Appendices

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 6. Datasets adoption for each task is described in Table 7. Note some datasets are adopted by more than one task.

Table 6: 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
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

Table 7: 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			
I-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
Sum	-	-	166k

A.2 Model Configuration

The model configuration of the proposed multi-scale Transformer is described in Table 8.

Table 8: Model configuration (with $n_q = 3$)

Hyper-parameter	Global Transforemr	Local Transformer
#layer	24	8
#Attention dim	1536	1536
#Attention head	12	12
#Feed-Forward dim	6144	6144
#Params (M)	744	238
Max context length (in #tokens)	3,000	3
Causality	Yes	Yes

A.3 OPTIMIZATION

The optimization configurations adopted in both the training and fine-tuning stages are presented in Table 9

Table 9: Optimization Configuration

Hyper-parameter	Pre-training	Fine-Tuning
Batch Size (#patches/GPU)	8k	8k
Peak Learning Rate	1e-4	1e-5
Warm-up Steps	10000	1000
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 the many previous SOTA models, Table 10 presents the results. For FastSpeech 2, we only conduct QMOS evaluation as its implementation adopts speaker id as input [12]. 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). More experiments, such as cross-lingual zero-shot TTS and Mandarin Chinese speech synthesis can be found in demo page. For VC task, we conducted experiments on VCTK dataset, we randomly chose 200 audio pairs. PPG-VC and YourTTS are trained on small-scale datasets. Make-A-Voice and LM-VC [13] 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 11 presents the results, we can see that UniAuido obtains the best DNSMOS score. The PESQ and VISQOL scores are lower than other SOTA methods, we think these metrics may not

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

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

Table 10: 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.

Model	Zero-shot	SIM (↑)	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	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., 2023b)	✓	0.498	5.7	3.74±0.08	3.11±0.06
NaturalSpeech 2 (Shen et al., 2023)	✓	0.620	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	**1.9**	-	-
UniAudio	✓	**0.708**	2.0	3.81±0.07	**3.56±0.10**
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., 2023b)	✓	0.678	6.2	3.43±0.09	3.47±0.10
LM-VC (Wang et al., 2023e)	✓	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**

accurately assess the performance of generative methods. The 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 [14] and SpeakBeam[15] are used as baseline systems. As Table 11 shows, we can see that UniAudio obtains the best performance in terms of DNSMOS and MOS.

Table 11: 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; 2) Make-A-Voice, a two-stage audio language model for singing voice generation. As illustrated in Table 12, we can see that UniAudio gets comparable results with Make-A-Voice and Diffsinger.

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)

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

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

Table 12: 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., 2023b)	3.96±0.03	4.04±0.05
UniAudio	**4.08±0.04**	4.04±0.05

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 13 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 13: 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	2k	7.8	6.53	-	-
AudioGen	4k	2.55	2.5	63.84±2.1	**72.12±1.8**
Tango	3.3k	3.61	2.59	**66.2±1.7**	68.57±1.5
Make-an-Audio 2	8.7k	2.13	2.49	61.52±1.6	69.9±1.5
AudioLMD	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	-	14.8	2.06	-	-
Mousai	-	7.5	1.59	-	-
MusicLM	280k	4.0	-	-	-
Noise2Music	280k	2.1	-	-	-
MusicGen	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., 2023d) 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 14 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.

B.6 INSTRUCTED TTS

Using instruction to guide speech synthesis has received great attention (Guo et al., 2023; Yang et al., 2023a). In this part, we fine-tune the UniAudio model on the PromptSpeech (Guo et al., 2023) dataset.

Table 14: 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 15: 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

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.[16] Table 15 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 dataset, 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 16 presents the results. We can see that UniAudio obtains the SOTA performance in speech dereverberation tasks with small-scale training data in terms of DNSMOS metric. Similar with speech enhancement task, we speculate that PESQ may not suitable for the generative methods.

Table 16: 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**

[16]Note that the authors of PromptTTS (Guo et al., 2023) told us their objective metrics tools, checkpoints, and generated samples have been lost due to the machine errors. Thus we cannot fairly compare with them.

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 THE INFLUENCE OF MULTI-TASK TRAINING

In this part, we explore whether multi-task training can bring better performance than task-specific training. To answer this question, we use the same model trained on different tasks, respectively. Table 17 shows the experimental results, UniAudio (single) means that the model is trained on a single task. We observe that multi-task training brings the gain over all of the tasks. In Appendix D, we give some potential reasons why multi-task training can bring improvement.

Table 17: The ablation study of the effectiveness of multi-task training.

Task	Model	Objective Evaluation		Subjective Evaluation	
		Metrics	Results	Metrics	Results
Text-to-Speech	UniAudio (Single)	SIM(↑) / WER(↓)	0.64 / 2.4	MOS(↑)	3.77±0.06 / 3.46±0.10
	UniAudio		**0.71 / 2.0**	/ SMOS(↑)	**3.81±0.07 / 3.56±0.10**
Voice Conversion	UniAudio (Single)	SIM(↑) / WER(↓)	0.84 / 5.4	MOS(↑)	3.45±0.07 / 3.44±0.07
	UniAudio		**0.87 / 4.8**	/ SMOS(↑)	**3.54±0.07 / 3.56±0.07**
Speech Enhancement	UniAudio (Single)	PESQ(↑)	2.35 / 2.30 / 3.45	MOS(↑)	3.65±0.08
	UniAudio	/ VISQOL(↑) / DNSMOS(↑)	**2.63 / 2.44 / 3.66**		**3.68±0.07**
Target Speaker Extraction	UniAudio (Single)	PESQ(↑)	**1.97** / 1.61 / 3.93	MOS(↑)	3.58±0.08
	UniAudio	/ VISQOL(↑) / DNSMOS(↑)	1.88 / **1.68 / 3.96**		**3.72±0.06**
Singing Voice Synthesis	UniAudio (Single)	-	-	MOS(↑)	**4.14±0.07** / 4.02±0.02
	UniAudio			/ SMOS(↑)	4.08±0.04 / **4.04±0.05**
Text-to-Sound	UniAudio (Single)	FAD (↓) / KL (↓)	3.84 / 2.7	OVL (↑)	60.0±2.1 / 61.2±1.8
	UniAudio		**3.12 / 2.6**	/ REL (↑)	**61.9±1.9 / 66.1±1.5**
Text-to-Music	UniAudio (Single)	FAD (↓) / KL (↓)	5.24 / **1.8**	OVL (↑)	64.4±2.1 / 66.2±2.4
	UniAudio		**3.65** / 1.9	/ REL (↑)	**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**

C.2 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 3 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.

C.3 THE INFLUENCE OF DATA QUANTITY

In this part, we conduct experiments to explore the influence of data quantity, we give three settings: (1) using all of the data; (2) using $1/2$ training data for each task; (3) using $1/4$ training data for each task. We present the results in Figure 4. Based on the experimental results, this work claims that the data quantity is a key point to building a strong audio foundation model. In the future, we will explore to use of more unlabeled data to help improve the performance.

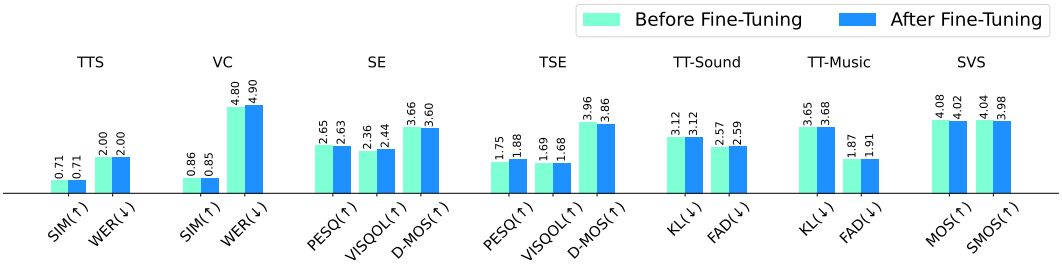


Figure 3: Performance comparison over 7 audio generation tasks before/after fine-tuning.

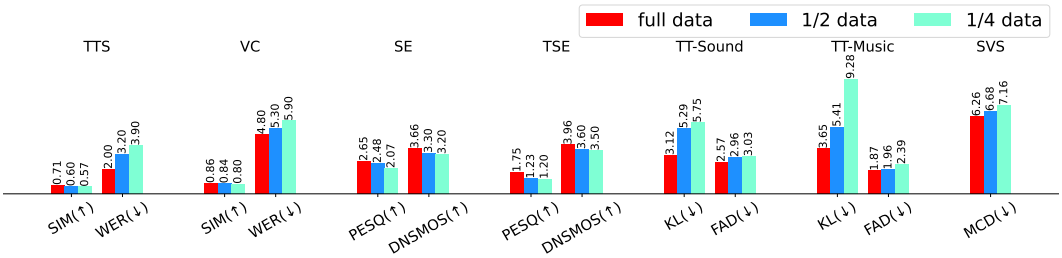


Figure 4: Performance comparison over different data quantity.

D WHY UNIAUDIO CAN WORK WELL?

From the previous discussions, we can see that the universal modeling strategy brings improvement for different tasks. In this part, we try to give some potential explanations.

(1) **Deterministic latent space**: we formulate different modalities into a deterministic latent space (fixed vocabulary) by tokenization. Different tokens can be seen as specific 'words', and we can use a next-token prediction strategy to train the model. Similar to GPT-series (Radford et al., 2018; 2019), such strategy creates the opportunity for the model to learn the intrinsic properties of audio and the interrelationship between audio and other modalities.

(2) **Shared information between different types of audio**: Although multiple types of audio (speech, sounds, music, and singing) present significant differences in the time domain or frequency domain, neural audio codec models effectively capture their shared information (rethinking the working principle of neural codecs, which similar information will be allocated the same token id). Due to the shared information that exists in different types of audio, multi-task training can be seen as increasing training data for each task.

(3) **Data augmentation perspective**: We speculate that multi-task training can be viewed as data augmentation for some tasks. Considering the TTS and VC task's definition:

```
TTS: <phoneme_sequence> <prompt> <audio_sequence>
VC: <semantic_token> <prompt> <audio_sequence>
```

We can see that the difference in task formulation for TTS and VC is that they use different ways to denote the phonetic information. In essence, they carry the same phonetic information. The difference is that semantic tokens include the duration information. Thus we can view the phoneme sequence as a special semantic sequence that drops the duration information. Such dropping operation is widely used as a data augmentation strategy (Park et al., 2019).

E 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 with the Encodec model, the difference includes: (1) we replace

Table 18: Performance comparison between encodec and our universal neural codec. FPS: frame per second; TPS: token per second. Perceptual evaluation of speech quality (PESQ↑); Short Term Objective Intelligibility (STOI↑).

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	n_q	FPS	TPS	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI
Encodec	8	75	600	2.18	0.79	2.23	0.48	1.86	0.57	1.95	0.76	2.05	0.65
Ours	3	50	150	2.96	0.85	2.42	0.49	1.99	0.57	3.13	0.85	2.62	0.69
Ours	4	50	200	3.11	0.86	2.5	0.51	2.08	0.59	3.27	0.86	2.73	0.71
Ours	8	50	400	**3.36**	**0.88**	**2.67**	**0.54**	**2.31**	**0.65**	**3.49**	**0.89**	**2.95**	**0.74**

the multi-scale STFT-based (MS-STFT) discriminator as our multi-scale Mel-based discriminator. (2) We rewrite the vector quantization implementation [17] based on Encodec's open-source version [18], making it more suitable for DDP training. Figure 5 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 (the SOTA sounds or music classification systems are based on the log-mel-spectrogram in the literature.). 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 [19]. Specifically, we set the hidden_dim as {64, 128, 256, 512, 512, 512} and the hop length as {32, 64, 128, 256, 512, 1024}. We train the neural audio codec model based on the Librilight and AudioSet datasets. Table 18 demonstrates that the neural codec model adopted in this work outperforms prior Encodec (Défossez et al., 2022).

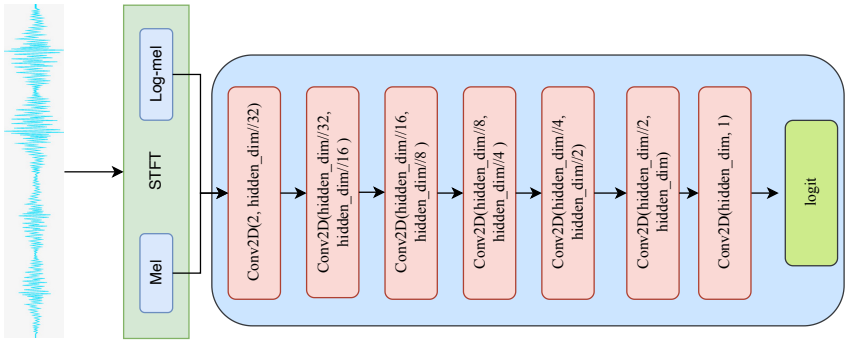


Figure 5: The overview of a single Mel-based discriminator. In practice, we will use multiple discriminators by setting different hop lengths and hidden dimensions.

F Subjective Evaluation

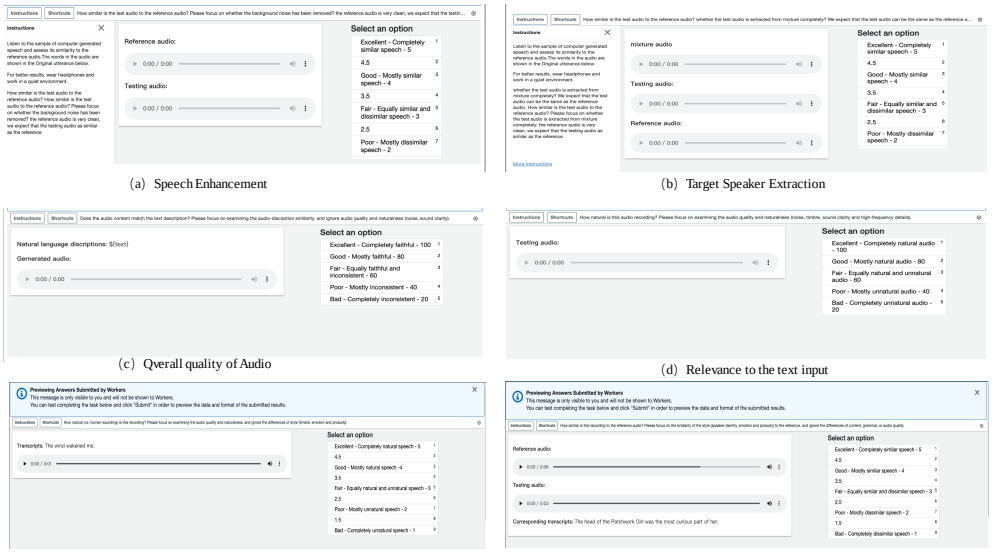
For TTS and VC tasks, we focus on speech quality (QMOS) and speaker similarity (SMOS). The details are as follows. For speech quality evaluation, we conduct the MOS (mean opinion score) tests and explicitly ask the raters to *focus on examining the audio quality and naturalness, and ignore the differences 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.

[17] Please refer to our source code to find the details.

[18] https://github.com/facebookresearch/encodec/blob/main/encodec/quantization/core_vq.py

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



(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 6: Screenshots of subjective evaluations.

For SE and TSE tasks, we write explicit instructions to ask the rater to assess the generated speech. Refer to Figure 6 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 6. We paid about $500 on participant compensation. A small subset of speech samples used in the test is available at `https://uniaudio666.github.io/demo_UniAudio/`.