AudioAgent: Enhancing Task Performance through Modality-Driven Prompt Optimization

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have achieved remarkable progress in serving as controllers to interpret user instructions and select models for audio tasks. However, current LLMs, when selecting tools, only consider the textual input, neglecting valuable information within the audio modality that could aid in choosing appropriate tools. Due to the possible ambiguity of instructions, errors in selection are common. To this end, we introduce AudioAgent, a versatile and adaptable agent framework for audio fields. It is the first system that emphasizes audio comprehension and utilizes these information to autonomously refine user-provided prompt by one finetuned LLM. Through clearer instructions, AudioAgent empowers the controller to precisely select the best tools and enhances the performance of tasks. Our framework also enables users to freely register tools and utilize any LLM as the core controller. Both subjective and objective metrics validate the effectiveness of our work. Result samples are available at https://AudioAgentTool.github.io.

1 Introduction

In recent times, there has been rapid advancement in LLMs(Brown et al., 2020; Floridi and Chiriatti, 2020; Ouyang et al., 2022; Zhang et al., 2022b; Bai et al., 2023; Chowdhery et al., 2023; Touvron et al., 2023), which are capable of receiving multimodal input and executing a series of complex tasks based on user's textual instructions(Le Scao et al., 2022; Achiam et al., 2023; Bai et al., 2023; Team et al., 2023).

Within the realm of these models, numerous End-to-End Voice LLMs showcase their outstanding capability in processing the audio modality. For instance, present work(Borsos et al., 2023; Kharitonov et al., 2023; Wang et al., 2023a) develope a series of audio generation methods that focus on individual tasks. In pursuit of creating comprehensive united framework for multitasking

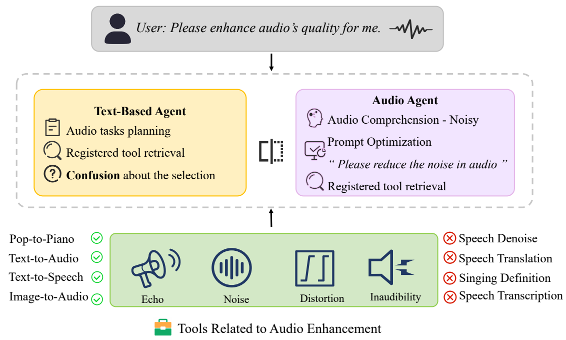


Figure 1: Upon receiving this instruction, Text-Based Agent is unable to determine the specific tool for audio enhancement due to multiple possibilities for audio's characteristics. Similarly, tasks marked with '×' are also affected by this uncertainty. Agent requires some audio features as a hint to choose, which is realized in the AudioAgent through modality comprehension.

through the LLM paradigm, endeavors like (Huang et al., 2023; Rubenstein et al., 2023; Yang et al., 2023a) have emerged. As multi-task frameworks continue to improve, users can now employ natural language to instruct the model in Qwen-Audio(Chu et al., 2023). These models take full advantage of the reasoning prowess and generalization abilities of LLMs. However, the overall number of tasks they can handle is still limited.

To fully harness the potential of LLMs and further expand the range of achieveable tasks, recent work has made great progress towards building agent-based LLMs(Du et al., 2021; Yang et al., 2023b; Qin et al., 2023; Ruan et al., 2023; Schick et al., 2024). Along this direction, several methods have been employed to enhance the tool's matching accuracy with textual instructions. Some focus on refining the tool's description to better clarify its functionality(Huang et al., 2024a; Shen et al., 2024), some narrow down the scope of tools before selection(Li et al., 2023a), others employ meticulous training on open-source LLMs to enhance their understanding of prompt(Ouyang et al., 2022).

However, the textual prompt easily leads to confusion. As illustrated in Figure 1, for the given textual prompt, text-based agent struggles to distinguish audio characteristics which determine the suitable tool. Actually, the modality comprehension process can play a significant role in this scenario. For example, if the audio contains noise, optimizing the prompt to "Please reduce the noise in audio" can assist the agent in making right choices.

In this work, we introduce AudioAgent, a comprehensive agent framework equipped with a versatile toolset to facilitate a wide range of audio tasks. It is the first agent framework that emphasizes audio comprehension and utilizes these information to autonomously refine user-provided prompt in content and expression, making it easier for agent-based LLMs to select the best tool.

To validate our approach, we construct a dataset mainly comprising two parts, which are ToolMMBench(Wang et al., 2024a) and one instruction set generated by GPT3.5-turbo with releated audio. We compare the optimized prompts achieved by AudioAgent across different types of instructions, demonstrating the importance of audio comprehension and prompt optimization in improving the accuracy of selection. Additionally, we utilize two baselines to validate the efficiency improvements through AudioAgent's optimal tool selection.

Overall, our contribution can be summarized in three main aspects as follows:

- Comprehension: AudioAgent distinguishes itself through its capacity to comprehend audio modality. Compared to previous agent models that focused solely on textual modality, we fully leverage this aspect to provide controllable features which improve the accuracy of selection.

- Optimization: AudioAgent offers one well fine-tuned LLM for prompt optimization, ensuring grammatical correctness and contextual richness in textual modality. The clearer instructions enable controller to select the best tools and enhance task performance across various scenario.

- Flexibility: AudioAgent enables users to flexibly register tools and utilize any LLM as controller. Furthermore, the component of modality comprehension and prompt optimization can be applicable to any agent framework.

2 Related work

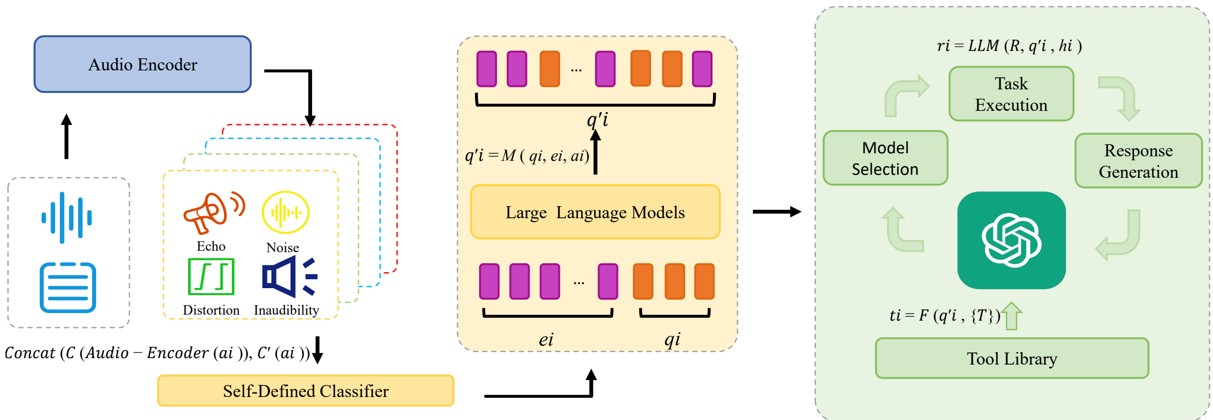
2.1 Large Language Models

Large Language Models have experienced rapid development in recent years, with some notable examples such as GPT4(Achiam et al., 2023), PaLM(Chowdhery et al., 2023), Qwen(Bai et al., 2023) and LLaMA(Touvron et al., 2023). Nowadays, there is a growing focus on leveraging the robust reasoning abilities of LLMs to tackle a wide array of multimodal challenges beyond text, such as audio, image and video tasks. Present research in this domain can be categorized into two main branches: One approach involves unified End-to-End LLMs to handle various tasks (Alayrac et al., 2022; Li et al., 2023b; Huang et al., 2024b). The other approach focuses on empowering LLMs to independently understand user prompt and utilize existing tools for solving multimodal tasks(Du et al., 2021; Qin et al., 2023; Ruan et al., 2023; Yang et al., 2023b; Schick et al., 2024).

2.2 Agent & Tool Learning

The use of LLMs as agents for executing complex tasks has gained increasing attention. Modelscope-Agent(Li et al., 2023a) deploys a flexible framework that allows any open-source LLMs to serve as the primary brain. Toolformer(Schick et al., 2024) pioneers the exploration of integrating LLM with external tools. HuggingGPT(Shen et al., 2024) broadens the spectrum of tasks by offering a wide array of models in HuggingFace. AudioGPT(Huang et al., 2024a) stands out as the first Agent tailored for audio. MLLM-Tool(Wang et al., 2024a) transforms audio into the MEL spectrum and then utilizes an image encoder to fine-tune a single-round dialogue Agent.

However, despite these advancements, most of these agent models still solely rely on the text-based understanding and reasoning ability of LLMs. The selection process is based on the user's textual instructions and the tool's description, making accuracy heavily dependent on the precision of the given text like the example in Figure 1. In other word, they only utilize audio for task execution part, thus lacking the incorporation of audio that could assist in enhancing the accuracy of tool selection. AudioAgent capitalizes on modality comprehension to extract information from the audio, enabling the creation of clear and grammatically correct prompt for LLM controller to understand and select the most suitable tool from toolset.



(a) Modality Comprehension **(b) Prompt Optimization** **(c) Task Execution and Dialogue**

Figure 2: Our network architecture: AudioAgent first receives the user's prompt and potentially existing audio file, and converts the prompt into grammatically correct new one with sufficient features of the audio. The LLM controller will then conduct historical retrieval, tool selection, task execution, and ultimately return the running results, which is then organized by the LLM controller to generate the final reply to the user.

3 Methods

3.1 Overview

The overall architecture of AudioAgent is in Figure 2, which consists of three parts: Modality Comprehension(C) in Figure 2(a), Prompt Optimization(M) in Figure 2(b), and Task Execution and Dialogue(L) with Tool Library(T) in Figure 2(c). The whole system can be defined as:

$$AudioAgent = (C, M, L, \{T_1^t\}) \tag{1}$$

When the user provides instructions and possible audio for processing, Modality Comprehension analyzes the audio, offering simple feature annotations. Subsequently, Prompt Optimization combines these annotations with user's textual prompt to generate a grammatically correct one with clear direction for the intended tools. Finally, in Task Execution and Dialogue part, AudioAgent utilizes the LLM controller to identify the tool in the set and organize the answer based on the execution result. If the user engages in multiple rounds of interaction, the results are retained in history for further iterations. The whole n-multiple dialogue can be formulated as the sequence:

$$D = \{(q_i, q_i^{'}, a_i, r_i)\} \tag{2}$$

The term q_i represents the query from the user and a_i represents the audio samples in this turn. Additionally, $q_i^{'}$ represents the optimized prompt obtained through AudioAgent from q_i and a_i. The r_i is target response generated for users.

3.2 Modality Comprehension

AudioAgent differs from existing agent models by enhancing audio comprehension, thus enabling a more comprehensive perception of task scenarios. Some dimensions, such as pitch and volume, do not significantly impact tool selection in audio fields. Therefore, these dimensions will not be considered in this context. Our primary focus is on dimensions that directly influence the selection of tools for audio tasks.

Dimensions such as the length of audio can be calculated using signal processing tools. As a result, we primarily focus on those cannot be directly measured. Initially, we draw inspiration from Qwen-Audio(Chu et al., 2023) and utilize an Audio Encoder module based on Whisper(Radford et al., 2023) to process the input audio. Within this module, audio is first resampled to 16,000 Hz, and an 80-channel log-magnitude Mel spectrogram representation is computed on 25-millisecond windows with a 10-millisecond stride. After that, the result undergos normalization, convolutional layers using GELU activation(Hendrycks and Gimpel, 2016), and Transformer layers employing pre-activation residual blocks(Child et al., 2019) to obtain the final representation.

Although Whisper is a pretrained multilingual translator under self-supervision, its encoded representation also contains rich information, and is capable of reconstructing the original speech(Gong et al., 2023; Zhang et al., 2023; Wang et al., 2024b). Qwen-Audio even utilize its embedding to infer discrete tokens in Voice LLM(Chu et al., 2023). So, leveraging this embedding to support the comprehension part is feasible.

The LSTM, known for effecitively capturing long-term dependencies and handling time-series data(Staudemeyer and Morris, 2019; Yu et al., 2019; Sherstinsky, 2020), is utilized to retain crucial information within sequences. By leveraging the representations extracted by the encoder, multiple classifiers based on LSTM are trained to provide annotations for audio. Specifically, for each sample, approximately 3 seconds of audio is randomly extracted, with the corresponding embedding serving as input for classifiers. Then, if e_i represents the result of the comprehension, it can be defined as:

$$e_i = C(Encoder(a_i)) \tag{3}$$

Also, we have designed interfaces that enable users to offer additional labels in text modality if necessary. For instance, if new label is to added, user can modify the e_i with new classifer C' as:

$$e_i = Concat(C(Encoder(a_i)), C'(a_i)) \tag{4}$$

3.3 Prompt Optimization

The current LLMs primarily rely on interpreting text when selecting tools. This approach may encounter issues such as grammatical disarray and lack of information in the initial prompt q_i, which significantly impacts tool selection accuracy. Leveraging the results of Modality Comprehension e_i and raw input q_i, Prompt Optimization component is trained to automatically refine the content and expression of user's instructions as better one with the finetuned LLM:

$$q_i^{'} = M(q_i, e_i, a_i) \tag{5}$$

For example, the feature "Neutral emotion, long time, English Language, Noisy feature" and the phrase "Please transcribed into text." will be transformed into "Please transcribe the long speech into English text.", which specifically points to the ASR tool designed for processing lengthy audio segments in English. Prompt Optimization needs to comprehend instruction and select labels to compose a new sentence.

In our experiments, we use GPT-3.5turbo to generate a training dataset as outlined in Section 4.1. Specifically, we use unlabeled sentences with grammatical errors and all audio labels as input, labeled sentences with correct grammar as target output. To accomplish the task of enriching content and refining expression in Equation 5, we finetune an open-source LLM. ChatGLM2-6B(Zeng et al., 2022), a bilingual LLM based on the General Language Model architecture, is selected. This model implements an efficient parameter P-tuning(Liu et al., 2021) method, reducing the number of parameters that need to be finetuned to the original 0.1%. Indeed, the flexibility of AudioAgent framework allows for any NLP model to complete the optimization process. We also develop interfaces through which users can select their own pretrained model to accomplish the prompt generation task.

3.4 Task and Dialogue Execution

When the above process acquires grammatically correct instructions containing sufficient information, they are able to provide a logical basis for LLM controller to select from the toolset. We then design a comprehensive framework capable of selecting any LLM as controller, supporting flexible registration of tools, and enabling multi-round dialogue as illustrated in Figure 2(c).

Specifically, during tool registration, users are required to provide the unique tool name, sufficient description, required parameters for the Tool Library(T). We have also prepared one toolset that includes nearly all of the current audio tasks, which can be seen in Appendix A. Regarding the controller, users can freely utilize any API interface of LLMs, which will receive the optimized prompt and retrieve the most suitable tools t_i in Tool Library base on text modality as:

$$t_i = F(q_i^{'}, \{T_1^t\}) \tag{6}$$

After obtaining the required tools, AudioAgent will automatically invoke these tools, provide their inputs, execute the tools, and obtain the output to return based on the tool's outcomes, user instructions $q_i^{'}$ and history h_i. Controller will continue to plan whether to call other tool to finish the sequential work. If another tool is needed, the process will be repeated, otherwise, the final comprehensive response is returned to the user. This turn of dialogue will also be encapsulated as history, enabling potential multi-round dialogue to utilize.

$$r_i = LLM(R, q_i^{'}, h_i) \tag{7}$$

4 Training and Evalutaion

4.1 Datasets

For modality comprehension, we combine datasets to train 4 classifers. We utilize multilingual audio for language identification; VCTK, M4singer, Audiocap for category recognition; ESD for emotion analysis; MS-SNSD and WSJ0+Reverb for characteristic discrimination. We provide details of these audio datasets in Appendix B.

For prompt optimization, our goal is to enable an incomplete, syntactically incorrect textual prompt to select appropriate audio labels from all extracted features and construct a correct one. Since there is no dataset available for training and validation, we design a method for generating a batch of data in pairs. Specifically, these pairs include 1) Raw: sentences without any audio feature labels; 2) Raw (err): Raw with specific grammatical errors; 3) GT: sentences with audio feature labels; 4) GT (err): GT with specific grammatical errors.

In short, the data for every audio task is generated separately. We first select the characteristics that each task needs to retain. For example, the ASR task needs [langauge] and [time]. Begin by creating a template with placeholders (such as *Transcribe the [time] speech into [language] text*). Then, replace the placeholders with labels or remove them to generate Raw and GT (such as *Transcribe the long speech into English text* for GT). Finally, introduce errors to obtain Raw(err) and GT(err) (such as *Transcrieb the speechs into txt* for GT(err)). After we get all template sentences for single task, GPT3.5-turbo is used to combine them to create new ones that require multiple tools (such as *Transcribe the [time] wav into [language] text. Then enhance the [feature] wav's quality*). We repeat the same process and finally get a total 3,000,000 pairs for training. More generation details and samples are in Appendix C.

For tool selection, we utilize two test sets. One is MLLM-Tool(Wang et al., 2024a), from which we enrich every prompt to four sentences with its original audio. The other is the dataset from prompt optimization, we manually select the correct tools and audio samples for every pair of prompt.

4.2 Evaluation Metrics

We mainly evaluate the agent framework through objective evaluation with some subjective evaluation part via Amazon Mechanical Turk. The explanation of metrics are as follows:

- **Feature accuracy**: When assessing modality comprehension, we directly employ the model's classification accuracy for the test set.
- **Grammar accuracy**: When evaluating the syntax error in the sentence, we utilize the independently trained grammar-checker as the arbiter(Warstadt et al., 2020).
- **Selection Accuracy**: We assess the accuracy of LLM in tool selection with accuracy, F1 and Edit Distance. The specific calculation method is detailed in the Appendix D.
- **Task Performance**: We compare the performance improvements through AudioAgent's optimal selection with other Agent and End-to-End Voice LLM, primarily employing the WER, BLEU and MOS.
- **Subjective evaluation**: We conduct informational integrity and MOS assessments. All process is held on the Amazon platform in English. Specifically, for integrity, the tester needs to select the answer from five options according to the tool's description and prompt. The accuracy is recorded as score. In MOS test, audio is rated scores on 1-5 scale. Details are in Appendix D.

4.3 Model Configurations

For Audio Encoder in Modality Comprehension, we utilize pre-trained Whisper which is a 32-layer Transformer model that includes two convolution down-sampling layers as a stem. The audio encoder is composed of about 640M parameter.

For ChatGLM2-6B in Prompt Optimization, it is finetuned with 4 2080Ti gpus for about one week and ends at about 20K step. Adam optimizer is used with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. The learning rate is 1×10^{-2} at beginning.

5 Results and Analysis

5.1 Modality Comprehension Result

We initially evaluate the accuracy of classifiers for Modality Comprehension. For dimensions that cannot be directly measured, the result is shown in the Table 1. The outcome further proves that the Audio Encoder contains rich information, and its embedding can be effectively used for highly accurate feature extraction. In our experiment, we primarily employ the LSTM structure to construct all classifiers, users can utilize other more complex structures to replace it if necessary. Furthermore, the feature dimension can be easily expanded by utilizing the Audio Encoder to train new classifiers or integrating other pre-trained models to provide labels, but we assume the feature utilized in AudioAgent is adequate for present audio tasks.

Type	Test Acc ↑
Language	98.64
Category	99.31
Emotion	84.02
Characteristic	97.77

Table 1: Results of Comprehension

5.2 Prompt Optimization Result

We test sentences Raw, Raw(err), GT, and GT(err) along with the results Ours obtained by AudioAgent from Raw(err) in grammar and integrity.

Following the assessment in Table 2, the scores of Ours closely align with the scores of GT in grammar tests, distinctly differing from sentences with incorrect grammar. Moreover, the subjective integrity test indicates that prompts with the correct labels guide the evaluators to select tools accurately, and Ours do the same. This suggests that the fine-tuned ChatGLM-6B model possesses the capability to correct grammatical errors and combine audio features into the context.

5.3 Model Selection Result

In this stage, we compare two scenarios: prompt for single tool selection and prompt for the sequential selection of multiple tools. To demonstrate the impact of correct grammar and comprehensive information on LLM's tool selection ability, we compare five types of prompt with totally 22 audio models for test, along with the open-source dataset MLLM-Tool. The details of LLM are in Appendix E and we use abbreviations here for simplicity.

	Obj. Syntax ↑	Subj. Integrity ↑
GT	83.96	95.49
GT(err)	25.67	92.91
Raw	79.87	34.74
Raw(err)	28.67	31.46
Ours	82.79	94.10

Table 2: Results of Prompt Optimization on Grammar and Integrity. Raw: text w/o audio labels. Gt: text w/ audio labels. (err): grammar error

5.3.1 Single Selection

Every item in the MLLM-Tool includes one prompt, one corresponding audio, and the tool to be selected. Its prompt comes with a full definition of audio, as it can be treated directly as GT. We use GPT3.5-turbo to remove the feature labels in GT and get Raw. Then, by introducing syntax errors, we obtain Raw(error) and GT(error). By modality comprehension and prompt optimization, Ours is the result from the audio and Raw(error).

It is found that although MLLM-Tool tries to combine the text prompt and audio to select the tool, its selection accuracy is affected when features in the text prompt are eliminated. LLM, which selects solely based on text mode, is more influenced by the feature labels in prompt for selection. This illustrates the importance of adequate audio feature labels in prompt for correct selection.

To further demonstrate the importance of feature labels in selection, we test on our own larger set. When considering the selection of a single model, the results also indicate that the accuracy of the prompt with features (GT, GT(err), Ours) significantly surpasses that of the prompt sentence without features (Raw, Raw(err)) in Table 4. This discrepancy arises because, in the absence of feature descriptions, LLMs lack the basis for selection and consequently exhibit reduced accuracy.

When considering the impact of grammatical correctness, it is observed that while a small number of LLMs, such as Claude, are less affected, the vast majority experience a notable decline in accuracy when encountering grammatical errors. This finding emphasizes the necessity of grammar correction in prompts to ensure accurate model selection. That is to say, although some higher-performing LLMs like Claude can better understand commands, even when they contain grammatical errors, due to the fact that the majority of these high-performing LLMs are currently closed-source or require payment, users can utilize open-source or affordable LLMs as controllers to ensure higher precision through prompt optimization.

5.3.2 Sequential Selection

For the sequential selection of multiple models, we choose Claude, GPT3.5-turbo, and Qwen, which exhibit the best performance in single-model selection as the basis. Then, we select the prompt involving multiple tasks and measure the characteristics of related audio samples to determine the correct tools and usage orders for generating the

	MLLM-Tool	Qwen	GPT3.5	Claude
GT	81.53 / 78.52	85.58 / 85.52	92.09 / 92.22	93.95 / 94.01
GT(err)	72.85 / 68.73	81.42 / 80.49	90.71 / 86.26	88.57 / 87.88
Raw	57.14 / 45.04	36.27 / 28.01	43.28 / 37.95	41.26 / 28.66
Raw(err)	56.91 / 44.72	35.71 / 27.04	37.14 / 21.25	38.57 / 29.49
Ours	79.84 / 75.32	83.26 / 83.04	91.62 / 91.45	95.34 / 95.32

Table 3: Selection Accuracy of Single Selection on MLLM-Tool's Dataset-Accuracy↑ / F1 ↑. Raw: text w/o audio labels. Gt: text w/ audio labels. (err): grammar error

	LLam2	Gemini	Vicuna	GPT3.5	Qwen	Claude
GT	40.00 / 23.01	92.86 / 92.31	52.86 / 40.84	92.85 / 92.31	94.43 / 88.53	97.14 / 96.79
GT(err)	34.29 / 19.39	72.86 / 68.66	35.71 / 20.51	82.86 / 78.07	82.86 / 78.07	94.29 / 93.73
Raw	34.29 / 17.87	30.00 / 19.03	17.14 / 9.91	44.29 / 36.10	35.71 / 27.09	40.00 / 26.66
Raw(err)	15.71 / 15.24	32.86 / 20.55	14.29 / 13.93	32.86 / 19.18	25.71 / 14.57	38.57 / 23.87
Ours	31.43 / 25.48	88.57 / 81.22	50.13 / 34.02	90.00 / 87.12	91.57 / 85.96	95.71 / 95.28

Table 4: Selection Accuracy of Single Selection-Accuracy↑ / F1↑. Raw: text w/o audio labels. Gt: text w/ audio labels. (err): grammar error

test set. More details are in Appendix C.

The results in Table 5 show that, although there is a certain degree of decline in overall correctness, the prominent pattern remains consistent with the selection of a single tool. Therefore, breaking down a long prompt into several shorter prompts with a specific sequence might be a method to improve precision in tool selection. We have identified this as one of our future research directions.

	Qwen	GPT3.5	Claude
GT	15.24 / 74.69	32.25 / 65.81	8.66 / 88.26
GT(err)	17.60 / 71.12	33.62 / 65.94	11.68 / 84.85
Raw	56.53 / 22.08	55.85 / 32.19	60.02 / 20.37
Raw(err)	57.28 / 21.52	58.25 / 28.53	60.76 / 18.85
Ours	16.57 / 73.73	32.90 / 67.33	9.28 / 87.18

Table 5: Selection Accuracy of Sequential Selection-ED ↓ / F1 ↑. Raw: text w/o audio labels. Gt: text w/ audio labels. (err): grammar error

5.4 Task and Dialogue Execution Result

After obtaining the required tool name, the agent framework will call the required tool, pass in parameters, collect the results and return them to the controller. The controller will then return content to the user based on the complete instructions, task description, model results, and interaction history, thereby completing a round of interaction. The complete process from receiving instructions to providing a response is illustrated in Figure 3. A more comprehensive dialogue from AudioAgent is shown in Appendix F.

Through the above process, AudioAgent is proved to enhance the accuracy of model selection, thereby significantly improving the efficiency of the task. To illustrate this, we compare the results by AudioAgent's optimal tool selection with those of two baseline models which can be instructed with natural language. Specifically, HuggingGPT is the typical Agent framework before and Qwen-Audio is the End-to-End Voice Large Language Model. Here, the model's input, as depicted in Figure 3, is Raw(err) without directive features.

The results presented in Table 6 indicate that, when compared to our AudioAgent, HuggingGPT performs poorly in task execution due to its lack of specific model discrimination ability. For instance, in transcription tasks, HuggingGPT consistently invokes English transcription tools as it cannot discern languages, resulting in nearly no useful output for audio inputs in other languages.

On the other hand, Qwen-Audio only needs to discern the task label to automatically execute the corresponding task. For example, if it identifies an Translation task, Qwen-Audio utilizes the unified framework for inference. However, it mainly generates outputs for the text modality and cannot

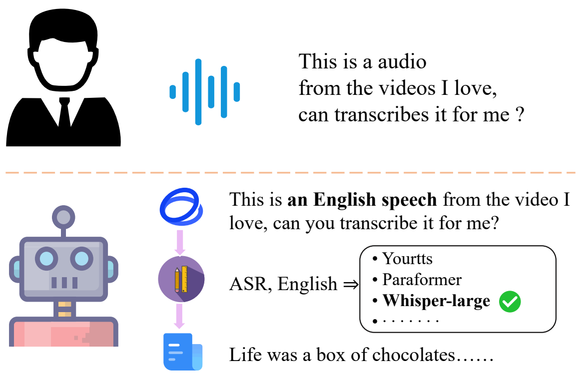


Figure 3: The Process of One-turn Dialogue

fulfill tasks requiring audio modality outputs, such as audio enhancement. In contrast, AudioAgent not only generates multimodal output but also achieves comparable efficiency to Voice LLMs in capability by precisely selecting and utilizing multiple individually trained models.

Model	ASR↓	ST↑	AE↑
HuggingGPT	43.2	0.1	3.53 ± 0.10
Qwen-Audio	4.0	28.8	/
Ours	3.4	31.2	4.10 ± 0.06

Table 6: Results of Task with WER, BLEU and MOS. HuggingGPT is Agent framework, Qwen-Audio is End-to-End Voice LLM.

6 Ablation Study

To demonstrate the direct impact of the feature dimensions on the accuracy of model selection, we conduct Ablation tests on Claude, GPT-3.5turbo, and Qwen, three best LLMs in above Experiments. The dataset used here is the same as that used in the Single Selection and Sequential Selection sections above, where we manually select the text instructions built by GPT3.5-turbo for each task and the audio corresponding to that task. But we will control the number of labels before prompt optimization to get different optimized prompt for the LLM controller to carry out tool selection.

Our findings, as depicted in Table 7, reveal a direct positive correlation between the accuracy of model selection and the number of features utilized. This emphasizes the critical role of modality comprehension and prompt optimization in guiding decision-making for LLMs. Normally, precise feature definitions contribute to a more robust logical foundation for LLMs, enabling them to make more accurate judgments.

In the future, we also plan to expand modal understanding to encompass additional modalities like image and video, further enhancing AudioAgent's capabilities.

	LLMs	Single Task		Sequential Task	
		Acc↑	F1↑	ED↓	F1↑
30%	Qwen	47.31	39.23	43.21	53.42
	GPT3.5	39.12	30.14	48.15	51.08
	Claude	48.93	31.76	40.12	54.13
50%	Qwen	68.12	62.38	38.17	59.94
	GPT3.5	59.83	55.43	61.63	23.21
	Claude	67.32	63.76	33.48	63.39
80%	Qwen	78.61	71.97	33.46	62.58
	GPT3.5	76.73	69.02	39.72	56.87
	Claude	85.21	79.01	10.31	85.63
100%	Qwen	91.57	85.96	17.60	71.12
	GPT3.5	90.00	87.12	33.62	67.33
	Claude	95.71	95.28	9.28	87.18

Table 7: Selection Accuracy of Ablation Study on Ours

7 Conclusion

In this paper, we introduce AudioAgent, an agent framework designed to address the common ambiguity in textual instructions and the poor task efficiency in execution for audio fields. In our method, AudioAgent comprehends the characteristics of the audio modality to optimize the prompt, rather than solely using audio as the tool's input. Therefore, it enables the controller to accurately select the optimal model for each type of task within a extensive toolset. Moreover, AudioAgent also employs a straightforward and flexible framework, enabling users to freely register tools and utilize any LLM's API as the controller. Both subjective and objective evaluations have demonstrated the effectiveness of our work in selection and execution. Additionally, relying on the exceptional scalability of our framework, we intend to extend its application to additional modalities such as images and videos in the future. In other words, through modality comprehension and prompt optimization, our framework can enhance the precision of tool selection across different modalities, leading to a unified multimodal Agent Framework. We hope AudioAgent will introduce a novel research paradigm in the realm of AI Agents.

8 Limitation

AudioAgent introduces a novel approach for scheduling tools in the audio domain. However, there are still several areas that require attention and enhancement: 1) Length Limitation: The maximum token limit is currently still determined by the Large Language Models used in AudioAgent. This limitation may impact multi-turn conversations because of the history in memory and calls for Prompt Optimization to condense user instructions. 2) Expression fluency: While we have devised a comprehensive process framework and enhanced selection accuracy in the interactive segment, the fluency of expression in interaction still relies on LLMs. Employing finetuned open source LLMs on Audio-related dataset tends to be beneficial. 3)Time Consumption: Improving selection accuracy entails modality comprehension and prompt optimization for input audio. Although the processing time is not extensive, it unavoidably extends user waiting time. In the future, we will further research lighter modal understanding components.

9 Potential Risks

AudioAgent reduces the barriers to entry for jobs within the audio domain, potentially leading to unemployment among professionals in related fields, such as speech engineering. Moreover, it could facilitate misuse within the vocal domain, providing illicit actors with tools to inflict harm upon society.

References

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

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

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

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

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

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

Felix Burkhardt, Johannes Wagner, Hagen Wierstorf, Florian Eyben, and Björn Schuller. 2023. Speech-based age and gender prediction with transformers. In *Speech Communication; 15th ITG Conference*, pages 46–50. VDE.

Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti. 2022. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning*, pages 2709–2720. PMLR.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Jongho Choi and Kyogu Lee. 2023. Pop2piano: Pop audio-based piano cover generation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*.

Harishchandra Dubey, Vishak Gopal, Ross Cutler, Ashkan Aazami, Sergiy Matusevych, Sebastian Braun, Sefik Emre Eskimez, Manthan Thakker, Takuya Yoshioka, Hannes Gamper, and Robert Aichner. 2022. Icassp 2022 deep noise suppression challenge. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9271–9275.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694.

Zhifu Gao, Shiliang Zhang, Ming Lei, and Ian McLoughlin. 2010. Universal asr: Unifying streaming and non-streaming asr using a single encoder-decoder model. In *arXiv preprint arXiv:2010.14099*.

Zhifu Gao, Shiliang Zhang, Ian McLoughlin, and Zhijie Yan. 2022. Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition. *arXiv preprint arXiv:2206.08317*.

Yuan Gong, Sameer Khurana, Leonid Karlinsky, and James Glass. 2023. Whisper-at: Noise-robust automatic speech recognizers are also strong general audio event taggers. *arXiv preprint arXiv:2307.03183*.

Josif Grabocka and Lars Schmidt-Thieme. 2018. Neuralwarp: Time-series similarity with warping networks. *arXiv preprint arXiv:1812.08306*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

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

Rongjie Huang, Chunlei Zhang, Yongqi Wang, Dongchao Yang, Luping Liu, Zhenhui Ye, Ziyue Jiang, Chao Weng, Zhou Zhao, and Dong Yu. 2023. Make-a-voice: Unified voice synthesis with discrete representation. *arXiv preprint arXiv:2305.19269*.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, et al. 2024b. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems*, 36.

Marek Kadlčík, Adam Hájek, Jürgen Kieslich, and Radosław Winiecki. 2023. A whisper transformer for audio captioning trained with synthetic captions and transfer learning. *Preprint*, arXiv:2305.09690.

Darioush Kevian, Usman Syed, Xingang Guo, Aaron Havens, Geir Dullerud, Peter Seiler, Lianhui Qin, and Bin Hu. 2024. Capabilities of large language models in control engineering: A benchmark study on gpt-4, claude 3 opus, and gemini 1.0 ultra. *arXiv preprint arXiv:2404.03647*.

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, and Neil Zeghidour. 2023. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *Transactions of the Association for Computational Linguistics*, 11:1703–1718.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. 2019. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 119–132.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model.

Chenliang Li, Hehong Chen, Ming Yan, Weizhou Shen, Haiyang Xu, Zhikai Wu, Zhicheng Zhang, Wenmeng Zhou, Yingda Chen, Chen Cheng, et al. 2023a. Modelscope-agent: Building your customizable agent system with open-source large language models. *arXiv preprint arXiv:2309.00986*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023c. Towards general text embeddings with multi-stage contrastive learning. *Preprint*, arXiv:2308.03281.

Haohe Liu, Ke Chen, Qiao Tian, Wenwu Wang, and Mark D Plumbley. 2024. Audiosr: Versatile audio super-resolution at scale. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1076–1080. IEEE.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Thorsten Müller and Dominik Kreutz. 2021. Thorsten-voice dataset 2021.02. Please use it to make the world a better place for whole humankind.

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

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

Chandan KA Reddy, Ebrahim Beyrami, Jamie Pool, Ross Cutler, Sriram Srinivasan, and Johannes Gehrke. 2019. A scalable noisy speech dataset and online subjective test framework. *Proc. Interspeech 2019*, pages 1816–1820.

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

Julius Richter, Simon Welker, Jean-Marie Lemercier, Bunlong Lay, and Timo Gerkmann. 2023. Speech enhancement and dereverberation with diffusion-based generative models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2351–2364.

Jingqing Ruan, Yihong Chen, Bin Zhang, Zhiwei Xu, Tianpeng Bao, Guoqing Du, Shiwei Shi, Hangyu Mao, Xingyu Zeng, and Rui Zhao. 2023. Tptu: Task planning and tool usage of large language model-based ai agents. *arXiv preprint arXiv:2308.03427*.

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. 2023. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*.

Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. 2018. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 351–355. IEEE.

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

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

Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306.

Ralf C Staudemeyer and Eric Rothstein Morris. 2019. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. 2021. Attention is all you need in speech separation. In *ICASSP 2021*.

Cem Subakan, Mirco Ravanelli, Samuele Cornell, François Grondin, and Mirko Bronzi. 2023. Exploring self-attention mechanisms for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2169–2180.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

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

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. 2023a. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Chenyu Wang, Weixin Luo, Qianyu Chen, Haonan Mai, Jindi Guo, Sixun Dong, Xiaohua (Michael) Xuan, Zhengxin Li, Lin Ma, and Shenghua Gao. 2024a. Mllm-tool: A multimodal large language model for tool agent learning. *arXiv preprint arXiv:2401.10727*.

Dong Wang and Xuewei Zhang. 2015. Thchs-30: A free chinese speech corpus. *arXiv preprint arXiv:1512.01882*.

Siyin Wang, Chao-Han Yang, Ji Wu, and Chao Zhang. 2024b. Can whisper perform speech-based in-context learning? In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13421–13425. IEEE.

Zhong-Qiu Wang, Samuele Cornell, Shukjae Choi, Younglo Lee, Byeong-Yeol Kim, and Shinji Watanabe. 2023b. Tf-gridnet: Integrating full-and sub-band modeling for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Alex Warstadt, Yian Zhang, Haau-Sing Li, Haokun Liu, and Samuel R Bowman. 2020. Learning which features matter: Roberta acquires a preference for linguistic generalizations (eventually). *arXiv preprint arXiv:2010.05358*.

Simon Welker, Julius Richter, and Timo Gerkmann. 2022. Speech enhancement with score-based generative models in the complex STFT domain. In *Proc. Interspeech 2022*, pages 2928–2932.

Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. 2023. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136.

Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al. 2019. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92).

Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, et al. 2023a. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*.

Hui Yang, Sifu Yue, and Yunzhong He. 2023b. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint arXiv:2306.02224*.

Guy Yariv, Itai Gat, Sagie Benaim, Lior Wolf, Idan Schwartz, and Yossi Adi. 2023. Diverse and aligned audio-to-video generation via text-to-video model adaptation. *Preprint*, arXiv:2309.16429.

Zhenhui Ye, Ziyue Jiang, Yi Ren, Jinglin Liu, Jinzheng He, and Zhou Zhao. 2023. Geneface: Generalized and high-fidelity audio-driven 3d talking face synthesis. *arXiv preprint arXiv:2301.13430*.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Lichao Zhang, Ruiqi Li, Shoutong Wang, Liqun Deng, Jinglin Liu, Yi Ren, Jinzheng He, Rongjie Huang, Jieming Zhu, Xiao Chen, et al. 2022a. M4singer: A multi-style, multi-singer and musical score provided mandarin singing corpus. *Advances in Neural Information Processing Systems*, 35:6914–6926.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022b. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2023. Speechtokenizer: Unified speech tokenizer for speech large language models. *arXiv preprint arXiv:2308.16692*.

Shengkui Zhao, Bin Ma, Karn N. Watcharasupat, and Woon-Seng Gan. 2022. Frcrn: Boosting feature representation using frequency recurrence for monaural speech enhancement. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9281–9285.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Kun Zhou, Berrak Sisman, Rui Liu, and Haizhou Li. 2022. Emotional voice conversion: Theory, databases and esd. *Speech Communication*, 137:1–18.

A Tool Details

When registering the tool, we design a well-crafted interface. It mainly consists of two modules. One primarily interacts with LLM. This part asks users to provide a detailed description of the function of tools and parameters. Audio-Agent will select the most suitable tool based on the optimized instructions and the functional descriptions.

Once the tool is determined, the LLM controller extracts parameter content from the instructions based on the parameter's description. Then, the next part will run the pre-trained model based on the parameter and return the result for LLM controller to generate the response.

In tool registration, the tool's function and the every parameter's description should be detailed but different from other tool in the toolset, it makes LLM more easily to select the most suitable one. Some examples of the registration are in Table 8.

To register tools as described above, we have prepared a detailed tool set. This tool set contains all the tasks we can think of in the audio field. If the input mode is audio, AudioAgent will understand the audio and optimize the prompt's content and syntax; If the input does not contain audio, the syntax of the prompt is optimized and the agent's process proceeds normally. The detail of the toolset is in Table 9.

B Dataset Statistics

In the modality comprehension section, we primarily use the following datasets to train the comprehension component. We do not use all the data because the Audio Encoder has rich information and can efficiently train classifiers with high accuracy. Specifically, we calibrate a set of data for each classifier, divided by dividers in Table 10.

C Dataset Construction

We use GPT3.5-turbo(Wu et al., 2023) to construct training data for Prompt Optimization part. Specifically, we set multiple task scenarios, generate sentence templates and replace the placeholders in the templates with keywords.

For instance, if the task scenario is in an Transcription environment, then the effective labels are the language and time. We use GPT3.5-turbo to first generate a template sentence with these label placeholder such as [language] and [time]. For instance, we get "Transcribe the [time] speech into [language] text". Next, we list all audio labels combination like "Long time; Chinese language; Angry emotion; Noisy feature; Speech type" and replace the placeholder with true labels to get GT, like "Transcribe the long speech into Chinese text". Since there are many combinations of such labels, one template sentence can be used multiple times.

Then, remove the locators from the template sentence, and we get the grammatically correct but unlabeled sentence (Raw). Making spelling, tense, etc. errors for Raw and GT gives Raw(error) and GT(error). In summary, a template sentence can be combined with different tags to enrich our data set.

Once we have these template sentences designed for a single task, we use GPT3.5turbo to merge pairwise into multi-task sentence combinations and repeat the same process for creation. Specific examples of the data can be seen in the Table 12.

After we get the prompt for each task scenario, we pick the appropriate audio to build the tool-selected test set. For example, for individual tool selection, ASR tasks use ASR's prompt and LibriTTS; Audio Enhancement tasks use AE's prompt and MS-SNSD. This builds the prompt and audio correspondence. For multi-tool selection, we first pick the prompt for multi-task. Then we manually pick audio samples, test its multi-label features with the classifier, and specify the correct tools and their sequence in usage by the prompt.

D Evalution Metrics

Here we supplement some details regarding the evaluation metrics.

D.1 Grammar

For grammar measurement, we utilize the open-source tool available on HuggingFace. This tool is based on the FacebookAI/roberta-base model(Warstadt et al., 2020). We present online example in Figrue 4. Through experiments, this tool can rapidly discern the correctness of word spelling and can also perceive grammatical details such as errors in tense, which is useful in our experiment.

D.2 Selection

For model selection in our testing, we mainly utilize the F1 score, ED, and Accuracy as the three primary metrics.

In multi-class classification problems, the F1 score is a commonly used performance metric that comprehensively considers a model's precision and recall. For datasets with imbalanced class distributions, the F1 score better reflects the model's

Tool Name: Yourtts(Casanova et al., 2022)
Tool Description: Convert the text into speech, provide the prompt wav as the speaker if needed. Parameter-Text: The text to be converted into the speech. Parameter-Prompt: The path of ossible wav to be the prompt. If user don't provide, be 'None'
Tool Name: Whisper-large-v2(Radford et al., 2023)
Tool Description: Translate the speech in language A into English text. Parameter-Language_A: The language of the speech file. Parameter-Path: The necessary path of the speech.
Tool Name: Chest_falsetto
Tool Description: Define the characteristic of the given song. Parameter-Path: The necessary path of the song.
Tool Name: Speech_frcrn_ans_cirm_16k(Dubey et al., 2022; Zhao et al., 2022)
Tool Description: Reduce the noise in the noisy wav when executing audio enhancement. Parameter-Path: The necessary path of the noisy wav file.
Tool Name: Make-An-Audio(Huang et al., 2023)
Tool Description: Comprehend the image and create the relevant audio based on it. Parameter-Path: The necessary path of the image.

Table 8: Example of Tool Registration

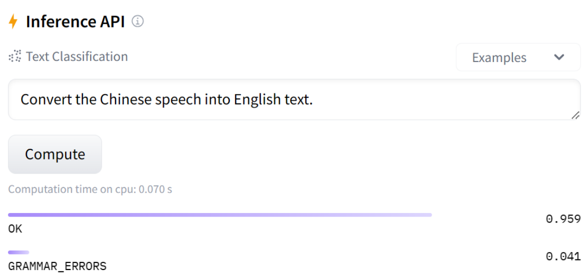


Figure 4: The Test on Sentence with Right Grammar

performance. We use the F1 metric to measure the number of correct tools selected by the LLM in both single and multiple selections.

When multiple models need to be sequentially selected, we also use the Edit Distance (ED) metric. Edit Distance, also known as Levenshtein distance, measures the similarity between two strings. It indicates the number of operations—insertions, deletions, and substitutions—needed to transform one string into another. This distance is useful for comparing the similarity between two strings. We use it to compare the format of the tool organization provided by LLM with the standard answer to gauge the correctness of our selection, which is also utilized in HuggingGPT(Shen et al., 2024).

Accuracy directly measures the proportion of correctly selected tools. It provides a straightforward assessment of the number of correctly chosen tools. We use this metric to visualize the results of a single tool selection when testing it. Overall, we assume the three metrics can demonstrate the selection result of the LLM model.

D.3 Subjective Metric

In the subjective assessment, we primarily submit the results to Amazon Mechanical Turk for testing.

To Integrity, since the instructions convey the user's intent, we have evaluators read the instructions to test their ability in selecting the correct results. Examples are in Table 13. Every question is rated by 4 testers and we design 50 question for Raw, Raw(err), GT, GT(err) and Ours. We believe that this can be used as an indicator of whether the instructions convey the necessary information for selection and how they influences the capability of the LLM in choosing the right tools.

For the performance improvement brought about by precise tool selection, we also conduct MOS evaluations for the audio quality enhancement with 95% confidence intervals (CI). We ask the testers to examine the audio quality and naturalness and ignore the content. We have 100 items in all and each data item is rated by 4 testers. The testers rate

Input	Output	Output	Model
Text-to-Speech	Text&Audio	Audio	Yourtts(Casanova et al., 2022)
Text-to-Audio	Text	Audio	Make-An-Audio(Huang et al., 2023)
Speech Transcription	Audio	Text	Paraformer(Gao et al., 2022), UniASR(Gao et al., 2010)...
Speech Translation	Audio	Text	Whisper-large-v2(Radford et al., 2023)
Audio Captioning	Audio	Text	Whisper-large-v2-audio-captioning(Kadlčík et al., 2023)
Speaker Verification	Audio	Text	Wav2vec2-large-robust-24-ft-age-gende(Burkhardt et al., 2023)
Singing Definition	Audio	Text	Chest_falsetto
Talking Head Synthesis	Audio	Video	GeneFace(Ye et al., 2023)
Audio-to-Video Generation	Audio	Video	TempoTokens(Yariv et al., 2023)
Speech Enhancement	Audio	Audio	AudioSR(Liu et al., 2024)
Speech Denoise	Audio	Audio	Speech_frcrn_ans_cirm_16k(Dubey et al., 2022; Zhao et al., 2022)
Speech Dereverberation	Audio	Audio	Sgmse(Welker et al., 2022; Richter et al., 2023)
Mono-to-Binaural	Audio	Audio	NeuralWarp(Grabocka and Schmidt-Thieme, 2018)
Pop-to-Piano	Audio	Audio	Pop2piano(Choi and Lee, 2023)
Audio Source Separation	Audio	Audio	Sepformer-libri3mix(Subakan et al., 2021, 2023)
Speech Separation	Audio	Audio	TF-GridNet(Wang et al., 2023b)
Image-to-Audio	Image	Audio	Make-An-Audio(Huang et al., 2023)

Table 9: Models for various audio processing tasks we have prepared, user can register tools into the original toolset freely and easily. We support any modality as input and output.

scores on 1-5 scales and are paid $8 hourly. The MOS evaluation is shown in Figure 5.

E LLM in Test

In our Tool Selection tests, we abbreviate the LLM's detail information. Here, we provide additional explanation for it in Table 11.

We find that the better the performance of the LLM as a controller selection model, the less it is influenced by syntactic instructions. However, regardless of the type of LLM, it cannot accurately select tools for instructions with incomplete content information. Furthermore, when the task scenario is clear, such as audio quality enhancement, the LLM cannot determine whether it should increase the sampling rate or remove noise. So, it consistently selects the same type of tool. This demonstrates the logic behind the LLM's tool selection and highlights the necessity of providing instructions with detailed information.

F Chat of LLM

In order to illustrate how the multi-round interaction works, we test AudioAgent and record the experimental results completely. The interaction can be shown in the Figure 6. It can be seen that AudioAgent can select the right tool according to the user's instructions, and Penguin can successfully complete multiple rounds of dialogue interaction.

Dataset	Hours
Language	
En:LibriTTS(Zen et al., 2019)	40.8
Zh:Thchs30(Wang and Zhang, 2015)	34.2
De:Thorsten(Müller and Kreutz, 2021)	27.1
Ja:Common Voice Corpus 8.0(Ardila et al., 2019)	41.0
Fr:Common Voice Delta Segment 11.0(Ardila et al., 2019)	39.0
Category	
Speech:VCTK(Yamagishi et al., 2019)	30.0
Song:M4Singer(Zhang et al., 2022a)	29.8
Audio:Audiocap(Kim et al., 2019)	35.0
Emotion	
ESD(Zhou et al., 2022)	29.1
Characteristic	
MS-SNSD(Reddy et al., 2019)	30.0
WSJ0+Reverb(Scheibler et al., 2018)	32.1

Table 10: Dataset in Modality Comprehension

Used	Details
NTE	Natural Text Embedding(Li et al., 2023c)
LLam2	Llama2-Chat-13B(Touvron et al., 2023)
Gemini	Gemini1.5-pro(Reid et al., 2024)
Vicuna	Vicuna-33b(Zheng et al., 2023)
GPT3.5	Gpt3.5-turbo(Wu et al., 2023)
Qwen	Qwen1.5-32b-chat(Bai et al., 2023)
Claude	Claude 3 Sonnet(Kevian et al., 2024)

Table 11: Details about the LLM in Test

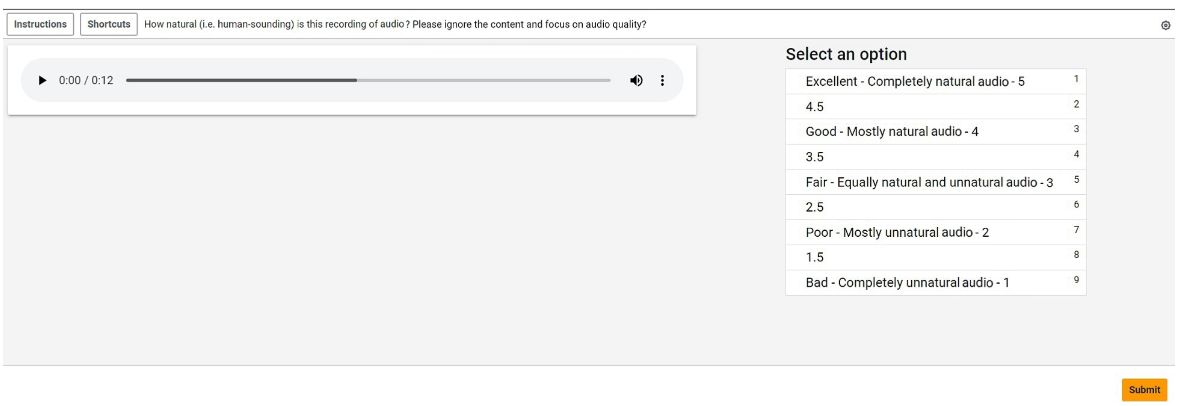


Figure 5: The test on Mos Quality

Speech Transcription Template: Could you transcribe the [time] dialogue and translate it into [language] for me? Raw: Could you transcribe the dialogue and translate it for me? Raw(err): Could you transcribing the dialogues and for me translates it? GT: Could you transcribe the long dialogue and translate it into Japanese for me? GT(err): Could you transcribing the long dialogues and translating it into Japanese for me?
Speech Translation Template: Could you translate the [language] speech into Spanish for me? Raw: Could you translate the speech into English for me? Raw(err): Translates the speech into English for I, Could you? GT: Could you translate the Chinese speech into English for me? GT(err): Could you translating the Chinese speeches into English for me?
Video Generation Template: Create the video associated with the [type]'s melody. Raw: Create the video associated with the melody. Raw(err): The video, create, associating with the melody. GT: Create the video associated with the speech's melody. GT(err): Create the video associateing with the speech melody.
Singing Definition Template: Comprehend the characteristics of the [emotion] [type]. Raw: Comprehend the characteristics of the wav. Raw(err): Comprehends the wav characteristics. GT: Comprehend the characteristics of the sad song. GT(err): The sad characteristics,comprehending of the song.
Audio Enhancement Template: Refine the quality of the [feature] recording. Raw: Refine the quality of the recording. Raw(err): The recoring, refining the quality. GT: Refine the quality of the noisy recording. GT(err): The noisy recordings, refined the quality of.
Sentence Combination 1 Template: Translate the [language] [type] into English text and define it. Raw: Translate the wav into English text and define it. Raw(err): Translates the wav into Englih text and define it. GT: Translate the French speech into English text and define it. GT(err): Translte the French speech into English txt and define it.
Sentence Combination 2 Template: Create the video related to the [type]'s melody. Then, enhance the [feature] recording. Raw: Create the video related to the melody. Then, enhance the recording. Raw(err): Create the video related to the meloyd. Thens, enhancing the recording. GT: Create the video related to the speech's melody. Then, enhance the echoing recording. GT(err): Create the video erlated to the speech's melody. Then, Enhance the Echoing recording.

Table 12: Example of our dataset. We only need to enumerate the keyword combinations and make corresponding substitutions to get a complete prompt. Raw: text w/o audio labels. Gt: text w/ audio labels. (err): grammar error

Examples on Single Tool Selection
Prompt:Transcribing the audios into text.(Raw(err))
A: UniASR:Japanese ASR for short speech B: Paraformer:Chinese ASR for long speech C: Yourtts:Text to Speech with prompt D: Sgmse:Reduce the audio's noise. E: Chest_falsetto:Define the song.
Prompt:Transcribe the long speech into Japanese text.(Ours)
A: UniASR:Japanese ASR for short speech B: Paraformer:Chinese ASR for long speech C: Yourtts:Text to Speech with prompt D: Sgmse:Reduce the audio's noise. E: Chest_falsetto:Define the song.
Prompt:Enhancig the audio's quality.(Raw(err))
A: AudioSR:Improve audio's sampling rate B: Sgmse:Reduce the audio's noise. C: Chest_falsetto:Define the song. D: Make-an-Audio:Text to Audio. E: TF-GridNet:Speech Seperation.
Prompt:Enhance the noisy audio's quality, please.(Ours)
A: AudioSR:Improve audio's sampling rate B: Sgmse:Reduce the audio's noise. C: Chest_falsetto:Define the song. D: Make-an-Audio:Text to Audio. E: TF-GridNet:Speech Seperation.
Prompt:Defning the wav features.(Raw(err))
A: Chest_falsetto:Define the characteristic of song. B: Wav2vec2(...):Define the speaker's feature. C: Whisper-large-v2:Translate the speech into other language. D: TF-GridNet:Speech Seperation. E: Chest_falsetto:Define the song.
Prompt:Can you define the song's features?(Ours)
A: Chest_falsetto:Define the characteristic of song. B: Wav2vec2(...):Define the speaker's feature. C: Whisper-large-v2:Translate the speech into other language. D: UniASR:Japanese ASR for short speech E: TF-GridNet:Speech Seperation.

Table 13: The Example of Subjective Test on Selection

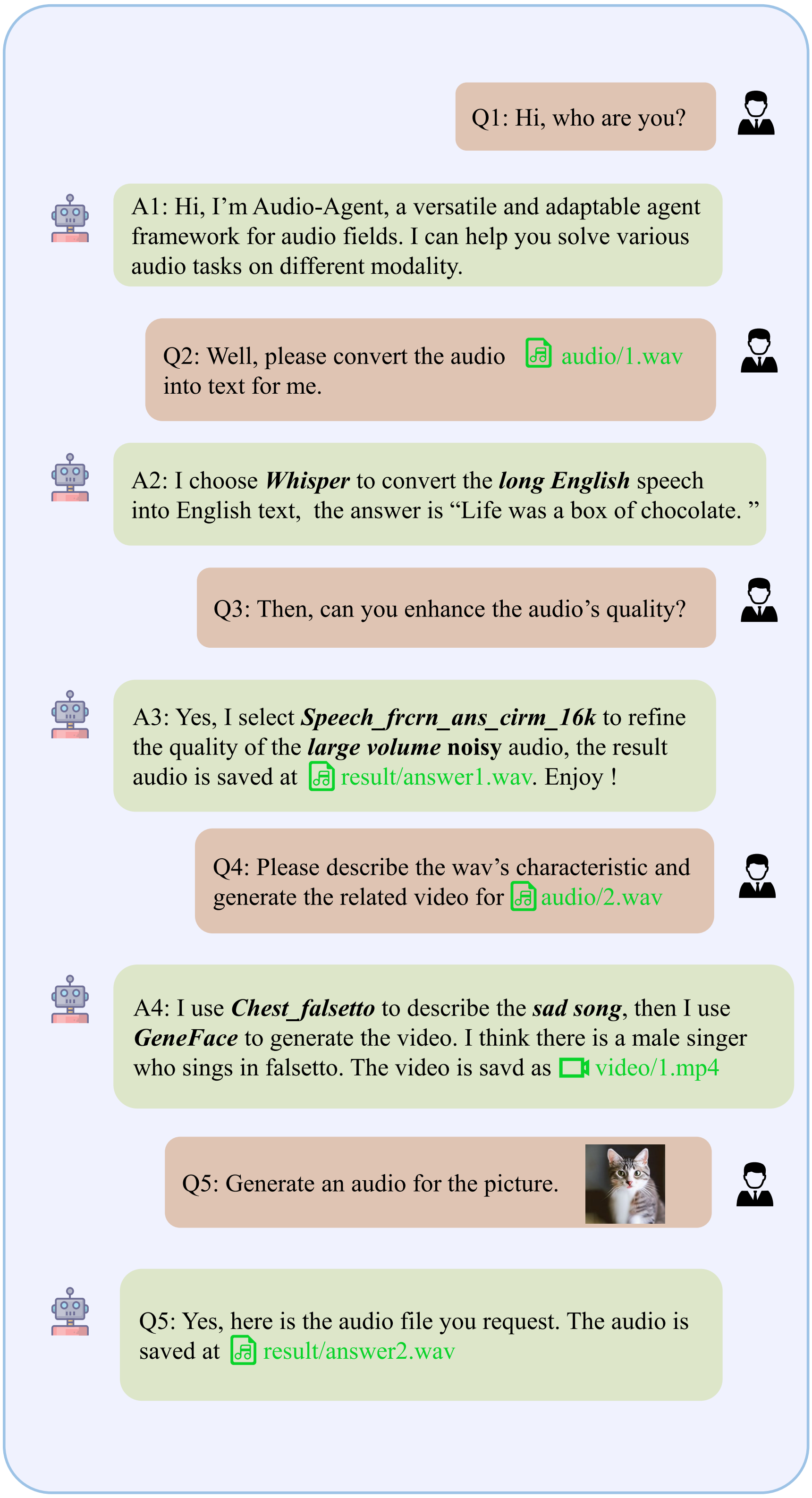


Figure 6: The Whole Process of Multi-turn Dialogue