao Wen, Hongming Wang, Jiaxuan Liu, and Yuanchun Li. "Droidbot-gpt: Gpt-powered ui automation for android". In: arXiv preprint arXiv:2304.07061 (2023).

[66] Bryan Wang, Gang Li, and Yang Li. "Enabling conversational interaction with mobile ui using large language models". In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. 2023, pp. 1–17.

[67] Jingkang Yang, Yuhao Dong, Shuai Liu, Bo Li, Ziyue Wang, Chencheng Jiang, Haoran Tan, Jiamu Kang, Yuanhan Zhang, Kaiyang Zhou, et al. "Octopus: Embodied vision-language programmer from environmental feedback". In: arXiv preprint arXiv:2310.08588 (2023).

[68] Yiran Qin, Enshen Zhou, Qichang Liu, Zhenfei Yin, Lu Sheng, Ruimao Zhang, Yu Qiao, and Jing Shao. "Mp5: A multi-modal open-ended embodied system in minecraft via active perception". In: arXiv preprint arXiv:2312.07472 (2023).

[69] Clemens Eppner, Kaiyu Hang, Berk Calli, Zhe Su, Matthew T Mason, Aaron M Dollar, Alberto Rodriguez, Jeannette Bohg, Danica Kragic, Allison M Okamura, and Stanley T Birchfield. "Robust grasping and manipulation of novel objects using a

modular robot platform". In: IEEE Transactions on Robotics 37.3 (2021), pp. 948–967.

[70] Junke Wang, Dongdong Chen, Chong Luo, Xiyang Dai, Lu Yuan, Zuxuan Wu, and Yu-Gang Jiang. "Chatvideo: A tracklet-centric multimodal and versatile video understanding system". In: arXiv preprint arXiv:2304.14407 (2023).

[71] Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. "Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities". In: arXiv preprint arXiv:2305.11000 (2023).

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

[73] Guy Gafni, Or Patashnik, Gal Daniel, and Daniel Cohen-Or. "Make-a-scene: Scene-based text-to-image generation with human priors". In: arXiv preprint arXiv:2203.17089 (2022).

[74] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical text-conditional image generation with clip latents". In: arXiv preprint arXiv:2204.06125 (2022).

[75] Yue Wu, Xuan Tang, Tom M. Mitchell, and Yuanzhi Li. "Smartplay: A benchmark for llms as intelligent agents". In: arXiv preprint arXiv:2310.01557 (2023).

[76] Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. "Human-level play in the game of Diplomacy by combining language models with strategic reasoning". In: Science 378.6624 (2022), pp. 1067–1074.

[77] Richard J Chen, Ming Y Lu, Wei-Hung Weng, Tiffany Y Chen, Drew FK Williamson, Trevor Manz, Maha Shady, and Faisal Mahmood. "Multimodal Co-Attention Transformer for Survival Prediction in Gigapixel Whole Slide Images". In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021, pp. 4015–4025.

[78] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning". In: arXiv preprint arXiv:1711.05225 (2017).

[79] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison". In: arXiv preprint arXiv:1901.07031 (2019).

[80] Emil Arnold, Emil Rehder, Stefan Milz, Johannes Domel, Thomas Klotz, and Christoph Stiller. "A Survey on 3D Object Detection Methods for Autonomous Driving". In: arXiv preprint arXiv:1907.12622 (2019).

[81] Jiageng Mao, Yuxi Qian, Hang Zhao, and Yue Wang. "Gpt-driver: Learning to drive with gpt". In: arXiv preprint arXiv:2310.01415 (2023).

[82] Xingcheng Zhou, Mingyu Liu, Bare Luka Zagar, Ekim Yurtsever, and Alois C Knoll. "Vision language models in autonomous driving and intelligent transportation systems". In: arXiv preprint arXiv:2310.14414 (2023).

[83] Licheng Wen, Xuemeng Yang, Daocheng Fu, Xiaofeng Wang, Pinlong Cai, Xin Li, Tao Ma, Yingxuan Li, Linran Xu, Dengke Shang, Zheng Zhu, Shaoyan Sun, Yeqi Bai, Xinyu Cai, Min Dou, Shuanglu Hu, and Botian Shi. "On the Road with GPT-4V(ision): Early Explorations of Visual-Language Model on Autonomous Driving". In: arXiv preprint arXiv:2311.05332 (2023).

[84] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. "Multi-Modal Fusion Transformer for End-to-End Autonomous Driving". In: Conference on Computer Vision and Pattern Recognition (CVPR). 2021.

[85] OpenAI. ChatGPT. `https://openai.com/blog/chatgpt/`. 2023.

[86] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. "Llama-adapter v2: Parameter-efficient visual instruction model". In: arXiv preprint arXiv:2304.15010 (2023).

[87] Gen Luo, Yiyi Zhou, Tianhe Ren, Shengxin Chen, Xiaoshuai Sun, and Rongrong Ji. "Cheap and quick: Efficient vision-language instruction tuning for large language models". In: arXiv preprint arXiv:2305.15023 (2023).

[88] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. "Finetuned language models are zero-shot learners". In: arXiv preprint arXiv:2109.01652 (2021).

[89] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi. "Instructblip: Towards general-purpose vision-language models with instruction tuning". In: (2023). arXiv: 2305.06500 [cs.CV].

[90] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. "Instruction tuning with gpt-4". In: arXiv preprint arXiv:2304.03277 (2023).

[91] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. "Aligning Large Multi-Modal Model with Robust Instruction Tuning". In: arXiv preprint arXiv:2306.14565 (2023).

[92] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. "Scaling instruction-finetuned language models". In: arXiv preprint arXiv:2210.11416 (2022).

[93] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: European conference on computer vision. Springer. 2014, pp. 740–755.

[94] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models". In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, pp. 10684–10695.

[95] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, and Yunxin Jiao. "DALL-E 3: Bridging Vision and Language with Few-Shot Image Generation". In: (2023). URL: https://cdn.openai.com/papers/dall-e-3.pdf.

[96] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. "Visual chatgpt: Talking, drawing and editing with visual foundation models". In: arXiv preprint arXiv:2303.04671 (2023).

[97] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. "Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface". In: arXiv preprint arXiv:2303.17580 (2023).

[98] Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. "Gpt4tools: Teaching large language model to use tools via self-instruction". In: arXiv preprint arXiv:2305.18752 (2023).

[99] Xu Yang, Yongliang Wu, Mingzhuo Yang, Haokun Chen, and Xin Geng. Exploring Diverse In-Context Configurations for Image Captioning. 2023. arXiv: 2305.14800 [cs.CV].

[100] Noam Rotstein, David Bensaid, Shaked Brody, Roy Ganz, and Ron Kimmel. "FuseCap: Leveraging Large Language Models to Fuse Visual Data into Enriched Image Captions". In: arXiv preprint arXiv:2305.17718 (2023).

[101] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. "Mimic-it: Multi-modal in-context instruction tuning". In: arXiv preprint arXiv:2306.05425 (2023).

[102] Zhenfei Yin, Jiong Wang, Jianjian Cao, Zhelun Shi, Dingning Liu, Mukai Li, Lu Sheng, Lei Bai, Xiaoshui Huang, Zhiyong Wang, et al. "LAMM: Language-Assisted Multi-Modal Instruction-Tuning Dataset, Framework, and Benchmark". In: arXiv preprint arXiv:2306.06687 (2023).

[103] Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, et al. "M3 IT: A Large-Scale Dataset towards Multi-Modal Multilingual Instruction Tuning". In: arXiv preprint arXiv:2306.04387 (2023).

[104] Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. "LLaVAR: Enhanced Visual Instruction Tuning for Text-Rich Image Understanding". In: arXiv preprint arXiv:2306.17107 (2023).

[105] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. "Improved baselines with visual instruction tuning". In: arXiv preprint arXiv:2310.03744 (2023).

[106] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality". In: See https://vicuna. lmsys. org (accessed 14 April 2023) (2023).

[107] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. "Referitgame: Referring to objects in photographs of natural scenes". In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014, pp. 787–798.

[108] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. "Visual genome: Connecting language and vision using crowdsourced dense image annotations". In: International journal of computer vision 123 (2017), pp. 32–73.

[109] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. "Generation and comprehension of unambiguous object descriptions". In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 11–20.

[110] Drew A Hudson and Christopher D Manning. "Gqa: A new dataset for real-world visual reasoning and compositional question answering". In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, pp. 6700–6709.

[111] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. "MMBench: Is Your Multi-modal Model an All-around Player?" In: arXiv preprint arXiv:2307.06281 (2023).

[112] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. "Vizwiz grand challenge: Answering visual questions from blind people". In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, pp. 3608–3617.

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

[114] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. "MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models". In: arXiv preprint arXiv:2306.13394 (2023).

[115] IDEFICS. Introducing IDEFICS: An Open Reproduction of State-of-the-Art Visual Language Model. 2023. URL: https://huggingface.co/blog/idefics.

[116] Timo Schick, Jane Dwivedi-Yu, Roberto Dessı, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. "Toolformer: Language models can teach themselves to use tools". In: Advances in Neural Information Processing Systems 36 (2024).

[117] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. 2023. arXiv: 2307.16789 [cs.AI].

[118] Dídac Surís, Sachit Menon, and Carl Vondrick. "Vipergpt: Visual inference via python execution for reasoning". In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023, pp. 11888–11898.

[119] Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. "Mobile-Agent: Autonomous multi-modal mobile device agent with visual perception". In: arXiv preprint arXiv:2401.16158 (2024).

[120] Yang Liu, Xinshuai Song, Kaixuan Jiang, Weixing Chen, Jingzhou Luo, Guanbin Li, and Liang Lin. "Multimodal embodied interactive agent for cafe scene". In: arXiv preprint arXiv:2402.00290 (2024).

[121] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks". In: Advances in Neural Information Processing Systems 33 (2020), pp. 9459–9474.

[122] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. "Metagpt: Meta programming for multi-agent collaborative framework". In: arXiv preprint arXiv:2308.00352 (2023).

[123] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. "Seeclick: Harnessing gui grounding for advanced visual gui agents". In: arXiv preprint arXiv:2401.10935 (2024).

[124] Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, and Yan Lu. "Responsible task automation: Empowering large language models as responsible task automators". In: arXiv preprint arXiv:2306.01242 (2023).

[125] Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, et al. "Audiogpt: Understanding and generating speech, music, sound, and talking head". In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 38. 21. 2024, pp. 23802–23804.

[126] Dingyao Yu, Kaitao Song, Peiling Lu, Tianyu He, Xu Tan, Wei Ye, Shikun Zhang, and Jiang Bian. "Musicagent: An ai agent for music understanding and generation with large language models". In: arXiv preprint arXiv:2310.11954 (2023).

[127] Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. "Gpt4tools: Teaching large language model to use tools via self-instruction". In: Advances in Neural Information Processing Systems 36 (2024).

[128] Wei-Ge Chen, Irina Spiridonova, Jianwei Yang, Jianfeng Gao, and Chunyuan Li. "LLaVA-Interactive: An All-in-One Demo for Image Chat, Segmentation, Generation and Editing". In: (2023).

[129] Zhengyuan Yang*, Linjie Li*, Jianfeng Wang*, Kevin Lin*, Ehsan Azarnasab*, Faisal Ahmed*, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. "MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action". In: (2023).

[130] Yanda Li, Chi Zhang, Gang Yu, Zhibin Wang, Bin Fu, Guosheng Lin, Chunhua Shen, Ling Chen, and Yunchao Wei. "Stablellava: Enhanced visual instruction tuning with synthesized image-dialogue data". In: arXiv preprint arXiv:2308.10253 (2023).

[131] Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. "Discuss before moving: Visual language navigation via multi-expert discussions". In: arXiv preprint arXiv:2309.11382 (2023).

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

[133] Yuanchun Li, Ziyue Yang, Yao Guo, and Xiangqun Chen. "Droidbot: a lightweight ui-guided test input generator for android". In: 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C). IEEE. 2017, pp. 23–26.

[134] Sunjae Lee, Junyoung Choi, Jungjae Lee, Munim Hasan Wasi, Hojun Choi, Steven Y. Ko, Sangeun Oh, and Insik Shin. Explore, Select, Derive, and Recall: Augmenting LLM with Human-like Memory for Mobile Task 2024. arXiv: 2312.03003 [cs.HC].

[135] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. "Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond". In: arXiv preprint arXiv:2308.12966 (2023).

[136] Harrison Chase. LangChain. Oct. 2022. URL: https://github.com/langchain-ai/langchain.

[137] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. "Real-time Scene Text Detection with Differentiable Binarization". In: Proc. AAAI. 2020.

[138] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. "Grounding dino: Marrying dino with grounded pre-training for open-set object detection". In: arXiv preprint arXiv:2303.05499 (2023).

[139] Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. "AutoDroid: LLM-powered Task Automation in Android". In: (2024).

[140] Saydulu Kolasani. "Optimizing natural language processing, large language models (LLMs) for efficient customer service, and hyper-personalization to enable sustainable growth and revenue". In: Transactions on Latest Trends in Artificial Intelligence 4.4 (2023).

[141] Muhammad Usman Hadi, Qasem Al Tashi, Abbas Shah, Rizwan Qureshi, Amgad Muneer, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, et al. "Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects". In: Authorea Preprints (2024).

[142] Graham Todd, Sam Earle, Muhammad Umair Nasir, Michael Cerny Green, and Julian Togelius. "Level generation through large language models". In: Proceedings of the 18th International Conference on the Foundations of Digital Games. 2023, pp. 1–8.

[143] Shyam Sudhakaran, Miguel González-Duque, Matthias Freiberger, Claire Glanois, Elias Najarro, and Sebastian Risi. "Mariogpt: Open-ended text2level generation through large language models". In: Advances in Neural Information Processing Systems 36 (2024).

[144] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. "Large language models as zero-shot conversational recommenders". In: Proceedings of the 32nd ACM international conference on information and knowledge management. 2023, pp. 720–730.

[145] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al.

"Openassistant conversations-democratizing large language model alignment". In: Advances in Neural Information Processing Systems 36 (2024).

[146] Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. "Survey of vulnerabilities in large language models revealed by adversarial attacks". In: arXiv preprint arXiv:2310.10844 (2023).

[147] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Man Cheung, and Min Lin. "On evaluating adversarial robustness of large vision-language models". In: Advances in Neural Information Processing Systems 36 (2024).

[148] Yuan Gong and Christian Poellabauer. "Crafting adversarial examples for speech paralinguistics applications". In: arXiv preprint arXiv:1711.03280 (2017).

[149] Andre Kassis and Urs Hengartner. "Practical attacks on voice spoofing countermeasures". In: arXiv preprint arXiv:2107.14642 (2021).

[150] Xingyu Zhang, Xiongwei Zhang, Wei Liu, Xia Zou, Meng Sun, and Jian Zhao. "Waveform level adversarial example generation for joint attacks against both automatic speaker verification and spoofing countermeasures". In: Engineering Applications of Artificial Intelligence 116 (2022), p. 105469.

[151] Nicholas Carlini and David Wagner. "Audio adversarial examples: Targeted attacks on speech-to-text". In: 2018 IEEE security and privacy workshops (SPW). IEEE. 2018, pp. 1–7.

[152] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks". In: Proceedings of the 23rd international conference on Machine learning. 2006, pp. 369–376.

[153] Yi Xie, Zhuohang Li, Cong Shi, Jian Liu, Yingying Chen, and Bo Yuan. "Enabling fast and universal audio adversarial attack using generative model". In: Proceedings of the AAAI conference on artificial intelligence. Vol. 35. 16. 2021, pp. 14129–14137.

[154] Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, et al. "On generative spoken language modeling from raw audio". In: Transactions of the Association for Computational Linguistics 9 (2021), pp. 1336–1354.

[155] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. "Audiolm: a language modeling approach to audio generation.(2022)". In: arXiv preprint arXiv:2209.03143 (2022).

[156] Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu. "X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages". In: arXiv preprint arXiv:2305.04160 (2023).

[157] Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu, Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linquan Liu, et al. "On decoder-only architecture for speech-to-text and large language model integration". In: 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE. 2023, pp. 1–8.

[158] Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Junteng Jia, Yuan Shangguan, Ke Li, Jinxi Guo, Wenhan Xiong, Jay Mahadeokar, Ozlem Kalinli, et al. "Prompting large language models with speech recognition abilities". In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASS IEEE. 2024, pp. 13351–13355.

[159] Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. "LLaMA-Omni: Seamless Speech Interaction with Large Language Models". In: arXiv preprint arXiv:2409.06666 (2024).

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

[161] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: Advances in neural information processing systems 25 (2012), pp. 1097–1105.

[162] Paarth Neekhara, Shehzeen Hussain, Prakhar Pandey, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. "Universal adversarial perturbations for speech recognition systems". In: arXiv preprint arXiv:1905.03828 (2019).

[163] Yi Xie, Cong Shi, Zhuohang Li, Jian Liu, Yingying Chen, and Bo Yuan. "Real-time, universal, and robust adversarial attacks against speaker recognition systems". In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (IC IEEE. 2020, pp. 1738–1742.

[164] Corey Kereliuk, Bob L Sturm, and Jan Larsen. "Deep learning and music adversaries". In: IEEE Transactions on Multimedia 17.11 (2015), pp. 2059–2071.

[165] Audio-Based Jailbreak Attacks on Multi-modal LLMs. 2023. URL: https://mindgard.ai/resources/audio-based-jailbreak-attacks-on-multi-modal-llms?hs_amp=true.

[166] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. "Tvqa: Localized, compositional video question answering". In: arXiv preprint arXiv:1809.01696 (2018).

[167] Microsoft. Azure Cognitive Services Speech SDK. https://github.com/Azure-Samples/cognitive-services-speech-sdk. 2023.

[168] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Semantically equivalent adversarial rules for debugging NLP models". In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (volu 2018, pp. 856–865.

[169] Jason Wei and Kai Zou. "Eda: Easy data augmentation techniques for boosting performance on text classification tasks". In: arXiv preprint arXiv:1901.11196 (2019).

[170] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. "Is bert really robust? a strong baseline for natural language attack on text classification and entailment". In:

Proceedings of the AAAI conference on artificial intelligence. Vol. 34. 05. 2020, pp. 8018–8025.

[171] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. "Bert-attack: Adversarial attack against bert using bert". In: arXiv preprint arXiv:2004.09984 (2020).

[172] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: Text summarization branches out. 2004, pp. 74–81.