Boosting Large Language Model for Speech Synthesis: An Empirical Study

Hongkun Hao[1], Long Zhou[2*], Shujie Liu[2], Jinyu Li[2], Shujie Hu[2], Rui Wang[1*], Furu Wei[2]
[1]*Shanghai Jiao Tong University* [2]*Microsoft Corporation*

***Abstract*—Large language models (LLMs) have made significant advancements in natural language processing and are concurrently extending the language ability to other modalities, such as speech and vision. Nevertheless, most of the previous work focuses on prompting LLMs with perception abilities like auditory comprehension, and the effective approach for augmenting LLMs with speech synthesis capabilities remains ambiguous. In this paper, we conduct a comprehensive empirical exploration of boosting LLMs with the ability to generate speech, by combining pre-trained LLM LLaMA/OPT and text-to-speech synthesis model VALL-E. We compare three integration methods between LLMs and speech synthesis models, including directly fine-tuned LLMs, superposed layers of LLMs and VALL-E, and coupled LLMs and VALL-E using LLMs as a powerful text encoder. Experimental results show that, using LoRA method to fine-tune LLMs directly to boost the speech synthesis capability does not work well, and superposed LLMs and VALL-E can improve the quality of generated speech both in speaker similarity and word error rate (WER). Among these three methods, coupled methods leveraging LLMs as the text encoder can achieve the best performance, making it outperform original speech synthesis models with a consistently better speaker similarity and a significant (10.9%) WER reduction.**

***Index Terms*—Speech Synthesis, Large Language Model, VALL-E, LLaMA**

I. Introduction

The emergence of large language models (LLMs), such as ChatGPT [1] and LLaMA [2], has revolutionized most traditional natural language processing (NLP) tasks, like text summarization and dialogue system [3]–[5]. The powerful language generation capabilities of LLMs have prompted exploration into their applications in other modalities, e.g., speech and vision [6]–[15]. For example, GPT-4V [6] enables users to instruct GPT-4 to analyze image inputs they provided. Video-LLaMA [8] empowers LLM with the ability to comprehend both visual and auditory content present in videos. These multimodal LLMs provide the potential to enhance the impact of text-only systems by integrating new interfaces and functionalities, allowing them to handle new tasks and deliver fresh experiences to users.

Regarding the application of LLMs to speech, the majority of earlier research primarily concentrates on aligning speech representation with the LLM input space [9], [16]–[18]. For instance, Speech-LLaMA [16] proposes an effective method to accomplish speech-to-text tasks by leveraging Connectionist Temporal Classification (CTC) [19] model and audio encoder to map the compressed acoustic features to the continuous semantic space of the LLM. Compared to understanding speech, enabling LLMs to generate speech is considerably more challenging, given that speech is a continuous signal significantly deviating from the output space of LLMs. To enable speech generation ability, existing works such as SpeechGPT [20] and AudioPaLM [21] employ the approach of directly fine-tuning a pre-trained LLM, which requires substantial computational resources and time. How to effectively enhance LLMs with the capabilities for speech synthesis remains a relatively unexplored area.

* Long Zhou and Rui Wang are corresponding authors.

This paper is supported in part by the National Natural Science Foundation of China No. 62176153.

To better understand this task, we are going to answer two questions: 1) Can the codec codes be treated by LLMs simply as a kind of language similar to other natural languages? 2) What kind of information can LLMs provide to improve the quality of synthesized speech? To answer these two questions, we propose and compare several integration approaches to enable the LLMs with speech synthesis capability. In this study, we focus on zero-shot text-to-speech (TTS) tasks following the state-of-the-art model VALL-E [22], which mainly uses an auto-regressive (AR) Transformer [23] decoder model to predict the discrete token of speech depending on the corresponding textual tokens. To enhance the speech generation of LLMs, we first discretize the continuous speech into multi-layer discrete codec codes via audio compression model Encodec [24], and expand the vocabulary of LLMs with the vocabulary of codec codes, e.g., 1024 tokens. We design three combination strategies to achieve the first-layer codec code prediction with LLM, like the AR model in VALL-E, as follows:

- **Directly Fine-tuned LLMs**. We directly fine-tune large language models via paired text and codec codes from speech recognition dataset, with full parameters or partial parameters (LoRA [25]), as shown in Figure 1.
- **Superposed LLMs and VALL-E**. Figure 2 illustrates this strategy that we superimpose the two models into one model. In this method, we use the large language model to encode both textual tokens and acoustic tokens, and then we feed them into the codec language model VALL-E.
- **Coupled LLMs and VALL-E**. As shown in Figure 3, we use an additional text-based large language model to encode the text sequence and then input them into the VALL-E AR model. The coupled method differs from the aforementioned superposed approach as it does not utilize LLMs to model codec codes.

After that, we use the non-autoregressive (NAR) model of VALL-E to generate codec codes of the rest quantizers, and utilize the Encodec decoder to recover the waveform of the speech. Models are trained on 44.5K hours Multilingual Librispeech English data and 960 hours LibriSpeech data and evaluated on LibriSpeech dev-clean, dev-other, test-clean, and test-other datasets. Experimental results demonstrate that coupled LLMs and VALL-E can achieve the best performance among baseline and our methods. Additionally, we perform thorough analyses of various facets of our approach, examining the impact of model size, continuous pre-training, and the pre-trained VALL-E. Based on the results, we can draw conclusions as follows:

- Codec codes can not be simply treated as another language since the results of directly fine-tuned LLM are not promising. The reason could be that, the sequence length of codec codes is much longer than the length of corresponding text, and also the information provided by codec codes is much more fine-grained and more diverse than that of text.
- While LLMs with LoRA may not excel at generating codec codes, they can serve as a unified encoder for processing both

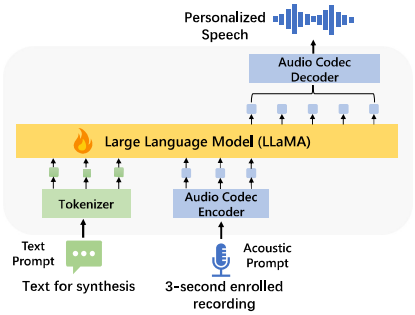

Fig. 1. Method A: Directly fine-tuned LLMs where LLMs are trained for predicting codec codes with an expanded vocabulary.

text and codec codes. The outputs generated by LLMs can provide valuable representation for a codec language model (e.g., VALL-E) to produce more accurate codec codes.

- LLM can be used as a powerful text encoder alone that can model pertinent and extensive content information, which is instrumental for VALL-E to generate speech of superior quality and enhanced robustness. The structure using LLM as a text encoder, coupled with a dedicated decoder module such as VALL-E, achieves the best performance.

II. Methodology

In this section, we first introduce model components in the proposed framework in subsection II-A, including large language model, speech compression model, and codec language model, then present three integration strategies for LLMs and VALL-E in subsection II-B.

A. Model Components

There are three core components in our framework including a large language model (i.e., OPT [26] or LLaMA [2]), a speech compression model (i.e., Encodec [24]), and a codec language model (i.e., VALL-E [22]). The large language model is employed to model textual tokens, with the option to include acoustic tokens as well. Meanwhile, the speech compression model is tasked with transforming continuous speech into discrete codec codes and subsequently reconstructing speech from these codes. Additionally, the codec language model is used to generate codec codes conditioning on the representation of textual tokens.

a) Large Language Model: We conduct experiments with various large language models including OPT [26] models with different sizes including 125M, 350M, and 1.3B, and the LLaMA-7B [2] model. These models will be adapted using either full fine-tuning or parameter-efficient fine-tuning methods such as LoRA [25].

b) Speech Compression Model: To enable the LLM with speech generation ability, we utilize an external speech compression model EnCodec [24] to convert continuous speech into discrete codec codes. EnCodec model is a convolution-based encoder-decoder network with residual vector quantization (RVQ) method. It first tokenizes speech data into L-layer acoustic tokens, and then recovers the speech waveform from all acoustic tokens using EnCodec decoder. In this paper, we adapt EnCodec with 6 kbps bandwidth and L=8 tokens.

c) Codec Language Model: The neural codec language model VALL-E [22] treats text-to-speech synthesis as a language model task and employs acoustic tokens as an intermediate representation of original speech. VALL-E contains two key modules, the auto-regressive (AR) codec language model and the non-autoregressive (NAR) codec language model. The former predicts the acoustic tokens of the first codec code for each frame in an auto-regressive manner, and the latter is used to generate the other 7-layer codes according to the sequence of the first-layer codes in parallel with the

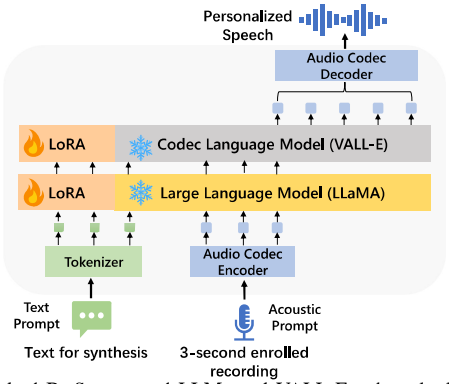

Fig. 2. Method B: Superposed LLMs and VALL-E, where both LLMs and VALL-E are used to model textual tokens and acoustic tokens successively.

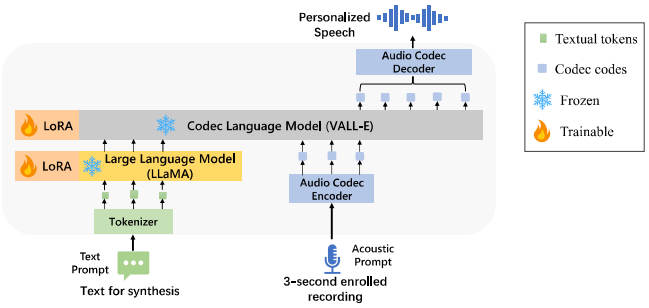

Fig. 3. Method C: Coupled LLMs and VALL-E, where the better text representation provided by LLM is regarded as the textual input of VALL-E.

layer-level iterative generation method. In this work, we follow the VALL-E AR model, to augment LLMs with speech synthesis ability.

B. Integration Strategies

We propose three methods to boost large language models with speech synthesis capability. Figure 1, 2, and 3 illustrate the different methods, including directly fine-tuned LLMs (Method A), superposed LLMs and VALL-E (Method B), and coupled LLMs and VALL-E (Method C). Initially, we propose to directly fine-tune LLMs in Method A to determine if acoustic tokens can be integrated into LLMs by treating them as a novel language. Furthermore, through Method B, we assess the capability of LLMs to encode both acoustic and textual tokens into a unified continuous embedding space, enhancing the performance of VALL-E in text-to-speech tasks. Finally, in Method C, we explore the potential of leveraging only the text encoding proficiency of LLMs to improve TTS outcomes without regarding acoustic tokens as a new language.

a) Method A: Directly Fine-tuned LLMs: In order to verify whether acoustic tokens can be incorporated into LLMs by simply regarding it as a new language, enabling the joint training of both acoustic and textual tokens, the most straightforward approach involves fine-tuning language models directly with TTS training data by either full fine-tuning or parameter-efficient fine-tuning, as shown in Figure 1. Through training on TTS data, we also augment large language models with speech synthesis ability at the same time. In practice, we found that using parameter-efficient fine-tuning methods such as LoRA in this way is less effective and results in relatively poor performance. We speculate that this is because large language models do not have the ability to generate codec codes inherently and it is more difficult for LLMs to generate speech than understand speech signals. Therefore, we directly fully fine-tune LLMs as one kind of approach that endows LLMs with speech synthesis ability.

TABLE I

MAIN EVALUATION RESULTS ON LIBRISPEECH DEV-CLEAN DATASET. FT* MEANS FULL FINE-TUNING, AND OTHER MODELS ADOPT LORA TECHNIQUES. VALL-E IS THE TEXT-TO-SPEECH BASELINE, METHOD A/B/C ARE INTRODUCED IN SECTION II-B, AND INFERENCE STRATEGIES I/II/III ARE LISTED IN SECTION III-D.

Methods	LLMs	Strategy I			Strategy II			Strategy III		
		WER↓	SS↑	SN↑	WER↓	SS↑	SN↑	WER↓	SS↑	SN↑
VALL-E	-	4.39	0.52	3.26	4.27	0.58	3.28	1.31	0.56	3.27
A	OPT-350M	10.28	0.49	3.20	9.74	0.53	3.21	3.97	0.51	3.20
	OPT-350M FT*	4.21	0.53	3.28	4.08	0.60	3.29	1.28	0.58	3.28
	LLaMA-7B	9.61	0.49	3.20	9.19	0.54	3.21	3.63	0.51	3.21
B	OPT-350M	4.12	0.53	3.28	3.94	**0.61**	3.29	1.25	0.57	3.29
	LLaMA-7B	4.05	0.53	3.29	3.82	**0.61**	**3.30**	1.23	0.58	3.29
C	OPT-350M	3.99	**0.54**	**3.30**	3.72	**0.61**	3.29	1.26	**0.59**	**3.30**
	LLaMA-7B	**3.91**	**0.54**	3.29	**3.66**	**0.61**	**3.30**	**1.22**	**0.59**	3.29

b) Method B: Superposed LLMs and VALL-E: Inspired by the observation of Method A introduced above, we aim to further explore the suitability of LLMs for encoding both acoustic tokens and textual tokens into continuous embedding space so that this representation can be used by VALL-E to perform TTS tasks better. As shown in Figure 2, in this approach, we superpose the pre-trained LLMs and VALL-E models to promote the speech generation ability of LLMs. Both textual tokens and acoustic tokens are encoded by LLM, and are sent to the codec language model to predict the first-layer codec code. Besides, a linear projection layer is added between LLM and codec language model to bridge the dimension gap between them.

c) Method C: Coupled LLMs and VALL-E: Given the distinct roles and strengths of LLMs and VALL-E, it would be interesting to investigate the effect of only utilizing the text encoding ability of LLMs, instead of treating acoustic tokens as a new language in previous methods, to promote TTS performance of VALL-E. Therefore, another natural idea is to take full use of the advantages of LLMs and VALL-E, and cascade the pre-trained LLMs and VALL-E into an end-to-end model. LLMs excel at encoding and generating text, while VALL-E specializes in producing speech tokens based on textual tokens. Hence, in this text-to-speech framework, we first use LLMs to encode text and get better text representation, then feed it to VALL-E as text input, as shown in Figure 3. In this method, we also incorporate a linear projection layer between the LLM and the codec language model to reconcile the disparity in dimensions.

III. EXPERIMENTS

A. Experiment Setup

a) Dataset: Pre-trained models are fine-tuned on two ASR datasets, which can also be used to train TTS tasks as VALL-E (X) [22], [27], [28]. Specifically, we use LibriSpeech (LS, 960 hours) [29] and the English part of Multilingual LibriSpeech (MLS) [30].The Multilingual LibriSpeech is a 50K-hour ASR corpus including 8 languages derived from read audiobooks of LibriVox, where English accounts for about 44.5K hours predominately. We evaluate our proposed methods on the LibriSpeech dev-clean, dev-other, test-clean, and test-other datasets. We use the samples that range in duration from 4 to 20 seconds from these datasets.Following [22], we use the first 3 seconds of the ground-truth speech as prompts for each sample synthesis. Each experiment is conducted thrice, with the average score being reported.

b) Data Preprocessing: To unify the training of speech and text modalities, we transform both into discrete tokens. In our approach, ASR data transcriptions are tokenized into subwords (semantic tokens) with the tokenizer from large language models. Meanwhile, speech data are quantized into acoustic tokens using the EnCodec, which operates at a 6 kbps bandwidth and a downsampling ratio of 320, producing 8 acoustic tokens per frame and 75 frames per second of audio. We concatenate the semantic tokens and corresponding acoustic tokens to form a cohesive training sample.

TABLE II

SUBJECTIVE EVALUATION RESULTS ON LIBRISPEECH DEV-CLEAN DATASET.

Methods	LLMs	MOS	SMOS	CMOS
Ground Truth	-	$4.34_{\pm 0.13}$	$4.16_{\pm 0.23}$	0.00
VALL-E	-	$4.09_{\pm 0.19}$	$3.99_{\pm 0.26}$	-0.49
C	OPT-350M	$4.15_{\pm 0.17}$	$4.02_{\pm 0.21}$	-0.37
	LLaMA-7B	**4.21**$_{\pm 0.21}$	**4.08**$_{\pm 0.19}$	**-0.35**

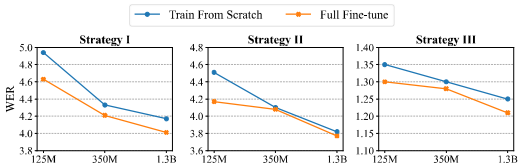

Fig. 4. WER results of using different model sizes in Method A under three inference strategies introduced in Section III-D.

B. Training Details

For Method A, we employ both LoRA and full fine-tuning techniques to train OPT models. However, due to computational resource limitations, we exclusively utilize LoRA for training the LLaMA-7B model. Additionally, we augment the LLMs' vocabulary with acoustic tokens, specifically incorporating 1024 Encodec tokens in our configuration. In Method B, we introduce LoRA parameters to LLM and codec language model respectively. The LLM is initialized with either a pre-trained OPT-350M or LLaMA-7B, while the codec language model is initialized with a pre-trained VALL-E. We also expand the vocabulary of LLM with acoustic tokens like Method A. Besides, the input acoustic and textual embeddings from VALL-E are omitted, as the LLM now provides the representations for both acoustic and textual tokens. Similarly, in Method C we also add LoRA parameters to pre-trained LLM and pre-trained VALL-E respectively, and discard the textual token embedding of VALL-E. We fix the LoRA parameter to $R = 64$ for adjusting self-attention parameters. Consequently, using Method A for LoRA training yields approximately 14M trainable parameters for OPT-350M and 71M for LLaMA-7B. In contrast, Method B incorporates codec code embedding, LoRA, and linear projection, resulting in around 21M trainable parameters for OPT-350M and 82M for LLaMA-7B. Meanwhile, Method C reduces the count of trainable parameters to 20M for OPT-350M and 78M for LLaMA-7B, as it does not utilize codec code embedding for the LLMs. Our models are trained using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$ [31]. All models are trained on TTS tasks for 400K steps on 32 V100 GPUs with a batch size of 100 seconds per GPU. The maximum learning rate is 5×10^{-4} with a warm-up step of 40K. We follow the configuration of VALL-E to train our non-autoregressive language model as introduced in Section II-A. We implement all those models by using the fairseq toolkit [32].

C. Evaluation Metrics

We use the automatic evaluation metrics, including the word error rate (WER), speaker similarity (SS), and speech naturalness (SN) to evaluate the generated speech for simplicity and convenience. The WER score is obtained by an open-source Conformer Transducer model,ranging from 0 to 100. Given generated and prompt speech utterances, the SS is measured by an automatic speaker verification (ASV) WavLM [33] model,ranging from -1 to 1. SN score of generated speech is measured by the open-source NISQA [34].

D. Inference Strategies

After training, we use sampling methods for our models to generate the acoustic tokens of the first layer codec codes. Specifically, we use top-p [35] sampling with $p = 1.0$ and temperature is 1.0. We adopt

TABLE III

EFFECT OF CONTINUAL PRE-TRAINING ON DEV-CLEAN SET WITH METHOD A AND OPT-350M. MLS+LS MEANS THAT THE FINE-TUNING DATA ARE MULTILINGUAL LIBRISPEECH AND LIBRISPEECH, AND LS MEANS LIBRISPEECH ONLY.

Data	Method	Strategy I			Strategy II			Strategy III		
		WER↓	SS↑	SN↑	WER↓	SS↑	SN↑	WER↓	SS↑	SN↑
MLS+LS	Train From Scratch	4.33	0.52	3.26	4.10	0.59	3.28	1.30	0.56	3.27
	Full Fine-tune	4.21	0.53	3.28	4.08	0.60	3.29	1.28	0.58	3.28
	Pre-train+Fine-tune	4.19	0.53	3.28	4.03	0.60	3.29	1.26	0.58	3.28
LS	Train From Scratch	5.71	0.51	3.26	5.11	0.58	3.28	1.97	0.55	3.28
	Full Fine-tune	5.65	0.50	3.26	5.10	0.57	3.27	1.99	0.53	3.28
	Pre-train+Fine-tune	5.47	0.51	3.26	4.99	0.58	3.29	1.91	0.55	3.30

three different strategies to choose sampled sequences following previous work [36].

- Strategy I performs one synthesis inference for one text, and then the sampled acoustic sequence is chosen as the final result.
- Strategy II conducts five inferences for a single text, selecting the utterance that yields the highest speaker similarity score.
- Strategy III also performs five inferences for a given text and selects the utterance that exhibits the lowest word error rate.

E. Main Results

We synthesize the English speech of corresponding text prompted by a 3s English speech utterance on selected samples of dev-clean, dev-other, test-clean, and test-other datasets, where Table I shows the results of dev-clean and others are shown in supplementary material. As summarized in Table I, we replicate the VALL-E baseline using parameters identical to those of [22], while the proposed three methods are validated using both LLaMA-7B and OPT-350M models. We apply the three inference strategies outlined in Section III-D, evaluating their performance using the metrics of word error rate (WER), sentence similarity (SS), and speaker naturalness (SN), as introduced in Section III-C.

According to the experimental results, we can draw three conclusions: (1) Directly fine-tuning LLMs by LoRA performs worse than the VALL-E baseline model. Although full fine-tuning can mitigate the problem and achieve comparable performance with VALL-E, it needs massive computational resources for large models. (2) Method B, when employed with both the OPT-350M or LLaMA-7B models, surpasses the VALL-E baseline in terms of WER, SS, and SN, which demonstrates that augmenting LLM with VALL-E can address the above challenge with LoRA methods, given that LLMs are capable of encoding both acoustic and textual tokens and VALL-E shares a portion of the burden for speech synthesis in LLMs. (3) By fully leveraging the respective strengths of both components, Method C achieves the best performance among the proposed methods, which significantly outperforms VALL-E on word error rate, speaker similarity, and speech naturalness. Compared to the VALL-E, the word error rate of Method C with LLaMA-7B is relatively decreased by 10.9%, 14.3%, and 6.9% under inference Strategy I, II, and III respectively, the speaker similarity is relatively improved by 0.02, 0.03, and 0.03, and the speech naturalness is improved by 0.03, 0.02, and 0.02 respectively.

F. Subjective Evaluation

We conduct subjective evaluations using three types of mean opinion scores (MOS) including MOS for assessing speech quality, Similarity MOS (SMOS) for measuring speaker similarity, and Comparative MOS (CMOS) for evaluating the comparative naturalness of the synthesized speech. As shown in Table II, method C's synthesized speech with LLaMA achieves the best performance across all metrics compared to VALL-E.

IV. ANALYSIS

To facilitate a clearer comprehension of our method, we conduct detailed analyses and ablation studies in this section.

TABLE IV

EFFECT OF PRE-TRAINED VALL-E ON DEV-CLEAN SET WITH METHOD B, WHERE VALL-E IS EITHER RANDOMLY INITIALIZED OR IS LEVERAGED AS A PRE-TRAINED MODEL. FT* MEANS FULL FINE-TUNING, AND MODELS WITH PRE-TRAINED VALL-E ADOPT LORA TECHNIQUES.

LLMs	VALL-E	Strategy I			Strategy II			Strategy III		
		WER↓	SS↑	SN↑	WER↓	SS↑	SN↑	WER↓	SS↑	SN↑
OPT-350M	Randomly (FT*)	4.31	0.52	3.27	4.09	0.59	3.28	1.36	0.56	3.27
	Pre-trained	4.12	0.53	3.28	3.94	0.61	3.29	1.25	0.57	3.29
LLaMA-7B	Randomly (FT*)	4.27	0.52	3.27	4.11	0.59	3.28	1.32	0.56	3.28
	Pre-trained	4.05	0.53	3.29	3.82	0.61	3.30	1.23	0.58	3.29

a) Effect of Model Size: The capacity of a large language model is significantly influenced by its parameter number. Consequently, we explore the impact of varying model sizes within the OPT framework through direct full fine-tuning (referred to as Method A in Table I), examining models with 125M, 350M, and 1.3B parameters. Additionally, we establish baselines by training these models from scratch. The results are depicted in Figure 4. The comparison between the two curves illustrates the effectiveness of using pre-trained LLMs. The largest OPT model with 1.3B parameters achieves the best performance overall compared to 125M and 350M. This finding suggests that increasing the model size could be a viable strategy for enhancing speech synthesis capabilities.

b) Effect of Continual Pre-training: Since unlabeled speech data is more common than paired speech-text data, we also investigate the way of utilizing massive unlabeled speech data to promote speech synthesis performance of LLMs. Inspired by the next token prediction objective of decoder-only LMs, we use EnCodec codes of the LibriLight [37] dataset to continually pre-train LLMs, so that they can adapt to speech modality better. Then we use paired speech-text data to fine-tune continually pre-trained models and compare them with those that have not been continually pre-trained. Table III shows the comparison results of (1) training from scratch, (2) directly full fine-tuning, and (3) continually pre-training and then full fine-tuning, on large (MLS+LS) and small (LS) datasets. The experimental results on Method A with OPT-350M show that the continual pre-training method achieves significant WER reduction than methods of full fine-tuning and training from scratch on the small fine-tuning dataset.

c) Effect of Pre-trained VALL-E: To validate the benefits of employing the pre-trained codec language model VALL-E, we undertake an ablation study focusing on the impact of random initialization versus pre-trained initialization. We fully fine-tune the randomly initialized VALL-E but use LoRA to fine-tune the VALL-E initialized with pre-trained weights. Table IV delineates the performance disparity between models with Method B that begin with random weights and those initialized with pre-trained VALL-E. The results clearly indicate that initializing with pre-trained VALL-E results in fewer trainable parameters and significantly surpasses random initialization across various inference strategies and evaluation criteria.

V. CONCLUSION

In this study, we explore various strategies for incorporating speech synthesis capabilities into large language models (LLMs). Our findings show that simply fine-tuning LLMs with LoRA fails to match the performance of the baseline, indicating the challenge of enhancing LLMs with speech synthesis capabilities. Further investigation demonstrates that LLMs augmented with a pre-trained text-to-speech synthesis model can surpass the performance of the baseline VALL-E model. In particular, by leveraging the respective strengths of LLMs and VALL-E, the coupled LLM and VALL-E method achieves the highest performance among the methods evaluated. Moreover, we conduct comprehensive analyses to better understand the proposed LLMs augmented with speech synthesis ability.

REFERENCES

[1] OpenAI, "Gpt-4 technical report," *ArXiv*, vol. abs/2303.08774, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257532815

[2] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[3] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[4] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, W. L. Tam, Z. Ma, Y. Xue, J. Zhai, W. Chen, Z. Liu, P. Zhang, Y. Dong, and J. Tang, "GLM-130b: An open bilingual pre-trained model," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=-Aw0rrrPUF

[5] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen *et al.*, "Palm 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[6] OpenAI, "Gpt-4v(ision) system card," 2023. [Online]. Available: https://cdn.openai.com/papers/GPTV_System_Card.pdf

[7] S. Huang, L. Dong, W. Wang, Y. Hao, S. Singhal, S. Ma, T. Lv, L. Cui, O. K. Mohammed, Q. Liu *et al.*, "Language is not all you need: Aligning perception with language models," *arXiv preprint arXiv:2302.14045*, 2023.

[8] H. Zhang, X. Li, and L. Bing, "Video-llama: An instruction-tuned audio-visual language model for video understanding," *arXiv preprint arXiv:2306.02858*, 2023.

[9] C. Tang, W. Yu, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, and C. Zhang, "Salmonn: Towards generic hearing abilities for large language models," *arXiv preprint arXiv:2310.13289*, 2023.

[10] 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," *arXiv preprint arXiv:2305.06500*, 2023.

[11] S. K. Muhammad Maaz, Hanoona Rasheed and F. Khan, "Video-chatgpt: Towards detailed video understanding via large vision and language models," *arXiv preprint arXiv:2306.05424*, 2023.

[12] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[13] S. Moon, A. Madotto, Z. Lin, T. Nagarajan, M. Smith, S. Jain, C.-F. Yeh, P. Murugesan, P. Heidari, Y. Liu *et al.*, "Anymal: An efficient and scalable any-modality augmented language model," *arXiv preprint arXiv:2309.16058*, 2023.

[14] Y. Chu, J. Xu, X. Zhou, Q. Yang, S. Zhang, Z. Yan, C. Zhou, and J. Zhou, "Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models," *arXiv preprint arXiv:2311.07919*, 2023.

[15] S. Hu, L. Zhou, S. Liu, S. Chen, L. Meng, H. Hao, J. Pan, X. Liu, J. Li, S. Sivasankaran, L. Liu, and F. Wei, "Wavllm: Towards robust and adaptive speech large language model," *arXiv preprint arXiv:2404.00656*, 2024.

[16] J. Wu, Y. Gaur, Z. Chen, L. Zhou, Y. Zhu, T. Wang, J. Li, S. Liu, B. Ren, L. Liu *et al.*, "On decoder-only architecture for speech-to-text and large language model integration," *arXiv preprint arXiv:2307.03917*, 2023.

[17] Y. Fathullah, C. Wu, E. Lakomkin, J. Jia, Y. Shangguan, K. Li, J. Guo, W. Xiong, J. Mahadeokar, O. Kalinli *et al.*, "Prompting large language models with speech recognition abilities," *arXiv preprint arXiv:2307.11795*, 2023.

[18] Y. Shu, S. Dong, G. Chen, W. Huang, R. Zhang, D. Shi, Q. Xiang, and Y. Shi, "Llasm: Large language and speech model," *arXiv preprint arXiv:2308.15930*, 2023.

[19] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

[20] D. Zhang, S. Li, X. Zhang, J. Zhan, P. Wang, Y. Zhou, and X. Qiu, "Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities," *arXiv preprint arXiv:2305.11000*, 2023.

[21] P. K. Rubenstein, C. Asawaroengchai, D. D. Nguyen, A. Bapna, Z. Borsos, F. d. C. Quitry, P. Chen, D. E. Badawy, W. Han, E. Kharitonov *et al.*, "Audiopalm: A large language model that can speak and listen," *arXiv preprint arXiv:2306.12925*, 2023.

[22] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023. [Online]. Available: https://arxiv.org/abs/2301.02111

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[24] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[25] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[26] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.

[27] Z. Zhang, L. Zhou, C. Wang, S. Chen, Y. Wu, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Speak foreign languages with your own voice: Cross-lingual neural codec language modeling," *arXiv preprint arXiv:2303.03926*, 2023. [Online]. Available: https://arxiv.org/abs/2303.03926

[28] S. Chen, S. Liu, L. Zhou, Y. Liu, X. Tan, J. Li, S. Zhao, Y. Qian, and F. Wei, "Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers," *arXiv preprint arXiv:2406.05370*, 2024.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[30] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "MLS: A Large-Scale Multilingual Dataset for Speech Research," in *Proc. Interspeech 2020*, 2020, pp. 2757–2761.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[32] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 48–53. [Online]. Available: https://aclanthology.org/N19-4009

[33] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[34] G. Mittag and S. Möller, "Deep Learning Based Assessment of Synthetic Speech Naturalness," in *Proc. Interspeech 2020*, 2020, pp. 1748–1752.

[35] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=rygGQyrFvH

[36] T. Wang, L. Zhou, Z. Zhang, Y. Wu, S. Liu, Y. Gaur, Z. Chen, J. Li, and F. Wei, "Viola: Unified codec language models for speech recognition, synthesis, and translation," *arXiv preprint arXiv:2305.16107*, 2023.

[37] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fügen, T. Likhomanenko, G. Synnaeve, A. Joulin, M. I. Abdelrahman, and E. Dupoux, "LIBRI-LIGHT: a benchmark for asr with limited or no supervision," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*. Barcelona / Virtual, Spain: IEEE, May 2020, pp. 7669–7673. [Online]. Available: https://hal.science/hal-02959460