UniAudio: Towards Universal Audio Generation with Large Language Models

Dongchao Yang [*1] **Jinchuan Tian** [*2] **Xu Tan** [3] **Rongjie Huang** [4] **Songxiang Liu** [5] **Haohan Guo** [1]
Xuankai Chang [2] **Jiatong Shi** [2] **Sheng Zhao** [3] **Jiang Bian** [3] **Zhou Zhao** [4] **Xixin Wu** [1] **Helen Meng** [1]

Abstract

Audio generation is a major branch of generative AI research. Compared with prior works in this area that are commonly task-specific with heavy domain knowledge, this paper advocates building universal audio generation models that can handle various tasks in a unified manner. As recent research on large language models (LLMs) has demonstrated their strong ability to handle multiple tasks, this work presents UniAudio, an LLM-based audio generation model that supports a wide range of audio generation tasks. Based on various input conditions, such as phoneme, text description, or audio itself, UniAudio can generate speech, sound, music, and singing voice. The proposed UniAudio is built with 100k hours of multi-source open-available audio data and is scaled to 1B parameters. The audio tokenization method and language model architecture are also specifically designed for both performance and efficiency. Experimentally, UniAuido supports 11 audio generation tasks and achieves competitive results on all tasks consistently. We also show that UniAudio can support new tasks seamlessly via simple fine-tuning [1].

1. Introduction

Recently, the popularity of generative AI has induced increasingly emergent and varying needs in audio generation, i.e., generating human speech, music, and other audio. Prior works on audio generation tasks are commonly task-specific: their designs heavily leverage domain knowledge and their usage is restricted to fixed setups (Tan et al., 2021b; Luo & Mesgarani, 2019; Zmolikova et al., 2023; Huang et al., 2021b; Cho et al., 2021). Instead of taking care of each task independently, this work advocates achieving universal audio generation, which intends to accomplish multiple audio generation tasks with only one unified model. Specifically, building a universal model in audio generation can not only leverage the massive data collected from various sources and tasks but also explore the shared prior knowledge among various data modalities and domains. Additionally, compared with building task-specific models, the proposed routine can save human effort considerably. Thus, building a universal audio generation model is a solution of both high performance and cost-effectiveness towards the increasing needs of generating diverse types of audio.

The superiority of Large Language Models (LLMs) in text-generative tasks inspires a series of LLM-based models in audio generation (Wang et al., 2023a; Kharitonov et al., 2023; Huang et al., 2023d; Agostinelli et al., 2023; Borsos et al., 2023). Among these works, LLM's capability in independent tasks has been extensively studied in tasks like speech generation (Wang et al., 2023a; Kharitonov et al., 2023; Huang et al., 2023d) and music generation (Agostinelli et al., 2023; Copet et al., 2023), and achieves competitive performance. However, LLM's ability to process multiple tasks with a unified model is less exploited in audio generation research: most existing LLM-based works are still designed for a single or a few tasks (Wang et al., 2023a; Kharitonov et al., 2023; Wang et al., 2023c), such as text-to-speech, speech enhancement. We argue that achieving universality in audio generation through the LLM paradigm is promising but has not yet been comprehensively studied before this work.

Toward universal audio generation, this work presents UniAudio, which adopts LLM techniques and can generate multiple types of audio (speech, sounds, music, and singing) conditioned on various input modalities, such as phoneme sequences, text descriptions, and audio itself. By various tokenization methods, UniAudio first transforms audio and conditions from other modalities as discrete tokens. The

[*]Equal contribution [1]The Chinese University of Hong Kong, Hong Kong SAR, China [2]Language Technologies Institute, Carnegie Mellon University, USA [3]Microsoft Research Asia , China [4]Zhejiang University, China [5]Independent Researcher, China. Correspondence to: Dongchao Yang <dcyang@se.cuhk.edu.hk>, Helen Meng <hhmeng@se.cuhk.edu.hk>, Xu Tan <xutan@microsoft.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1]Part of the work was done by Dongchao Yang as an intern at Microsoft Research Asia. Demo and code are released http://dongchaoyang.top/UniAudio_demo/

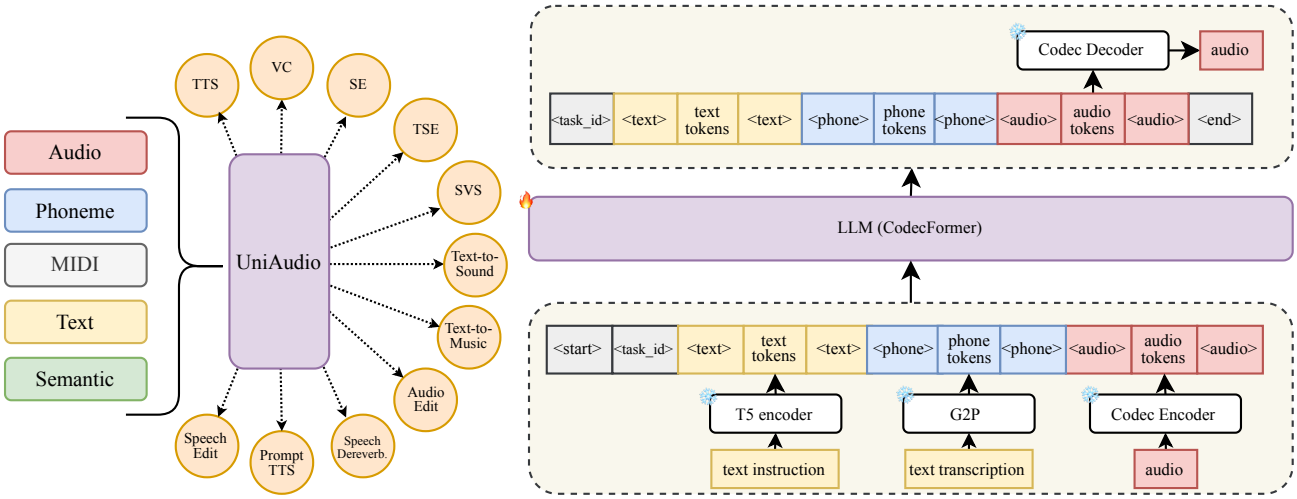


Figure 1. UniAudio is a versatile audio generation model that conditions on multiple types of inputs and performs a variety of audio generation tasks. All modalities are represented as discrete tokens. Based on the given conditions, the audio tokens are predicted by an LLM (CodecFormer, see Section 2.3) and then recovered as audio. The sequence layout of the prompt TTS task is shown as an example. Other tasks are also processed as sequences (see Table 1).

LLM then generates audio token sequences based on the condition sequences. The latter can vary along with the different task formulations. Finally, waveforms are generated from detokenizing the generated audio token sequences. This work follows (Kharitonov et al., 2023; Wang et al., 2023a) to adopt a neural codec model (Zeghidour et al., 2021) for audio tokenization. To represent multiple types of audio uniformly and to preserve the efficiency of our LLM-based model, a novel codec model is built and adopted. Meanwhile, the neural codec model adopts the residual vector quantization (RVQ) technique, which results in an overly long sequence issue in audio sequences. With a mild assumption inspired by RVQ, we propose an improved LLM architecture that significantly alleviates the negative impact raised by audio sequence length.

UniAudio is claimed as a multi-functional model and can seamlessly support new audio generation tasks. To demonstrate this, the building process of UniAudio is intentionally decoupled into two stages. Firstly, the proposed UniAudio is trained on multiple audio generation tasks jointly, which allows the model to obtain sufficient prior knowledge not only of the intrinsic properties of audio but also of the interrelationship between audio and other input modalities. Secondly, through fine-tuning, the trained model can support more new audio generation tasks. Thus, UniAudio has the potential to become a foundation model for universal audio generation: it can continuously support emergent needs in audio generation.

Experimentally, UniAudio supports 11 audio generation tasks in total. The building process of UniAudio is scaled up to 100k hours of audio and 1B parameters. Among the 11 tasks, UniAudio consistently obtains competitive performance in both objective and subjective evaluations. We further conduct a comprehensive ablation study to verify that building this unified audio generation model by joint training is mutually beneficial to each task involved. Demo and code are released, in the hope that UniAudio can support emergent audio generation in future research.

2. UniAudio

This section introduces the technical details of the proposed UniAudio. Section 2.1 explains how audio and other modalities are tokenized. Then, all considered audio generation tasks are uniformly formulated in section 2.2. Finally, the CodecFormer architecture is presented in section 2.3 to handle the long audio token sequences.

2.1. Tokenization

In this work, audio and all other input modalities are tokenized before being processed. These processes for each modality are completed by independent modules. All of these modules are fixed in the optimization of UniAudio.

2.1.1. Audio

All audio is tokenized as discrete sequences using neural audio codec models (short as *codec models*) (Défossez et al., 2022; Yang et al., 2023b; Zeghidour et al., 2021) before being modeled by the LLM. As demonstrated in Fig. 2, the codec models are a series of neural networks that follow the encoder-decoder paradigm and quantize the intermediate encoder output into discrete tokens. Given the summed embedding of these discrete tokens, the codec decoder can

Table 1. Sequence formats of all tasks supported by UniAudio. Text color represents modality. black: audio; green: phoneme; blue: MIDI; purple: text; brown: semantic token. ♣ means tasks that generate audio with deterministic length. ♢: means tasks that are only included in the fine-tuning stage. The speaker prompt is a 3-second speech and is used to represent the speaker-related information.

Task	Conditions	Audio Target
Text-to-Speech (TTS) (Wang et al., 2023a)	phoneme, speaker prompt	speech
Voice Conversion (VC) ♣ (Wang et al., 2023f)	semantic token, speaker prompt	speech
Speech Enhancement (SE) ♣ (Wang et al., 2023b)	noisy speech	speech
Target Speech Extraction (TSE) ♣ (Wang et al., 2018)	mixed speech, speaker prompt	speech
Singing Voice Synthesis (SVS) (Liu et al., 2022)	phoneme (with duration), speaker prompt, MIDI	singing
Text-to-Sound (Sound) (Yang et al., 2023c)	textual description	sound
Text-to-Music (Music) (Agostinelli et al., 2023)	textual description	music
Audio Edit (A-Edit) ♣♢ (Wang et al., 2023e)	textual description, original sounds	sounds
Speech dereverberation (SD) ♣♢ (Wu et al., 2016)	reverberant speech	speech
Prompt TTS (P-TTS)♢ (Guo et al., 2023)	phoneme, textual instruction	speech
Speech Edit (S-Edit) ♢ (Tae et al., 2021)	phoneme (with duration), original speech	speech

faithfully recover the audio waveform. Thus, these intermediate discrete tokens of codec models have been extensively used as the predicting targets in LLM-based audio generation, such as Borsos et al. (2023) and Kharitonov et al. (2023). A key feature of codec models is the adoption of residual vector quantization (RVQ) (Zeghidour et al., 2021) in the hidden space, which uses multiple quantization layers (3 in Fig. 2) to progressively reduce the quantization error and achieve better reconstruction performance.

The adoption of RVQ usually raises a trade-off between quality and efficiency: although the modeling quality can increase with more RVQ layers due to less quantization error, the length of the audio sequences grows proportionally with the number of RVQ layers, which makes the audio sequence too long to be modeled by LLM efficiently (Borsos et al., 2023). This work follows (Kharitonov et al., 2023) and demonstrates that even 3 RVQ layers are sufficient to achieve impressive modeling quality. Additionally, using only 3 RVQ layers boosts the audio modeling efficiency due to the reduced audio sequence length. We also set our codec model to 50 frame-per-second (FPS) for efficiency.

UniAudio is designed to generate audio of multiple types (speech, sounds, music, or singing) with a single model. It thus requires a codec model that: (1) can represent all these types of audio in a shared latent space; (2) the encode-quantization-decode process is near-lossless; (3) using few RVQ layers to facilitate efficient training and inference. Following the pioneer works (Défossez et al., 2022; Kumar et al., 2023), we build our codec model with carefully designed model structure, training strategy, and data. Table. 2 presents a comparison between our codec model with the prior works to show that our codec model achieves superior performance on all 4 audio types, especially when only 3 RVQ layers are adopted. The details of the building process of our codec model, along with the detailed performance compared with prior works, are presented in Appendix D.

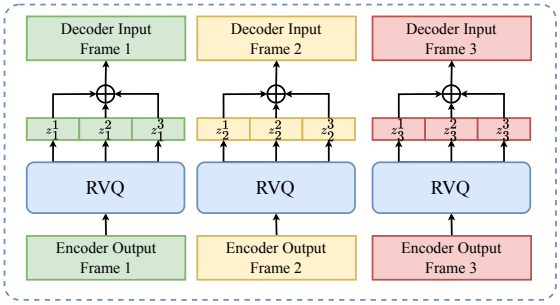


Figure 2. An overview of residual vector quantization (RVQ) adopted in neural codec models. Discrete tokens $\mathbf{z}_t = [z_1, ..., z_N]$ stands for the t-th frame.

2.1.2. Other Modalities

Besides audio, other modalities considered in UniAudio also need to be represented as sequences. In addition, most of these sequences are transformed into discrete ones through tokenization. The tokenization of these input modalities, along with their key features, are briefly summarized below.

Phoneme: Phonemes are the basic units of speech pronunciation in linguistics. Phoneme sequences have multiple sources: (1) when only text is available, phoneme sequence without duration information can be obtained by text-to-phoneme mapping using a pronunciation dictionary; (2) when only speech is available, phoneme sequence with duration information is obtained by beam search of the DNN-HMM system (Hinton et al., 2012); (3) when both text and speech are available, phoneme sequence with duration information is obtained by forced-align operation of the DNN-HMM system.

MIDI: MIDI (Zhang et al., 2022) is widely used for singing voice synthesis tasks. F0 and duration information are included in the MIDI. We use the duration information to flatten the F0 sequence so that the frame-level F0 sequence is obtained.

Table 2. Performance comparison between open-sourced universal audio codec models and our universal neural codec. FPS: frame per second; TPS: token per second. Perceptual evaluation of speech quality (PESQ↑); Short Term Objective Intelligibility (STOI↑). We conduct MOS evaluation for each group. Specifically, we hire 10 professional listeners to conduct the subjective study. Then we ask them to give a score (1-5) to assess the reconstruction performance

Type					Speech (VCTK) (Veaux et al., 2017)		Sound (cloth) (Drossos et al., 2020)		Music (musiccaps) (Agostinelli et al., 2023)		Sing (m4sing) (Zhang et al., 2022)		Average -		
Model	Size (M)	N	FPS	TPS	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI	MOS
Encodec	14.1	3	75	225	2.18	0.79	2.03	0.48	1.86	0.57	1.95	0.76	2.05	0.65	2.12
Encodec	14.1	4	75	600	2.76	0.83	2.36	0.55	2.01	0.59	2.44	0.82	2.54	0.71	2.83
Encodec	14.1	8	75	600	**3.01**	**0.83**	**2.56**	**0.60**	**2.21**	**0.62**	**2.92**	**0.87**	**2.67**	**0.73**	**4.01**
DAC	70.7	3	50	150	1.76	0.78	1.97	0.48	1.48	0.52	1.36	0.68	1.64	0.615	2.08
DAC	70.7	4	50	200	2.05	0.83	2.14	0.51	1.63	0.59	1.53	0.74	1.84	0.67	2.45
DAC	70.7	8	50	400	**3.35**	**0.91**	**2.79**	**0.61**	**2.45**	**0.74**	**2.98**	**0.88**	**2.96**	**0.78**	**4.33**
Ours	15.0	3	50	150	2.96	0.85	2.42	0.49	1.99	0.57	3.13	0.85	2.62	0.69	3.75
Ours	15.0	4	50	200	3.11	0.86	2.5	0.51	2.08	0.59	3.27	0.86	2.73	0.71	3.92
Ours	15.0	8	50	400	**3.36**	**0.88**	**2.67**	**0.54**	**2.31**	**0.65**	**3.49**	**0.89**	**2.95**	**0.74**	**4.41**

Text: Text acts as an effective carrier of human instructions in audio generation tasks (Yang et al., 2023a; Copet et al., 2023; Guo et al., 2023). In this work, these instructions are represented as **continuous** embedding sequences derived from pre-trained text T5 model (Raffel et al., 2020), as these embedding sequences contain rich textual semantics.

Semantic Token: The semantic tokens are derived from the continuous embeddings output by audio self-supervised learning (SSL) models. These continuous representations are highly informative and can be adopted in both speech understanding (Rubenstein et al., 2023) and generative tasks (Borsos et al., 2023). Following Huang et al. (2023d), these continuous representations are turned discrete by performing K-means clustering (Hsu et al., 2021) over these continuous representations.

2.2. Unified Task Formulation

UniAudio does each audio generation task based on the given conditions and task identifier. All tasks can be uniformly formulated as sequential modeling tasks that can be processed by LLM: both the target audio and the conditions are firstly transformed as sub-sequences and then spliced as [*conditions*, *target*] before being processed by the LLM. To avoid ambiguity, some special tokens are inserted to indicate (1) the start and end of the whole sequence; (2) the start and end of each sub-sequence of a certain modality; and (3) the task identifier. For example, the sequence layout of the prompt TTS (Guo et al., 2023) task is shown in Fig.1, in which target speech is generated from the textual description and phoneme sequence. The detailed sequential format of each task is in Table 1.

2.3. CodecFormer

Denote the number of audio frames and RVQ layers as T and N respectively. Previous work on LLM-based audio generation (Borsos et al., 2023; Kharitonov et al., 2023) advocates modeling the audio tokens sequences in the flattened format to emphasize generation quality. However, these sequences are processed in the length of $T \times N$, which is highly challenging considering the quadratic space complexity of Transformer (Vaswani et al., 2017) w.r.t. the sequence length. Inspired by Yu et al. (2023), the CodecFormer architecture is specifically designed for discrete audio sequences modeling, which is a hierarchical model that processes the inter- and intra-frame correlation of audio separately. The overview of the proposed architecture is in Figure 3.

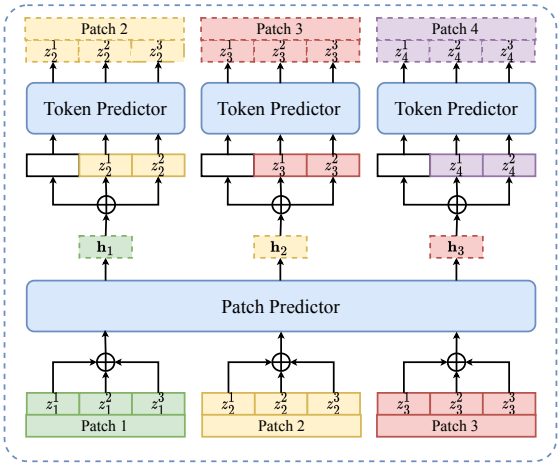


Figure 3. CodecFormer architecture. z_t^k denotes the k-th token in t-th patch. Dashed boxes mean predicted values.

The CodecFormer consists of a patch predictor and a token predictor, both of which are Transformer decoder-only models with full causality. Define every N consecutive tokens as a *patch*. Then, for audio token sequences modeling, each patch exactly represents one audio frame. First, to align with RVQ (Défossez et al., 2022), each patch is represented by the sum of all token embeddings within that patch, and then digested by the patch predictor. Second, for each t-th frame $\mathbf{z}_t$, the patch predictor outputs the continuous vector $\mathbf{h}_t$ that encodes patch $\mathbf{z}_t$ and all its previous content. Third, based on $\mathbf{h}_t$, the tokens within patch $\mathbf{z}_{t+1}$ is predicted by

token predictor auto-regressively. Note the $\mathbf{h}_t$ involves in this process by simply adding it to the input embeddings of each token [2].

The proposed CodecFormer architecture is also compatible with both discrete and continuous sequences besides audio. For all discrete tokens except audio (phoneme, semantic, MIDI, and special tokens), each token has independent semantics and thus should account for one whole patch. So these discrete tokens repeat for N times to fill each patch. The continuous text embeddings are also repeated for N times for the same purpose [3].

The design of the CodecFormer is based on the mild assumption that the context-dependent vector $\mathbf{h}_t$ is sufficient to predict the token within the patch $\mathbf{z}_{t+1}$. As shown in Fig. 2, this assumption is reasonable for audio tokens from codec models, as the original RVQ process for each frame depends on the encoder output of that frame only. The design of the proposed CodecFormer can effectively reduce computational complexity, as the equivalent sequence length for the patch predictor is reduced from $T \times N$ to T, which is no longer proportional to N. The token predictor empirically has fewer parameters than the patch predictor as it only works on short sequences of fixed length N.

3. Experiments

This section first introduces the experimental setup in Section 3.1. The results for the training stage and the fine-tuning stage are presented in Section 3.2 and 3.3 respectively. Ablation studies are presented in Section 3.4.

3.1. Experimental Setup

Data and Model: UniAudio is built on 12 datasets, all of which are publicly available. The overall audio volume is about 100K hours. Detailed data statistics and their adoption for each task are in Appendix A.1. Discrete tokens from all modalities, along with the special tokens, form a joint vocabulary of size 4212. Both patch predictor and token predictor are vanilla decoder-only Transformers (Vaswani et al., 2017). The overall parameter budget is roughly 1B. Detailed model configuration is in Appendix A.2.

Training and Inference: To verify that UniAudio can seamlessly support new audio generation tasks by fine-tuning, our model is primarily trained on 7 tasks while 4 additional tasks are introduced in the fine-tuning stage [4]. Both the training and fine-tuning are completed with 16 AMD MI200-64G GPUs. The detailed configuration of optimization is in Appendix A.3. Cross entropy loss is uniformly applied to the whole sequence, including both condition and target. To retain the performance of previous tasks during fine-tuning, following Conneau et al. (2020), the training data are re-balanced with $\alpha = 0.05$ across tasks. Top-k sampling is adopted consistently for inference, in which k and the temperature are set to 30 and 0.8, respectively. As the patch predictor does not directly predict tokens, the sampling process only happens in the token predictor inference.

Evaluation: Most tasks are evaluated using both objective and subjective metrics. Generally, for objective evaluation, Word Error Rate (WER) is used to evaluate the intelligibility of generated speech; Similarity Score (SIM) is for similarity in terms of speaker identity (Wang et al., 2023a); Perceptual Evaluation of Speech Quality (PESQ), VISQOL, DNSMOS (Reddy et al., 2021) and Mel Cepstral Distortion (MCD) are signal-level quality metrics derived from human auditory research; Following (Copet et al., 2023), Fréchet Audio Distance (FAD), Kullback-Leiber (KL) Divergence, and Fréchet Distance (FD) are for audio fidelity and audio similarity; For subjective evaluation, Mean Opinion Scores (MOS) and Similarity Mean Opinion Scores (SMOS) are adopted to provide human-centric judgment for speech and sing related tasks. For text-to-sound and text-to-music tasks, we use overall quality (OVL), and relevance to the text input (REL) (Copet et al., 2023). Note all subjective results are obtained from the third-party evaluation (Amazon Mechanical Turk) for a fair comparison. Appendix E shows details of the evaluation.

3.2. Training Stage Results

Table 3 presents the overall evaluation results of the proposed UniAudio model over 7 audio generation tasks during the training stage. In this table, we compare UniAudio with one of the most advanced prior works in each task. Detailed comparison with other competitors, including not only the LLM-based methods but also the diffusion model-based methods as well as other conventional audio generation methods, is presented in Appendix B.

As suggested in Table 3, UniAudio is a versatile system that can handle all 7 audio generation tasks together and achieve competitive performance. Per subjective evaluation, UniAudio surpasses the baselines in 3 out of 6 tasks (TTS, VC, Text-to-Sound); per objective evaluation, it achieves better results on 5 out of the 7 tasks except SVS and Music. We also find that UniAudio underperforms on several metrics. UniAudio's subjective performance for SE and TSE is less

[2]We append a *start-of-setence* and a *end-of-sentence* patches at the start and end of each sequence respectively.

[3]The corresponding predicting targets of continuous representations are consecutive special tokens <*continuous_token*>.

[4]The task split is shown in Table 3 and 4. We intentionally select the tasks with more massive data for the first training stage and then fine-tune the model on tasks with less data available, which is aligned with the realistic situation that the newly emerged tasks are generally of low resources.

Table 3. Performance evaluation for UniAudio and selected prior works in the training stage

Task	Model	Objective Evaluation		Subjective Evaluation	
		Metrics	Results	Metrics	Results
Text-to-Speech	Shen et al. (2023)	SIM(↑) / WER(↓)	0.62 / 2.3	MOS(↑)	**3.83±0.10** / 3.11±0.10
	UniAudio		**0.71 / 2.0**	/ SMOS(↑)	3.81±0.07 / **3.56±0.10**
Voice Conversion	Wang et al. (2023f)	SIM(↑) / WER(↓)	0.82 / 4.9	MOS(↑)	3.41±0.08 / 3.17±0.09
	UniAudio		**0.87 / 4.8**	/ SMOS(↑)	**3.54±0.07 / 3.56±0.07**
Speech Enhancement	Chen et al. (2022)	PESQ(↑) / VISQOL(↑) / DNSMOS(↑)	**3.41 / 2.99** / 3.34	MOS(↑)	3.56±0.08
	UniAudio		2.63 / 2.44 / **3.66**		**3.68±0.07**
Target Speaker Extraction	Žmolíková et al. (2019)	PESQ(↑) / VISQOL(↑) / DNSMOS(↑)	**2.89 / 2.25** / 3.18	MOS(↑)	3.43±0.09
	UniAudio		1.88 / 1.68 / **3.96**		**3.72±0.06**
Singing Voice Synthesis	Liu et al. (2022)	-	-	MOS(↑)	3.94±0.02 / **4.05±0.06**
	UniAudio			/ SMOS(↑)	**4.08±0.04** / 4.04±0.05
Text-to-Sound	Ghosal et al. (2023)	FAD (↓) / KL (↓)	3.61 / 2.6	OVL (↑)	**66.2±1.7 / 68.6±1.5**
	UniAudio		**3.12 / 2.6**	/ REL (↑)	61.9±1.9 / 66.1±1.5
Text-to-Music	Copet et al. (2023)	FAD (↓) / KL (↓)	4.52 / **1.4**	OVL (↑)	**73.3±1.5 / 71.3±1.7**
	UniAudio		**3.65** / 1.9	/ REL (↑)	67.9±1.7 / 70.0±1.5

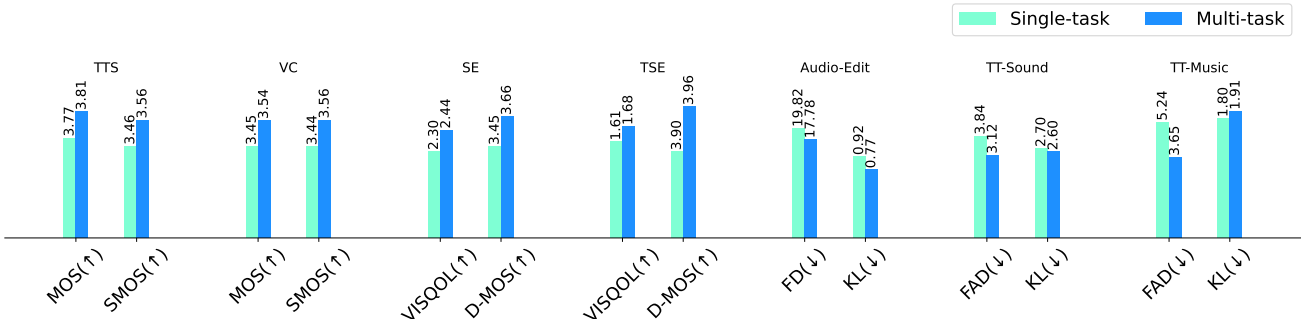


Figure 4. The ablation study of the effectiveness of multi-task training. The up-arrow denotes higher metrics are better, instead down-arrow denotes lower metrics are better.

competitive compared with its competitors, which is also observed in previous literature (Erdogan et al., 2023) that the signal-level evaluation metrics may not be suitable for LLM-based generative methods. UniAudio cannot surpass the selected competitor (Copet et al., 2023) in the Text-to-Music task. We note that (Copet et al., 2023) is built with more private labeled data than our UniAudio.

3.3. Support New Tasks

As UniAudio is designed to continuously support new audio generation tasks, this section reports UniAudio's performance on unseen tasks. The model is obtained by fine-tuning over 4 new tasks jointly and the results are presented in Table 4. Similar to section 3.2, for each task, we compare UniAudio's performance with one selected prior work and report the detailed results in Appendix B.

As shown in Table 4, the fine-tuned UniAudio model surpasses its baselines in audio edit and speech dereverberation and is approaching the ground-truth quality in the prompt TTS task. For speech editing, UniAudio shows considerable improvement compared to generating the whole sentence [5]. In Appendix C.1, we additionally validate that fine-tuning over the 4 new audio generation tasks does not affect UniAudio's performance on the original 7 tasks.

3.4. Ablation Study

3.4.1. Benefit of Building a Unified Model

This work claims that building a unified model for all 11 audio generation tasks is promising and beneficial. To validate this claim, we conduct ablation study to compare the performance of the jointly trained UniAudio and the corresponding models that are trained only on one task each. In Figure 4, we present partial results of 7 tasks. Details of this comparison on all 11 tasks are in Appendix C.2. We observe that the jointly trained model outperforms the task-specific models consistently on all tasks, regardless they are included in the training stage or the fine-tuning stage, which primarily validates the benefit of building the unified model. The benefit of training a unified audio generation model jointly is further validated in two factors: data scale and cross-domain learning.

[5]Following Tan et al. (2021a), generating the whole sentence with a TTS system is generally adopted as the baseline.

Table 4. Performance evaluation for UniAudio and selected prior works in the fine-tuning stage.

Task	Model	Evaluation Metrics	Evaluation Results
Audio Edit	AUDIT	FD (↓) / KL (↓)	20.78 / 0.86
	UniAudio		**17.78 / 0.77**
Speech Derev.	SGMSE+	PESQ(↑) / DNSMOS(↑)	**2.87** / 3.42
	UniAudio		2.13 / **3.51**
P-TTS	GroundTruth	MOS(↑) / SMOS(↑)	**3.77±0.07 / 3.85±0.08**
	UniAudio		3.61±0.09 / 3.71±0.09
Speech Edit	TTS regeneration	MCD(↓) / MOS(↑)	6.98 / 3.69±0.08
	UniAudio		**5.12 / 3.82±0.06**

Data Scale: We note that UniAudio is trained on a mixture of 12 datasets. By contrast, only a few out of these 12 datasets are applicable when training the task-specific models. Thus, compared with the models that are built for one task, the jointly trained UniAudio is built with datasets collected from multiple tasks, for which it enjoys massive data volume and then achieves non-trivial improvement over those task-specific models. Fig. 5 experimentally demonstrates the importance of data volume: when reducing the data volume by randomly sampling the training data to 1/2 and 1/4 of its original size, the UniAudio model encounters considerable performance degradation.

Cross-Domain Learning: UniAudio is trained on various domains of audio data, such as speech, sound, and music. We conjecture that, through joint training, the out-of-domain data can also contribute to the in-domain learning task by exploring the shared features of all domain (OpenAI, 2023; Kondratyuk et al., 2023). Here we present a case study on how the audio data from other domains (sound and speech) can help improve the generation performance on the music generation task, i.e., text-to-music.

With keeping all other setups the same, we train UniAudio on 4 distinctive data combinations: (1) data for the text-to-music task only ; (2) data for text-to-{music, sound} tasks; (3) data for text-to-{music, speech} tasks; (4) data for text-to-{music, sound, speech} tasks. We then evaluate their performance on the text-to-music task and report the results in Table 5. As suggested in that Table, compared with the model trained only on text-to-music data, consistent performance improvement on subjective metrics (OVL, REL) is achieved by introducing extra sound data, speech data, or both. Additionally, setup (4) contains the most diverse training data (3 domains) and achieves the best subjective evaluation results. All these observations support that learning the out-of-domain speech and sound data can improve the text-to-music task.

3.4.2. Analysis on CodecFormer

As in Section 2.1.1 & 2.3, the adoption of RVQ-based neural codecs has become a popular choice of LLM-based audio generation but causes an overly long sequence issue that needs further consideration. As described in Copet et al. (2023), there are at least 4 approaches to predicting these audio token sequences: (1) *Flattening* prediction, e.g., SPEARTTS (Kharitonov et al., 2023); (2) *Coarse first* prediction, e.g., VALL-E (Wang et al., 2023a)); (3) *Parallel* prediction, e.g., AudioGen (Kreuk et al., 2022); and (4) *Delay* prediction e.g., MusicGen (Copet et al., 2023). We argue that the proposed CodecFormer is an improved implementation of the flattening approach. Then, as shown in Table 6 and 7, we experimentally show that the CodecFormer obtains better modeling quality than the other 3 approaches while achieving better efficiency compared with the naive implementation of flattening (Kharitonov et al., 2023). Our experiments are conducted on text-to-speech and text-to-music tasks.

Auto-Regression and Performance: Among all 4 approaches, Copet et al. (2023) claims that the flattening series methods provide the best audio generation quality. They further conclude that the superior performance of flattening prediction is mainly attributed to the auto-regressive property; the other three methods do not reserve this property as the concurrent prediction is introduced. Under the scenario of codec adoption, we reinterpret the auto-regressive property as: the prediction of the current token z_t^k is based on tokens in prior frames: $\{z_{t'}^{k'} | t' < t\}$ and the tokens in the same frame but in the shallower RVQ layers: $\{z_{t'}^{k'} | t' = t, k' < k\}$. By adopting the full causality in both patch predictor and token predictor, the proposed CodecFormer also satisfies this definition and thus is an implementation of flattening.

Aligned with Copet et al. (2023), our experiments also validate the importance of the auto-regressive property. As in Table 6 and 7, flattening prediction brings better generation quality than parallel, coarse first, and delay prediction. Additionally, with the same auto-regressive property, our proposed CodecFormer achieves a comparable performance with the naive flattening prediction in terms of gen-

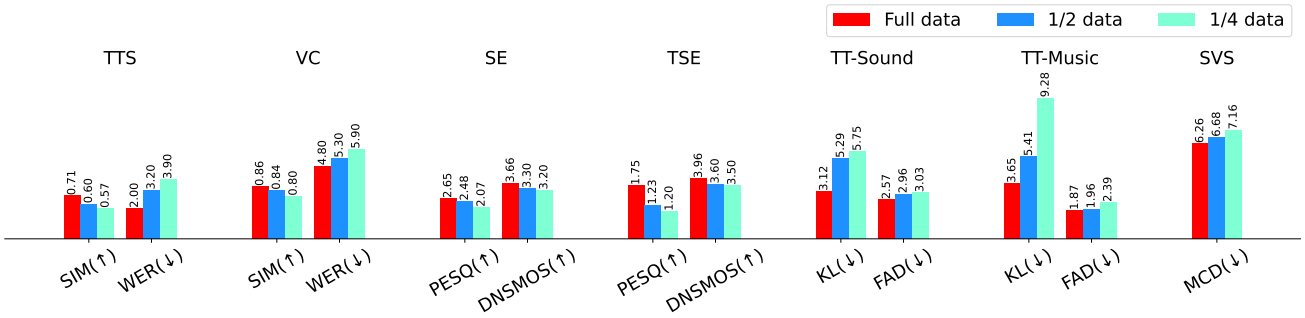


Figure 5. Performance comparison over different data quantity. We use a uniform sampling strategy for each task to select the subsets. The up-arrow denotes higher metrics are better, instead down-arrow denotes lower metrics are better.

Table 5. Ablation study on different domain compositions of training datasets.

Data	FAD (↓)	KL (↓)	OVL. (↑)	REL. (↑)
Music only	5.24	**1.80**	64.4±2.1	66.2±2.4
Music + Sound	4.35	1.93	65.8±1.9	66.5±2.3
Music + Speech	4.66	1.97	64.9±1.7	67.6±2.0
Music + Sound + Speech	**3.65**	1.90	**67.9±1.8**	**70.0±1.5**

eration quality, which, again, validates the importance of auto-regression.

Efficiency: Besides generation quality, efficiency is a major concern of audio generation. Although with the desired auto-regressive property to achieve high quality, the naive flattening prediction is sub-optimal in terms of efficiency: it is a standard Transformer decoder model that works on the $T \times N$ long sequence, which has a space complexity of $O((T * N)^2)$ in self-attention. As increasing N gives higher reconstruction quality at the cost of longer sequences and more computation, this issue becomes more severe when a larger N is adopted. Since the sequence length grows proportionally with N, we experimentally find it difficult to adopt naive flattening with $N \geq 4$. By contrast, the proposed CodecFormer distributes the inter- and intra-frame modeling to the patch predictor and token predictor respectively, which thus alleviates the space complexity to $O(T^2)$. Finally, without the requirement of auto-regression, approaches like parallel, coarse first, and delay predictions achieve better efficiency due to the adoption of concurrent predictions. Since the space complexity is independent of N, training a larger N with the CodecFormer is then feasible.

4. Related Works

This work is an attempt to achieve universal audio generation through LLM-based techniques. There is a long research history for many audio generation tasks. Conventionally, the design of these tasks heavily leverages the domain knowledge of each specific task, and their workflows are distinctive from each other: For tasks like TTS, SE, TSE, VC, S-Edit, SVS, (1) their neural network architectures are based on Transformer (Ren et al., 2020) or others (Oord et al., 2016); (2) their training objectives can be either in time-domain (Luo & Mesgarani, 2019), frequency-domain (Yu et al., 2017) or others (Gu et al., 2021); (3) their designs are inspired by and derived from linguistics and phonetics (Zen et al., 2013), signal processing (Griffin & Lim, 1984), auditory perception (Shadle & Damper, 2001) and machine learning (Wang et al., 2016) research, etc; (4) they use different generative models, such as diffusion model (Shen et al., 2023), and Seq2Seq (Ren et al., 2020).

The prosperity of LLM techniques (Radford et al., 2019; OpenAI, 2023) significantly promotes progress in audio generation research in several directions. First, the large language models, along with the prompt methods, inspired multiple emergent audio generation tasks that are based on textual instruction or descriptions from humans, such as prompt-TTS (Yang et al., 2023a), text-to-sound (Huang et al., 2023c) and text-to-music (Copet et al., 2023; Agostinelli et al., 2023). Second, besides the text, audio can also be tokenized as discrete sequences (Zeghidour et al., 2021; Défossez et al., 2022; Kumar et al., 2023) that can be further processed by LLMs. LLM-based audio generative models then show superior capability in generalization towards unseen speakers (Wang et al., 2023a), low resources (Kharitonov et al., 2023), and multilingual (Zhang et al., 2023) scenarios. These methods also achieve state-of-the-art results in overall performance within their scopes.

It is laborious to handle each audio generation task case-by-case, especially when considering the data shortage as well as the emergent and varying needs in this area. Alternatively, building a universal audio generation model can

Table 6. Model comparison among various audio token prediction methods. Experiments were conducted on the LibriTTS (Zen et al., 2019) dataset. GPU memory and training time are obtained by a 20-second audio (average of 100 trials). All models have a similar parameter budget.

Structure	N	MOS (↑)	MCD (↓)	GPU Mem. (GB)	Time (s) / Iter.
Coarse first	8	3.48±0.05	7.37	18.7	0.58
Parallel	3	3.14±0.07	7.89	13.56	0.53
Delay	3	3.48±0.05	6.95	13.65	0.59
Naive Flattening	3	3.80±0.09	6.56	36.7	1.63
CodecFormer	3	3.77±0.05	6.52	19.4	0.73
CodecFormer	8	3.84 ±0.06	6.27	24.0	1.10

Table 7. Ablation study on text-to-music task using Million Song dataset (McFee et al., 2012). Experiments were conducted on various audio token prediction approaches.

Structure	N	FAD (↓)	KL (↓)	OVL. (↑)	REL. (↑)
Parallel	3	6.92	2.36	60.4±2.3	61.3±1.5
Delay	3	6.07	2.23	62.8±1.9	63.9±1.6
Naive Flattening	3	5.18	1.83	64.8±1.8	65.2±2.0
Ours	3	5.24	1.80	64.4±2.1	66.2±2.4

effectively utilize multiple data sources, which is a promising and practical paradigm. Given the rapid progress in audio generation research, recent designs of audio generation, including LLM-based ones, tend to support multiple audio generation tasks simultaneously. Some pioneer works (Wang et al., 2023c; Le et al., 2023; Jiang et al., 2023; Vyas et al., 2023; Liu et al., 2023a) clearly consider supporting multiple tasks as a key strength; the designs of other prior works (Borsos et al., 2023; Kharitonov et al., 2023; Shen et al., 2023) do show the potential to generate audio in a broader sense than what they originally claim. Following these pioneering research works, UniAudio supports an extended coverage of 11 audio generation tasks in a unified model.

5. Conclusion

To handle the emergent and varying needs in audio generation research, this work advocates building universal audio generation models. We present UniAudio, an LLM-based generative model that unifies a wide range of audio generation tasks. Experimentally, the proposed UniAudio achieves competitive results on all tasks and demonstrates its ability to support newly emergent tasks. Comprehensive ablation studies are also conducted to validate the advantage of building this unified model over the task-specific models. Based on these observations, we primarily validate the feasibility of building universal audio generation models with LLM-based routines.

Impact Statement

This work aims to advance universal audio generation, which will ease the effort of developing multiple task-specific models. The multitask learning on an extensive and diverse set of audio generation tasks, enhancing its capabilities to manipulate different tasks. While our model can produce a myriad of audio content, there's potential for misuse in the generation of misinformation, deepfake audio, or any harmful content. We advocate training a detection model to help humans identify these generated audios.

Acknowledgement

This research is partially affiliated with the CUHK MoE-Microsoft Key Laboratory for Human-centric Computing and Interface Technologies.

References

Agostinelli, A., Denk, T. I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., Huang, Q., Jansen, A., Roberts, A., Tagliasacchi, M., et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

Barrault, L., Chung, Y.-A., Meglioli, M. C., Dale, D., Dong, N., Duquenne, P.-A., Elsahar, H., Gong, H., Heffernan, K., Hoffman, J., et al. Seamlessm4t-massively multilingual & multimodal machine translation. *arXiv preprint arXiv:2308.11596*, 2023.

Borsos, Z., Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Roblek, D., Teboul, O., Grangier, D., Tagliasacchi, M., et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

Casanova, E., Weber, J., Shulby, C. D., Junior, A. C., Gölge, E., and Ponti, M. A. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone.

In *International Conference on Machine Learning*, pp. 2709–2720. PMLR, 2022.

Chen, H., Yu, J., Luo, Y., Gu, R., Li, W., Lu, Z., and Weng, C. Ultra dual-path compression for joint echo cancellation and noise suppression. *arXiv preprint arXiv:2308.11053*, 2023.

Chen, J., Wang, Z., Tuo, D., Wu, Z., Kang, S., and Meng, H. Fullsubnet+: Channel attention fullsubnet with complex spectrograms for speech enhancement. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7857–7861. IEEE, 2022.

Cho, Y.-P., Yang, F.-R., Chang, Y.-C., Cheng, C.-T., Wang, X.-H., and Liu, Y.-W. A survey on recent deep learning-driven singing voice synthesis systems. In *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 319–323. IEEE, 2021.

Conneau, A., Baevski, A., Collobert, R., Mohamed, A., and Auli, M. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*, 2020.

Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y., and Défossez, A. Simple and controllable music generation. *arXiv preprint arXiv:2306.05284*, 2023.

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

Drossos, K., Lipping, S., and Virtanen, T. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 736–740. IEEE, 2020.

Erdogan, H., Wisdom, S., Chang, X., Borsos, Z., Tagliasacchi, M., Zeghidour, N., and Hershey, J. R. Tokensplit: Using discrete speech representations for direct, refined, and transcript-conditioned speech separation and recognition. *arXiv preprint arXiv:2308.10415*, 2023.

Forsgren, S. and Martiros, H. Riffusion-stable diffusion for real-time music generation, 2022. *URL https://riffusion.com/about*, 6, 2022.

Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Ghosal, D., Majumder, N., Mehrish, A., and Poria, S. Text-to-audio generation using instruction-tuned llm and latent diffusion model. *arXiv preprint arXiv:2304.13731*, 2023.

Godfrey, J. J., Holliman, E. C., and McDaniel, J. Switchboard: Telephone speech corpus for research and development. In *Acoustics, speech, and signal processing, ieee international conference on*, volume 1, pp. 517–520. IEEE Computer Society, 1992.

Gong, Y., Chung, Y.-A., and Glass, J. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021.

Griffin, D. and Lim, J. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243, 1984.

Gu, R., Zhang, S.-X., Zou, Y., and Yu, D. Complex neural spatial filter: Enhancing multi-channel target speech separation in complex domain. *IEEE Signal Processing Letters*, 28:1370–1374, 2021.

Guo, Z., Leng, Y., Wu, Y., Zhao, S., and Tan, X. Prompttts: Controllable text-to-speech with text descriptions. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Hao, X., Su, X., Horaud, R., and Li, X. Fullsubnet: A full-band and sub-band fusion model for real-time single-channel speech enhancement. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6633–6637. IEEE, 2021.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012. doi: 10.1109/MSP.2012.2205597.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Huang, J., Ren, Y., Huang, R., Yang, D., Ye, Z., Zhang, C., Liu, J., Yin, X., Ma, Z., and Zhao, Z. Make-an-audio 2: Temporal-enhanced text-to-audio generation. *arXiv preprint arXiv:2305.18474*, 2023a.

Huang, Q., Park, D. S., Wang, T., Denk, T. I., Ly, A., Chen, N., Zhang, Z., Zhang, Z., Yu, J., Frank, C., et al. Noise2music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*, 2023b.

Huang, R., Chen, F., Ren, Y., Liu, J., Cui, C., and Zhao, Z. Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3945–3954, 2021a.

Huang, R., Huang, J., Yang, D., Ren, Y., Liu, L., Li, M., Ye, Z., Liu, J., Yin, X., and Zhao, Z. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. *arXiv preprint arXiv:2301.12661*, 2023c.

Huang, R., Zhang, C., Wang, Y., Yang, D., Liu, L., Ye, Z., Jiang, Z., Weng, C., Zhao, Z., and Yu, D. Make-a-voice: Unified voice synthesis with discrete representation. *arXiv preprint arXiv:2305.19269*, 2023d.

Huang, T.-h., Lin, J.-h., and Lee, H.-y. How far are we from robust voice conversion: A survey. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 514–521. IEEE, 2021b.

Jiang, Z., Ren, Y., Ye, Z., Liu, J., Zhang, C., Yang, Q., Ji, S., Huang, R., Wang, C., Yin, X., et al. Mega-tts: Zero-shot text-to-speech at scale with intrinsic inductive bias. *arXiv preprint arXiv:2306.03509*, 2023.

Jung, J.-w., Heo, H.-S., Tak, H., Shim, H.-j., Chung, J. S., Lee, B.-J., Yu, H.-J., and Evans, N. Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6367–6371, 2022.

Kahn, J., Rivière, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P.-E., Karadayi, J., Liptchinsky, V., Collobert, R., Fuegen, C., et al. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7669–7673. IEEE, 2020.

Kharitonov, E., Vincent, D., Borsos, Z., Marinier, R., Girgin, S., Pietquin, O., Sharifi, M., Tagliasacchi, M., and Zeghidour, N. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *arXiv preprint arXiv:2302.03540*, 2023.

Kim, C. D., Kim, B., Lee, H., and Kim, G. 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)*, pp. 119–132, 2019.

Ko, T., Peddinti, V., Povey, D., Seltzer, M. L., and Khudanpur, S. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5220–5224. IEEE, 2017.

Kondratyuk, D., Yu, L., Gu, X., Lezama, J., Huang, J., Hornung, R., Adam, H., Akbari, H., Alon, Y., Birodkar, V., et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

Kreuk, F., Synnaeve, G., Polyak, A., Singer, U., Défossez, A., Copet, J., Parikh, D., Taigman, Y., and Adi, Y. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*, 2022.

Kumar, R., Seetharaman, P., Luebs, A., Kumar, I., and Kumar, K. High-fidelity audio compression with improved RVQGAN. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=qjnl1QUnFA.

Le, M., Vyas, A., Shi, B., Karrer, B., Sari, L., Moritz, R., Williamson, M., Manohar, V., Adi, Y., Mahadeokar, J., and Hsu, W.-N. Voicebox: Text-guided multilingual universal speech generation at scale. 2023. URL https://dl.fbaipublicfiles.com/voicebox/paper.pdf.

Leng, Y., Guo, Z., Shen, K., Tan, X., Ju, Z., Liu, Y., Liu, Y., Yang, D., Zhang, L., Song, K., et al. Prompttts 2: Describing and generating voices with text prompt. *arXiv preprint arXiv:2309.02285*, 2023.

Liu, A. H., Le, M., Vyas, A., Shi, B., Tjandra, A., and Hsu, W.-N. Generative pre-training for speech with flow matching. *arXiv preprint arXiv:2310.16338*, 2023a.

Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., and Plumbley, M. D. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023b.

Liu, J., Li, C., Ren, Y., Chen, F., and Zhao, Z. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 11020–11028, 2022.

Liu, S., Cao, Y., Wang, D., Wu, X., Liu, X., and Meng, H. Any-to-many voice conversion with location-relative sequence-to-sequence modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 1717–1728, 2021.

Livingstone, S. R. and Russo, F. A. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.

Lu, Y.-J., Wang, Z.-Q., Watanabe, S., Richard, A., Yu, C., and Tsao, Y. Conditional diffusion probabilistic model

for speech enhancement. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7402–7406. IEEE, 2022.

Luo, Y. and Mesgarani, N. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8):1256–1266, 2019.

McFee, B., Bertin-Mahieux, T., Ellis, D. P., and Lanckriet, G. R. The million song dataset challenge. In *Proceedings of the 21st International Conference on World Wide Web*, pp. 909–916, 2012.

Mei, X., Meng, C., Liu, H., Kong, Q., Ko, T., Zhao, C., Plumbley, M. D., Zou, Y., and Wang, W. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *arXiv preprint arXiv:2303.17395*, 2023.

Mesaros, A., Heittola, T., Diment, A., Elizalde, B., Shah, A., Vincent, E., Raj, B., and Virtanen, T. Dcase 2017 challenge setup: Tasks, datasets and baseline system. In *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

Nguyen, T. Q. Near-perfect-reconstruction pseudo-qmf banks. *IEEE Transactions on signal processing*, 42(1): 65–76, 1994.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2204.06125*, 2023.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., and Collobert, R. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*, 2020.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Reddy, C. K., Gopal, V., and Cutler, R. Dnsmos: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6493–6497. IEEE, 2021.

Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T.-Y. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, 2020.

Richter, J., Welker, S., Lemercier, J.-M., Lay, B., and Gerkmann, T. Speech enhancement and dereverberation with diffusion-based generative models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

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

Schneider, F., Kamal, O., Jin, Z., and Schölkopf, B. Mo\^ usai: Text-to-music generation with long-context latent diffusion. *arXiv preprint arXiv:2301.11757*, 2023.

Shadle, C. H. and Damper, R. I. Prospects for articulatory synthesis: A position paper. In *4th ISCA tutorial and research workshop (ITRW) on speech synthesis*, 2001.

Shen, K., Ju, Z., Tan, X., Liu, Y., Leng, Y., He, L., Qin, T., Zhao, S., and Bian, J. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *arXiv preprint arXiv:2304.09116*, 2023.

Shi, Y., Bu, H., Xu, X., Zhang, S., and Li, M. Aishell-3: A multi-speaker mandarin tts corpus and the baselines. *arXiv preprint arXiv:2010.11567*, 2020.

Tae, J., Kim, H., and Kim, T. Editts: Score-based editing for controllable text-to-speech. *arXiv preprint arXiv:2110.02584*, 2021.

Tan, D., Deng, L., Yeung, Y. T., Jiang, X., Chen, X., and Lee, T. Editspeech: A text based speech editing system using partial inference and bidirectional fusion. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 626–633. IEEE, 2021a.

Tan, X., Qin, T., Soong, F., and Liu, T.-Y. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*, 2021b.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Veaux, C., Yamagishi, J., MacDonald, K., et al. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 6:15, 2017.

Vyas, A., Shi, B., Le, M., Tjandra, A., Wu, Y.-C., Guo, B., Zhang, J., Zhang, X., Adkins, R., Ngan, W., et al. Audiobox: Unified audio generation with natural language prompts. *arXiv preprint arXiv:2312.15821*, 2023.

Wang, C., Chen, S., Wu, Y., Zhang, Z., Zhou, L., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023a.

Wang, Q., Muckenhirn, H., Wilson, K., Sridhar, P., Wu, Z., Hershey, J., Saurous, R. A., Weiss, R. J., Jia, Y., and Moreno, I. L. Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking. *arXiv preprint arXiv:1810.04826*, 2018.

Wang, W., Xu, S., Xu, B., et al. First step towards end-to-end parametric tts synthesis: Generating spectral parameters with neural attention. In *Interspeech*, pp. 2243–2247, 2016.

Wang, W., Yang, D., Ye, Q., Cao, B., and Zou, Y. Nadiffuse: Noise-aware diffusion-based model for speech enhancement. *arXiv preprint arXiv:2309.01212*, 2023b.

Wang, X., Thakker, M., Chen, Z., Kanda, N., Eskimez, S. E., Chen, S., Tang, M., Liu, S., Li, J., and Yoshioka, T. Speechx: Neural codec language model as a versatile speech transformer. *arXiv preprint arXiv:2308.06873*, 2023c.

Wang, Y., Wang, X., Zhu, P., Wu, J., Li, H., Xue, H., Zhang, Y., Xie, L., and Bi, M. Opencpop: A high-quality open source chinese popular song corpus for singing voice synthesis. *arXiv preprint arXiv:2201.07429*, 2022.

Wang, Y., Chen, H., Yang, D., Yu, J., Weng, C., Wu, Z., and Meng, H. Consistent and relevant: Rethink the query embedding in general sound separation. *arXiv preprint arXiv:2312.15463*, 2023d.

Wang, Y., Ju, Z., Tan, X., He, L., Wu, Z., Bian, J., and Zhao, S. Audit: Audio editing by following instructions with latent diffusion models. *arXiv preprint arXiv:2304.00830*, 2023e.

Wang, Z., Chen, Y., Xie, L., Tian, Q., and Wang, Y. Lm-vc: Zero-shot voice conversion via speech generation based on language models. *arXiv preprint arXiv:2306.10521*, 2023f.

Wu, B., Li, K., Yang, M., and Lee, C.-H. A reverberation-time-aware approach to speech dereverberation based on deep neural networks. *IEEE/ACM transactions on audio, speech, and language processing*, 25(1):102–111, 2016.

Yang, D., Wang, H., Ye, Z., Zou, Y., and Wang, W. Radur: A reference-aware and duration-robust network for target sound detection. *arXiv preprint arXiv:2204.02143*, 2022.

Yang, D., Liu, S., Huang, R., Lei, G., Weng, C., Meng, H., and Yu, D. Instructtts: Modelling expressive tts in discrete latent space with natural language style prompt. *arXiv preprint arXiv:2301.13662*, 2023a.

Yang, D., Liu, S., Huang, R., Tian, J., Weng, C., and Zou, Y. Hifi-codec: Group-residual vector quantization for high fidelity audio codec. *arXiv preprint arXiv:2305.02765*, 2023b.

Yang, D., Yu, J., Wang, H., Wang, W., Weng, C., Zou, Y., and Yu, D. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023c.

Yu, D., Kolbæk, M., Tan, Z.-H., and Jensen, J. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 241–245. IEEE, 2017.

Yu, L., Simig, D., Flaherty, C., Aghajanyan, A., Zettlemoyer, L., and Lewis, M. Megabyte: Predicting million-byte sequences with multiscale transformers. *arXiv preprint arXiv:2305.07185*, 2023.

Zeghidour, N., Luebs, A., Omran, A., Skoglund, J., and Tagliasacchi, M. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

Zen, H., Senior, A., and Schuster, M. Statistical parametric speech synthesis using deep neural networks. In *2013 ieee international conference on acoustics, speech and signal processing*, pp. 7962–7966. IEEE, 2013.

Zen, H., Dang, V., Clark, R., Zhang, Y., Weiss, R. J., Jia, Y., Chen, Z., and Wu, Y. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

Zhang, L., Li, R., Wang, S., Deng, L., Liu, J., Ren, Y., He, J., Huang, R., Zhu, J., Chen, X., et al. M4singer: A multi-style, multi-singer and musical score provided mandarin singing corpus. *Advances in Neural Information Processing Systems*, 35:6914–6926, 2022.

Zhang, Z., Zhou, L., Wang, C., Chen, S., Wu, Y., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., et al. Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. *arXiv preprint arXiv:2303.03926*, 2023.

Žmolíková, K., Delcroix, M., Kinoshita, K., Ochiai, T., Nakatani, T., Burget, L., and Černockỳ, J. Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures. *IEEE Journal of Selected Topics in Signal Processing*, 13(4):800–814, 2019.

Zmolikova, K., Delcroix, M., Ochiai, T., Kinoshita, K., Černockỳ, J., and Yu, D. Neural target speech extraction: An overview. *IEEE Signal Processing Magazine*, 40(3): 8–29, 2023.

Appendices

Table 8. Dataset adoption of all tasks

Task	Training dataset	Test set	Train Volume (hrs)
Training Stage			
TTS	Librilight	LibriSpeech clean-test	60k
VC	Librilight	VCTK	60k
SE	MLS, Audioset	TUT2017 Task1, VCTK	20k
TSE	MLS	Libri2Mix test set	10k
Sound	AudioCaps, WavCaps	Cloth test set	7k
Music	MSD	MusicCaps	7k
Singing	OpenCPOP, OPenSinger, AISHEELL-3	M4Singer test set	150
Fine-Tuning Stage			
Prompt-TTS	PromptSpeech	PromptSpeech test set	200
Speech dereverberation	LibriTTS, openSLR26, openSLR28	LibriTTS test set	100
Speech edit	LibriTTS	LibriTTS test set	100
Audio edit	AudioCaps, WavCaps	AudioCaps test set	500

A. Experimental Setup

This appendix describes experimental setups in detail, including data statistics, model architecture, and optimization strategy.

A.1. Data Description

12 public datasets are adopted in this work for training. Besides, several test sets are additionally used only for zero-shot evaluation. The statistics of these datasets are in Table 8. Datasets adoption for each task is described in Table 9. Note some datasets are adopted by more than one task.

Table 9. Data statistics

Dataset	Type	Annotation	Volume (hrs)
Training			
LibriLight (Kahn et al., 2020)	speech	-	60k
LibriTTS (Zen et al., 2019)	speech	text	1k
MLS (Pratap et al., 2020)	speech	-	20k
AudioSet (Gemmeke et al., 2017)	sound	-	5.8k
AudioCaps (Kim et al., 2019)	sound	text description	500
WavCaps (Mei et al., 2023)	sound	text description	7k
Million Song Dataset (McFee et al., 2012)	music	text description	7k
OpenCPOP (Wang et al., 2022)	singing	text, MIDI	5.2
OpenSinger (Huang et al., 2021a)	singing	text, MIDI	50
AISHELL3 (Shi et al., 2020)	speech	text	85
PromptSpeech (Guo et al., 2023)	speech	text, instruction	200
openSLR26,openSLR28 (Ko et al., 2017)	room impulse response	-	100
Total	-	-	about 100k
Test			
LibriSpeech test-clean (Panayotov et al., 2015)	speech	text	8
VCTK (Veaux et al., 2017)	speech	text	50
TUT2017 Task1 (Mesaros et al., 2017)	Noise	-	10
Cloth (Drossos et al., 2020)	Sound	text description	3
MusicCaps (Agostinelli et al., 2023)	Music	text description	15
M4Singer(Zhang et al., 2022)	singing	text, MIDI	1

A.2. Model Configuration

The model configuration of the proposed CodecFormer is described in Table 10. We apply pre-layer normalization to the decoder-only model. We set the dropout rate as 0. The learning rate is $6e-4$. For each batch, we set the sequence length as 6000. The activation function is GeLU. The model is trained with AdamW optimizer. We train the model with 2 epoch.

Table 10. Model configuration (with $N = 3$)

Hyper-parameter	Patch Predictor	Token Predictor
#layer	24	8
#Attention dim	1536	1536
#Attention head	12	12
#Feed-Forward dim	6144	6144
#Position encoding	Absolute position	Absolute position
#Norm Types	Layer normalization	Layer normalization
#Params (M)	744	238
Max context length (in #tokens)	3,000	4
Causality	Yes	Yes

A.3. Optimization

The optimization configurations adopted in both the training and fine-tuning stages are presented in Table 11

Table 11. Optimization Configuration using AdamW optimizer

Hyper-parameter	Pre-training	Fine-Tuning
Batch Size (#patches/GPU)	8k	8k
Peak Learning Rate	1e-4	1e-5
AdamW Betas	(0.9, 0.95)	(0.9, 0.95)
Warm-up Steps	100k	1k
Training Steps	800k	50k
Learning rate decay	Noam (Vaswani et al., 2017)	Noam (Vaswani et al., 2017)

B. The Details of Experiments

This section presents detailed experimental results on each task. In the following, if the training set and test sets come from different datasets, we label them as zero-shot settings.

B.1. TTS and VC tasks

For TTS tasks, UniAudio is compared with many previous SOTA models, Table 12 presents the results. Considering the AR-style sampling bring bad cases, we sampling 5 samples for each utterance, and choose a best one by calculating the edit distance using our ASR system. For FastSpeech 2, we only conduct QMOS evaluation as its implementation adopts speaker id as input [6]. We can see that UniAudio obtains better performance in terms of WER, SIM than YourTTS, VALL-E, NaturalSpeech 2, and Make-A-Voice. Compared with VoiceBox, UniAudio also gets comparable performance in terms of objective metrics. From the MOS evaluation, we can see that UniAudio can generate high-quality speech compared with previous SOTA works. Furthermore, UniAudio realizes the best zero-shot clone ability (*e.g.* SMOS is 3.56 and SIM is 0.708). The demos for cross-lingual zero-shot TTS and Mandarin Chinese speech synthesis can be found on the demo page. For the VC task, we conducted experiments on the VCTK dataset, we randomly chose 200 audio pairs. Table 13 shows the results. PPG-VC and YourTTS are trained on small-scale datasets. Make-A-Voice and LM-VC [7] are trained on large-scale datasets as the same as UniAudio. Compared with previous work, UniAudio got better performance in voice conversion tasks.

B.2. Speech Enhancement and Target Speaker Extraction

For the SE task, we compare with previous SOTA methods, including discriminative methods (such as FullSubNet and FullSubNet+) and generative methods (such as SGMSE+ and NADiffuSE). Note that the CDiffuSE and NADiffuSE are both trained on the voicebank-demand dataset. Other models never saw the VCTK dataset in the training stage. We obtain the inference results based on their open-source models. Table 14 presents the results, we can see that UniAuido obtains the

[6]https://github.com/ming024/FastSpeech2

[7]We seek help from the authors, they provide the inference results.

Table 12. The performance comparison with previous SOTA methods in TTS and VC tasks. We do not conduct MOS evaluation for VALL-E, SPEARTTS and VoiceBox due to the models are not released. SIM-o denotes that we calculate the similarity between generated samples and ground truth. Without specifically stated, we follow VALL-E (Wang et al., 2023a), calculate the simialarity between generated smaples and reconstructed one (SIM-r).

Model	Zero-shot	SIM-r (↑)	SIM-o (↑)	WER (↓)	MOS (↑)	SMOS (↑)
Text-to-Speech						
GroundTruth	-	-	-	1.9	3.99±0.08	-
FastSpeech 2 (Ren et al., 2020)	✗	-	-	-	3.81±0.10	-
YourTTS (Casanova et al., 2022)	✓	0.337	0.31	7.7	3.66±0.07	3.02±0.07
VALL-E (Wang et al., 2023a)	✓	0.580	-	5.9	-	-
Make-A-Voice (TTS) (Huang et al., 2023d)	✓	0.498	0.45	5.7	3.74±0.08	3.11±0.06
NaturalSpeech 2 (Shen et al., 2023)	✓	0.62	0.55	2.3	**3.83±0.10**	3.11±0.10
SPEAR-TTS (Kharitonov et al., 2023)	✓	0.560	-	/	-	-
VoiceBox (Le et al., 2023)	✓	0.681	**0.66**	**1.9**	-	-
UniAudio	✓	**0.708**	0.56	2.0	3.81±0.07	**3.56±0.10**

Table 13. The performance comparison with previous SOTA methods in VC task.

Model	Zero-shot	SIM (↑)	WER (↓)	MOS (↑)	SMOS (↑)
Voice Conversion					
GroundTruth	-	-	3.25	3.74±0.08	-
PPG-VC (Liu et al., 2021)	✗	0.78	12.3	3.41±0.10	3.47±0.10
YourTTS (Casanova et al., 2022)	✓	0.719	10.1	3.61±0.10	3.26±0.10
Make-A-Voice (VC) (Huang et al., 2023d)	✓	0.678	6.2	3.43±0.09	3.47±0.10
LM-VC (Wang et al., 2023f)	✓	0.820	4.91	3.41±0.08	3.17±0.09
UniAudio	✓	**0.868**	4.8	3.54±0.07	**3.56±0.07**

best DNSMOS score. The PESQ and VISQOL scores are lower than other SOTA methods, we think these metrics may not accurately assess the performance of generative methods. A similar finding is also observed in previous literature (Erdogan et al., 2023) that the signal-level evaluation metrics may not be suitable for generative methods. In contrast, we recommend using DNSMOS and MOS scores as the main metrics. UniAuido can get good results in extremely noisy environments, we recommend readers refer to the demo page. For the TSE task, we conducted experiments on the LibriMix test set. The popular TSE systems: VoiceFilter [8] and SpeakBeam[9] are used as baseline systems. As Table 14 shows, we can see that UniAudio obtains the best performance in terms of DNSMOS and MOS.

Table 14. The performance of SE and TSE tasks comparison with previous SOTA methods.

Model	Zero-shot	PESQ (↑)	VISQOL(↑)	DNSMOS(↑)	MOS(↑)
Speech Enhancement					
CDiffuSE (Lu et al., 2022)	✗	1.88	1.21	2.54	-
NADiffuSE (Wang et al., 2023b)	✗	2.96	2.41	3.03	3.30±0.08
SGMSE+ (Richter et al., 2023)	✓	3.21	2.72	3.29	3.56±0.08
FullSubNet (Hao et al., 2021)	✓	3.21	2.77	3.37	3.61±0.10
FullSubNet+ (Chen et al., 2022)	✓	3.41	2.99	3.34	3.42±0.08
UniAudio	✓	2.63	2.44	**3.66**	**3.68±0.07**
Target Speaker Extraction					
SpeakerBeam (Žmolíková et al., 2019)	✗	2.89	2.25	3.18	3.68±0.1
VoiceFilter (Wang et al., 2018)	✗	2.41	2.36	3.35	3.43±0.09
UniAudio	✓	1.88	1.68	**3.96**	**3.72±0.06**

B.3. Singing Voice Synthesis

Following Make-A-Voice, we conduct experiments on the M4Singer test set. We compare the generated singing samples with other systems, including 1) Diffsinger; and 2) Make-A-Voice, a two-stage audio language model for singing voice generation. As illustrated in Table 15, we can see that UniAudio gets comparable results with Make-A-Voice and Diffsinger.

[8]https://github.com/Edresson/VoiceSplit

[9]https://github.com/BUTSpeechFIT/speakerbeam

Table 15. Quality and style similarity of generated samples in singing voice synthesis.

Model	MOS (↑)	SMOS (↑)
Diffsinger (Liu et al., 2022)	3.94±0.02	**4.05±0.06**
Make-A-Voice (Huang et al., 2023d)	3.96±0.03	4.04±0.05
UniAudio	**4.08±0.04**	4.04±0.05

B.4. Text-to-sound and text-to-music generation

The text-to-sound generation task has attracted great interest in audio research. Following Diffsound (Yang et al., 2023c), most of the methods evaluate their systems on the AudioCaps (Kim et al., 2019) test set. However, we found that if the training data includes the AudioCaps data, the model is easy to overfit with AudioCaps. As a result, the best performance can be obtained when the model only trains on the Audiocaps. In this study, we conduct a zero-shot evaluation on the Cloth test set (Drossos et al., 2020). Table 16 shows the results. We can see that UniAudio obtains better performance than Diffsound and AudioLDM. Compared to recent SOTA models, such as Tango and Make-an-Audio 2, UniAudio also gets comparable performance. For the text-to-music task, we follow MusicGen (Copet et al., 2023), evaluating our methods on MusicCaps (Agostinelli et al., 2023). Compared with previous SOTAs, UniAudio gets a comparable performance with other models. From the MOS evaluation performance, we can see that MusicGen is better than our current models. We speculate one of the reasons is that MusicGen uses a large-scale high-quality dataset (20k hours).

Table 16. Text-to-sound and text-to-music evaluation. We report the subjective metrics including FAD(↓), and KL(↓). Furthermore, we also conduct objective evaluation. Note that the training data of AudioGen includes Cloth datatset, thus can not be seen as zero-shot setting.

Model	Training Data (Hours)	FAD	KL	OVL.	REL.
Text-to-Sound Generation					
Reference	/	/	/	70.47±1.9	78.84±1.5
Diffsound (Yang et al., 2023c)	2k	7.8	6.53	-	-
AudioGen (Kreuk et al., 2022)	4k	2.55	2.5	63.84±2.1	**72.12±1.8**
Tango (Ghosal et al., 2023)	3.3k	3.61	2.59	**66.2±1.7**	68.57±1.5
Make-an-Audio 2 (Huang et al., 2023a)	8.7k	2.13	2.49	61.52±1.6	69.9±1.5
AudioLMD (Liu et al., 2023b)	9k	4.93	2.6	60.95±1.9	65.7±1.8
UniAudio	7k	3.12	2.57	61.9±1.9	66.1±1.5
Text-to-Music Generation					
Riffusion (Forsgren & Martiros, 2022)	-	14.8	2.06	-	-
Mousai (Schneider et al., 2023)	-	7.5	1.59	-	-
MusicLM (Agostinelli et al., 2023)	280k	4.0	-	-	-
Noise2Music (Huang et al., 2023b)	280k	2.1	-	-	-
MusicGen (Copet et al., 2023)	20k	4.52	1.41	**73.28±1.5**	**71.28±1.7**
UniAudio	8k	3.65	1.87	67.85±1.70	70.0±1.5

B.5. Audio Edit

Audio edit aims to edit the original audio based on Human's instruction. AUDIT (Wang et al., 2023e) is the SOTA model in audio edit task, which designs a data simulation strategy to get triplet training and test data (*e.g.*, {audio, audio, text}). The authors set 5 different tasks, including adding, dropping, replacing, inpainting, and super-resolution, and simulated large-scale data for each task. To validate that our pre-trained model can be fine-tuned with small-scale data, we choose adding, dropping, and super-resolution tasks to fine-tune simultaneously. To finish the fine-tuning process, we define a new task label: *Audit_task*. The experimental results as Table 17 shows. We can observe that: (1) UniAudio can get better performance with the previous SOTA model. (2) Fine-tuning pre-trained UniAudio can get better performance than training it from scratch, which further validates the effectiveness of pre-training a model on large-scale training data.

Table 17. Audio edit task evaluation.

Type	Model	FD	KL
Adding task			
	AUDIT	21.80	**0.92**
	UniAudio (scratch)	20.2	0.99
	UniAudio (fine-tune)	**19.69**	0.934
Dropping task			
	AUDIT	22.40	0.95
	UniAudio (scratch)	27.76	1.38
	UniAudio (fine-tune)	**23.1**	**1.10**
Super-Resolution task			
	AUDIT	18.14	0.73
	UniAudio (scratch)	11.51	0.29
	UniAudio (fine-tune)	**10.54**	**0.289**

Table 18. Quality and style similarity of generated samples for Instructed TTS task.

Model	MOS (↑)	SMOS (↑)
GT	3.77±0.07	3.85±0.08
UniAudio (scratch)	3.62±0.07	3.67±0.08
UniAudio (tuning)	3.61±0.09	3.71±0.09

B.6. Prompt TTS

Using instruction to guide speech synthesis has received great attention (Guo et al., 2023; Yang et al., 2023a; Leng et al., 2023). In this part, we fine-tune the UniAudio model on the PromptSpeech (Guo et al., 2023) dataset. Furthermore, we also try to train a UniAudio model from scratch with the PromptSpeech dataset. Different from previous works that designed special style encoders to capture the style information from text descriptions, we directly use the T5 text encoder to extract representations from text and then combine it with the phoneme sequence input to the UniAudio, which is more convenient. [10] Table 18 shows the results, we can see that UniAudio has good performance in terms of style control and speech quality when compared with the ground truth samples.

B.7. Speech Dereverberation

For the speech dereverberation task, we use the Room Impulse Response (RIR) data from the openSLR26 and openSLR28 datasets and the speech data from the LibriTTS clean part. We simulate about 100 hours of training data and 1 hour of test data. We compare with previous SOTA systems, such as FullSubNet, FullSubNet+, and SGMSE+. Table 19 presents the results. We can see that UniAudio obtains the SOTA performance in speech dereverberation tasks with small-scale training data in terms of the DNSMOS metric. Similar to the speech enhancement task, we speculate that PESQ may not be suitable for the generative methods.

Table 19. Results comparison with previous speech Dereverberation systems.

Model	PESQ (↑)	DNSMOS(↑)
SGMSE+	2.87	3.42
FullSubNet	2.29	3.32
FullSubNet+	2.27	3.25
UniAudio (scratch)	1.23	3.18
UniAudio (tuning)	2.13	**3.51**

[10] PromptTTS is not compared here as their implementation is not publicly available.

B.8. Speech Edit

For the speech edit task, we use the LibriTTS dataset. In practice, we randomly choose some words to mask in the training stage. We expect the model to recover the whole speech based on the phoneme sequence. In the inference stage, we can mask the region that we want to update in the speech and input the new words so that the model can edit the speech. For this task, we take the TTS system that regenerates a complete waveform from the whole sentence to be edited as the baseline. In the evaluation, we mainly validate three situations: (1) word replacement; (2) insert a new word; and (3) delete a word. For each situation, we randomly chose 10 sentences from the LibriTTS test clean set.

C. Ablation study

C.1. Fine-tuning the pre-trained model on the new task will influence the performance on previous tasks?

In this part, we conduct experiments to explore whether fine-tuning the pre-trained model on new tasks will influence the performance of previous tasks. We evaluate the pre-trained UniAudio model (trained on 7 tasks) and fine-tuned UniAudio model (fine-tuned on 4 new tasks) on 7 tasks. Figure 6 shows the results. We can see that the performance does not significantly drop on previous training tasks, which demonstrates that UniAudio has the potential to add new tasks continuously without losing previous task knowledge.

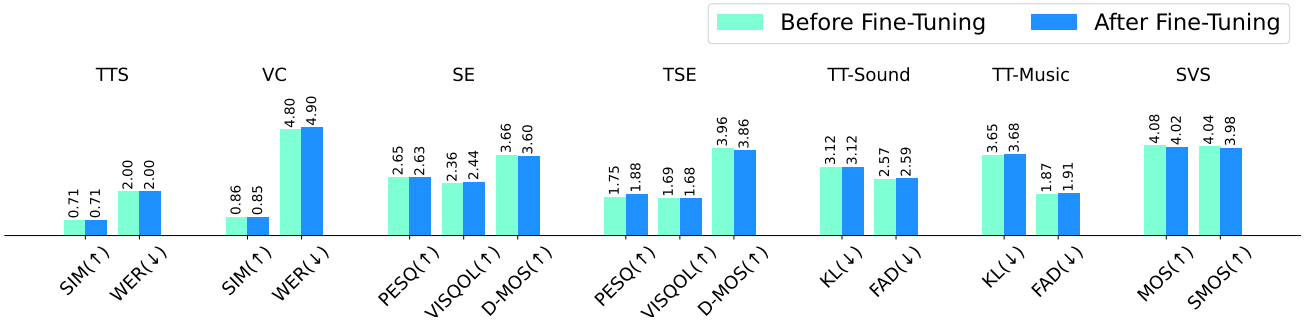


Figure 6. Performance comparison over 7 audio generation tasks before/after fine-tuning. The up-arrow denotes higher metrics are better, instead down-arrow denotes lower metrics are better.

C.2. The details of ablation study of multi-task training

In this part, we present the comparison between training UniAudio with each task separately and with all tasks jointly. The details are shown in Table 20.

D. The details of Audio Codec Models

In this part, we give more details about our neural audio codec model in Section 2.1.1. We adopt a similar encoder-decoder framework to the Encodec model, but we design a different discriminator and propose a sub-band STFT reconstruction loss. In the following, we present the architecture and the training loss.

D.1. Model Architecture

Figure 7 shows the details of the mel-based discriminator. We combine the mel-spectrogram and log-mel-spectrogram features and then input them into a network consisting of several convolutional layers. Our motivation is that the mel-spectrogram has a strong intrinsic inductive bias, especially for sounds and music-related audio (Gong et al., 2021; Yang et al., 2022). Thus, we speculate that choosing a mel-spectrogram-based discriminator can better promote high-fidelity audio reconstruction. In our experiments, we use 6 different discriminators with different configurations [11]. Specifically, we

[11] In our experiments, we find the mel-based discriminator brings better reconstruction performance when we train a universal neural audio codec.

Table 20. The ablation study of the effectiveness of multi-task training. UniAudio (Single) means the comparable model trained on one task only.

Task	Model	Objective Evaluation Metrics	Objective Evaluation Results	Subjective Evaluation Metrics	Subjective Evaluation Results
Text-to-Speech	UniAudio (Single)	SIM(↑) / WER(↓)	0.64 / 2.4	MOS(↑) / SMOS(↑)	3.77±0.06 / 3.46±0.10
	UniAudio		**0.71 / 2.0**		**3.81±0.07 / 3.56±0.10**
Voice Conversion	UniAudio (Single)	SIM(↑) / WER(↓)	0.84 / 5.4	MOS(↑) / SMOS(↑)	3.45±0.07 / 3.44±0.07
	UniAudio		**0.87 / 4.8**		**3.54±0.07 / 3.56±0.07**
Speech Enhancement	UniAudio (Single)	PESQ(↑) / VISQOL(↑) / DNSMOS(↑)	2.35 / 2.30 / 3.45	MOS(↑)	3.65±0.08
	UniAudio		**2.63 / 2.44 / 3.66**		**3.68±0.07**
Target Speaker Extraction	UniAudio (Single)	PESQ(↑) / VISQOL(↑) / DNSMOS(↑)	**1.97** / 1.61 / 3.90	MOS(↑)	3.58±0.08
	UniAudio		1.88 / **1.68 / 3.96**		**3.72±0.06**
Singing Voice Synthesis	UniAudio (Single)	-	-	MOS(↑) / SMOS(↑)	**4.14±0.07** / 4.02±0.02
	UniAudio				4.08±0.04 / **4.04±0.05**
Text-to-Sound	UniAudio (Single)	FAD (↓) / KL (↓)	3.84 / 2.7	OVL (↑) / REL (↑)	60.0±2.1 / 61.2±1.8
	UniAudio		**3.12 / 2.6**		**61.9±1.9 / 66.1±1.5**
Text-to-Music	UniAudio (Single)	FAD (↓) / KL (↓)	5.24 / **1.8**	OVL (↑) / REL (↑)	64.4±2.1 / 66.2±2.4
	UniAudio		**3.65** / 1.9		**67.9±1.7 / 70.0±1.5**
Audio Edit	UniAudio (single)	FD (↓) / KL (↓)	19.82 / 0.92	-	-
	UniAudio		**17.78 / 0.77**		-
Speech Dereverb.	UniAudio (single)	PESQ(↑) / DNSMOS(↑)	1.23 / 3.18	-	-
	UniAudio		**2.13 / 3.51**		-
Instructed TTS	UniAudio (single)	-	-	MOS(↑) / SMOS(↑)	**3.62±0.07** / 3.67±0.08
	UniAudio		-		3.61±0.09 / **3.71±0.09**
Speech Edit	UniAudio (single)	MCD (↓)	5.26	MOS(↑)	3.73±0.07
	UniAudio		**5.12**		**3.82±0.06**

set the hidden_dim as {64, 128, 256, 512, 512, 512} and the hop length as {32, 64, 128, 256, 512, 1024}.

D.2. Training Loss

The audio codec mainly consists of generators and discriminators. For discriminators, it trained with the discriminator loss d:

$$\mathcal{L}_d = \frac{1}{K_d}\sum_{i=1}^{K_d} max(0, 1 - D_k(\boldsymbol{x})) + max(0, 1 + D_k(\hat{\boldsymbol{x}})) \tag{1}$$

where x and $\hat{x}$ denotes the input waveform and the reconstructed one. K_d denotes the number of discriminators. For generator, it trained with following terms: the adversarial loss $\mathcal{L}_{adv}$, feature loss $\mathcal{L}_{feat}$, the commitment Loss $\mathcal{L}_c$ and reconstruction loss $\mathcal{L}_{rec}$.

$$\mathcal{L}_{adv} = \frac{1}{K_d}\sum_{i=1}^{K_d} max(0, 1 - D_k(\hat{\boldsymbol{x}})) \tag{2}$$

$$\mathcal{L}_{feat} = \frac{1}{K_d * L}\sum_{k=1}^{K_d}\sum_{l=1}^{L} \frac{||D_k^l(\boldsymbol{x}) - D_k^l(\hat{\boldsymbol{x}})||_1}{mean(||D_k^l(\boldsymbol{x})||_1)} \tag{3}$$

$$\mathcal{L}_c = \sum_i ||\boldsymbol{z}_i - q_i(\boldsymbol{z}_i)||_2^2 \tag{4}$$

$$\mathcal{L}_{rec} = \frac{1}{N_s}\sum_{i=1}^{N_s} L_i(x, \hat{x}) \tag{5}$$

where L denotes the number of layers in a discriminator. z_i denotes the latent features produced by the q_i quantizer. L_i denotes the i-th STFT sub-bands extractor. N_s denotes the number of sub-bands. Band-split (Nguyen, 1994) aims to divide

the spectrogram into multiple sub-bands of predefined bandwidth. Recently, many works (Wang et al., 2023d; Chen et al., 2023) found that splitting a spectrogram into multiple sub-bands is useful for speech or music separation tasks because it can effectively split import information into several sub-bands. Inspired by these works, a sub-band STFT reconstruction loss is proposed. The motivation is that sub-bands STFT reveal more spectrogram details, which can improve the reconstruction performance. In our study, we set $N_s = 6$.

D.3. Training dataset

We carefully select the training dataset for different types of audio. For speech data, we use the Librilight dataset. For sound and music data, we choose the audio from AudioSet. Note that we remove all speech data from AudioSet, and only retain the sound and music-related data based on the label information. We do not choose a special singing voice dataset for codec training. However, we find the codec model can be well generalized to the singing voice data.

D.4. Training configurations

We train the audio codec models with the learning rate of $1e-4$, the batch size is 12 for each GPUs (we use 8 V100-32G). We set a fixed audio segment (2 seconds) during the training. We train the model with AdamW optimizer, exponentially decayed learning rate, with $\gamma = 0.99$. We set the training step as 200k.

D.5. Results Analysis

To evaluate the reconstruction performance for different types of audio, we choose 4 different test sets: VCTK (speech), Cloth (sound), MusicCaps (music), and M4Sing (singing voice). For each set, we randomly choose 100-200 audios. Table 2 shows the results. We can see that previous audio codec models perform poorly with few VQ layers, *e.g.* if the number of RVQ layers is less than 4, the performance of Encodec and DAC is significantly decreasing. In contrast, the proposed audio codec has better reconstruction performance even only using 3 VQ layers. This advantage is very useful for LLM-based models for the reason that it effectively reduces the length of the target sequence.

D.6. Discussion

Recently, LLM-based audio generation has attracted great interest in the research community. Audio codec models play a very important role, which effectively transfer the continuous audio signal into a token sequence. In theory, the reconstruction performance is the limit of the generation model. Thus developing a good audio codec model is very important. Nowadays, the reconstruction performance of audio codec models relies on the number of VQ layers, as Table 2 shows. A core problem is that using more VQ layers will increase the target token sequence. Considering current LLM models are autoregressive and the backbone is Transformer, a long sequence inevitably increases the training and inference costs. To overcome this issue, previous works, such as VALL-E and AudioLM try to develop a two-stage generation model. We agree that such settings are useful and effective. In this study, we explore to develop a one-stage unified model. Thus, designing a few RVQ layers codec is useful.
All of the training process and pre-trained models will be released in the future.

E. Evaluation Metrics

For all of objective evaluation, we calculate the score based on the generated samples and the target audio resynthesized using the codec models. For Word Error Rate (WER), we follow VALL-E (Wang et al., 2023a), using a pre-trained HuBERT-Large model, then fine-tuning it on LibriSpeech dataset by using ESPNet tools. For speaker similarity (SIM), we use WavLM-TDCNN [12] to extract the speaker embedding. For FAD, FD and KL, we follow previous text-to-sound works (Yang et al., 2023c; Wang et al., 2023e), using the same models to evaluate the performance. The details of Subjective evaluation as follows.

[12]https://huggingface.co/docs/transformers/model_doc/wavlm

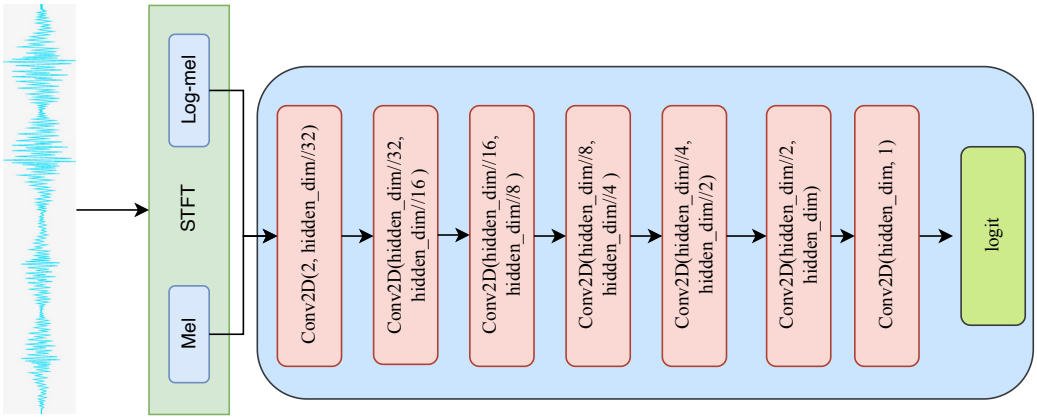


Figure 7. The overview of a single Mel-based discriminator. In practice, we will use multiple discriminators by setting different hop lengths and hidden dimensions.

E.1. Subjective Evaluation

For TTS and VC tasks, we focus on speech quality (QMOS) and speaker similarity (SMOS). The details are as follows. For speech quality evaluation, we conduct the MOS (mean opinion score) tests and explicitly ask the raters to *focus on examining the audio quality and naturalness, and ignore the differences of style (timbre, emotion, and prosody*. The testers present and rate the samples, and each tester is asked to evaluate the subjective naturalness on a 1-5 Likert scale.

For speaker similarity evaluation, we ask the raters to *focus on the similarity of the speaker identity (timbre) to the reference, and ignore the differences in content, grammar, or audio quality*. We paired each synthesized utterance with a reference utterance to evaluate how well the synthesized speech matched that of the target speaker.

For SE and TSE tasks, we write explicit instructions to ask the rater to assess the generated speech. Refer to Figure 8 to see the details.

For SVS, we also conduct quality MOS (QMOS) and style similarity MOS (SMOS). Different from TTS's SMOS evaluation, we explicitly instruct the raters to *focus on the similarity of the style (timbre, emotion, and prosody) to the reference, and ignore the differences in content, grammar, or audio quality*.

For sound and music generation tasks, we follow AudioGen (Kreuk et al., 2022) and MusicGen (Copet et al., 2023) to evaluate (1) overall quality (OVL), and (2) relevance to the text input (REL).

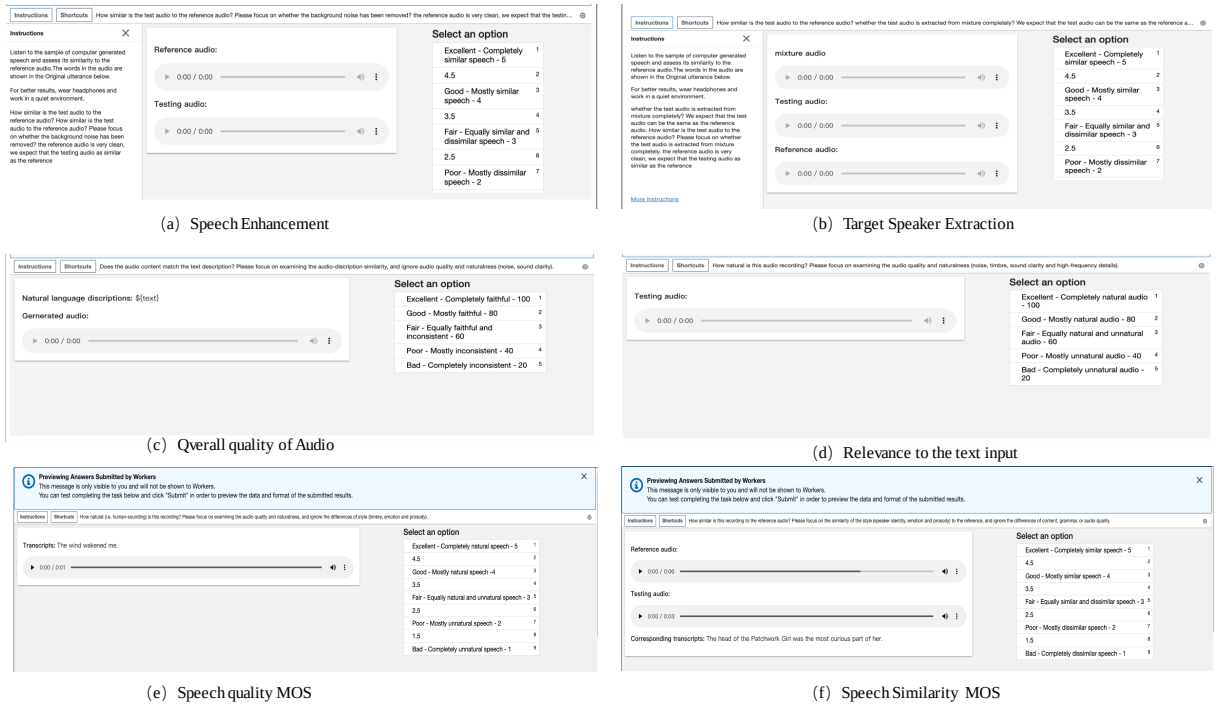
Our subjective evaluation tests are crowd-sourced and conducted by 20 native speakers via Amazon Mechanical Turk. The screenshots of instructions for testers have been shown in Figure 8. We paid about $500 on participant compensation. A small subset of speech samples used in the test is available at `http://dongchaoyang.top/UniAudio_demo/`.

F. Limitation

Not all known audio generation tasks are included in the proposed UniAudio, such as noise removal, noise speech edit (Wang et al., 2023c) and speech-to-speech translation (Rubenstein et al., 2023; Barrault et al., 2023). All new tasks added in fine-tuning are formulated with the known modalities in the training stage; Introducing new modalities during fine-tuning is unexplored in this work. Current UniAudio considers neither unlabeled data nor domain-specific foundation models, which can possibly further improve the overall performance. The samples generated by UniAudio are not guaranteed in quality and may contain errors.

G. Ethical Statement

We are devoted to prevent our AI models from being abused. Specifically, we demonstrate that the audio generated from our codec model and UniAudio can be easily distinguished from the natural audio. We train a binary classification model using the training recipe provided in (Jung et al., 2022), which only contains 297k parameters. The dataset contains three types of



(a) Speech Enhancement

(b) Target Speaker Extraction

(c) Qverall quality of Audio

(d) Relevance to the text input

(e) Speech quality MOS

(f) Speech Similarity MOS

Figure 8. Screenshots of subjective evaluations.

Precision	Recall	F1 Score
0.998	0.998	0.998

Table 21. Binary classification results of detecting the synthesized audio

data: (1) speech from humans; (2) speech resynthesized from Codec model and (3) generated speech from UniAudio. All types of speech are derived from the combination of LibriSpeech Test, Dev-Clean, Other subsets with a ratio of 2:1:1. (1) is considered positive examples while (2) and (3) are considered negative examples. We randomly reserve 1k examples for validation and testing respectively; other examples are used for training. We select the decision threshold on the validation set and apply it to the test set. The accuracy, recall, and F1-score for this binary classification task are as in Table 21. Based on these results, we demonstrate that detecting the synthesized audio from UniAudio model is a trivial task.

H. Additional Experimental Results During the Peer Review Stage

We received many constructive comments during the peer review stage and did experiments accordingly. We summarize our experiments and the corresponding findings during this stage as follows.

H.1. Impact of model size

We demonstrate that, with the same training data, increasing model size consistently leads to performance improvement, as shown in Table 22.

H.2. The Mutual Benefits of Training Unified Speech Generation Models across Modalities

In Section 3.4.1, we demonstrate including speech data contributes to the performance of the music generation task. In the tables below, we provide two further examples to demonstrate: (1) music data contributes to the sound generation task and (2) music data contributes to the speech generation task. Results are in Table 23.

Model	TTS SIM(↑)	VC SIM(↑)	TSE DNSMOS(↑)	SE DNSMOS(↑)	Text-to-Sound FAD(↓)	Text-to-Music FAD(↓)
Small (200M)	0.619	0.850	3.70	3.49	4.03	5.72
Medium (200M)	0.640	0.857	3.79	3.52	3.89	5.06
Large (1B)	**0.708**	**0.868**	**3.96**	**3.66**	**3.65**	**4.31**

Table 22. Model performance with varying parameter scales

Data Setup	Text-to-Sound		Data Setup	Text-to-Speech	
	FAD(↓)	KL (↓)		WER(↓)	SIM(↑)
Sound	3.84	2.70	Speech	2.40	0.64
Sound + Music	**3.42**	**2.66**	Music + Speech	2.40	**0.65**

Table 23. Mutual Benefits of Training Unified Speech Generation Models across Modalities

H.3. Additional Zero-shot TTS evaluation

we additionally test our model on conversational test sets SwitchBoard (Godfrey et al., 1992), and expressive test sets RAVDESS (Livingstone & Russo, 2018) for TTS task. For the SwitchBoard dataset, the test set includes 80 different speakers, we randomly choose two utterances for each speaker. We use the first one as an audio prompt, and the second one as a target. We try to evaluate our model's zero-shot cloning ability on the SwitchBoard dataset. We choose our reproduced VALL-E (Wang et al., 2023a) model as the baseline. We use the speaker similarity score as the metric. The results are as Table 24 shows:

RAVDESS dataset is an emotional TTS dataset featuring 24 professional actors across 8 emotions (neutral, calm, happy, sad, angry, fearful, surprise, and disgust). This dataset provides speech samples with the same text from the same speaker across eight different emotions. We aim to evaluate the ability to clone emotion from the prompt. The test set includes 192 utterances (24 speakers, 8 different emotions from each speaker). Similarly, we choose a different utterance from the same speaker with the same emotion as the prompt, expecting the model to clone the emotion from the utterance. To evaluate the performance, we use Mel-Cepstral Distortion(MCD) for prosody evaluation by measuring the differences between generated samples and ground truth samples. Furthermore, we also train an emotion classification model on the RAVDESS dataset. We use classification accuracy as one metric to assess the emotion transfer ability. The results are as Table 25 shows.

Table 24. The zero-shot TTS evaluation on SwitchBoard.

Model	Similarity Score ↑
VALL-E	0.71
ours	**0.77**

Table 25. The zero-shot TTS evaluation on RAVDESS dataset.

Model	MCD ↓	Accuracy ↑
VALL-E	4.84	60.4
Ours	**4.42**	**68.7**
GT	-	87.2