Metis: A Foundation Speech Generation Model with Masked Generative Pre-training

Yuancheng Wang, Jiachen Zheng, Junan Zhang, Xueyao Zhang, Huan Liao, Zhizheng Wu
The Chinese University of Hong Kong, Shenzhen
yuanchengwang@link.cuhk.edu.cn, wuzhizheng@cuhk.edu.cn

Abstract

We introduce ***Metis***, a foundation model for unified speech generation. Unlike previous task-specific or multi-task models, Metis follows a pre-training and fine-tuning paradigm. It is pre-trained on large-scale unlabeled speech data using masked generative modeling and then fine-tuned to adapt to diverse speech generation tasks. Specifically, 1) Metis utilizes two discrete speech representations: SSL tokens derived from speech self-supervised learning (SSL) features, and acoustic tokens directly quantized from waveforms. 2) Metis performs masked generative pre-training on SSL tokens, utilizing 300K hours of diverse speech data, without any additional condition. 3) Through fine-tuning with task-specific conditions, Metis achieves efficient adaptation to various speech generation tasks while supporting multimodal input, even when using limited data and trainable parameters. Experiments demonstrate that Metis can serve as a foundation model for unified speech generation: Metis outperforms state-of-the-art task-specific or multi-task systems across five speech generation tasks, including zero-shot text-to-speech, voice conversion, target speaker extraction, speech enhancement, and lip-to-speech, even with fewer than 20M trainable parameters or 300 times less training data. Audio samples are are available at https://metis-demo.github.io/. We release the code and model checkpoints at https://github.com/open-mmlab/Amphion.

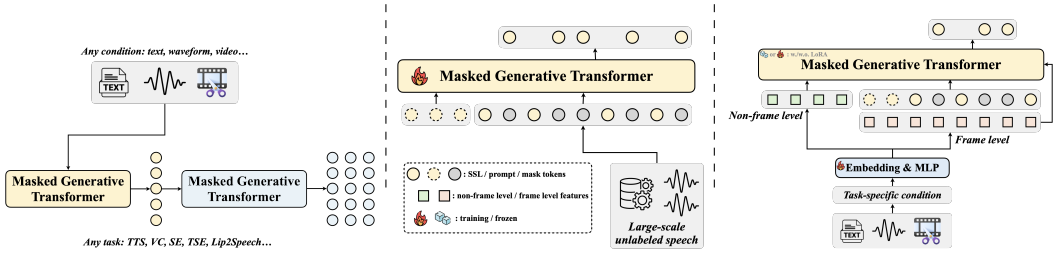
1 Introduction

Advancing a unified framework capable of addressing diverse tasks is a central research objective within the domain of artificial intelligence. In natural language processing [1, 2, 3] and computer vision [4, 5, 6], foundation models leveraging large-scale self-supervised pre-training have demonstrated remarkable adaptability across a wide spectrum of downstream tasks. However, in the domain of speech generation, this potential remains underexplored. A unified speech can integrate various speech generation technologies, such as text-to-speech [7, 8, 9, 10, 11, 12], voice conversion [13, 14, 15, 16], and speech enhancement [17, 18, 19]. This integration reduces redundant development, facilitates broader applications across diverse domains, and enhances the efficiency of human-machine interaction.

Previous speech generation models are mostly expert models that require extensive task-specific designs [7, 20, 21, 22, 23, 24, 25]. UniAudio [26] and SpeechX [27] are pioneering works that attempt to use autoregressive language models for multiple speech generation tasks. However, *they require a large amount of paired training data for each task and face challenges in extending to new tasks based on pre-trained models*. In addition, the autoregressive approach leads to suboptimal results in certain tasks and is relatively inefficient.

In this paper, we address the research question of ***how to design a unified speech generation framework that leverages large-scale unlabeled speech data for pre-training and efficiently adapts to diverse speech generation tasks through fine-tuning.*** Inspired by previous two-stage speech

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) Two-stage generation. (b) Pre-training. (c) Fine-tuning.

Figure 1: An illustration of Metis. (a) provides an overview of the two-stage speech generation framework, which consists of task-specific (yellow block) and task-independent (light blue block) processes. In this work, we focus on developing a pre-training model for the first stage, as illustrated in (b). (c) demonstrates the fine-tuning process, where the pre-trained model is adapted to specific tasks.

generation models [28, 29, 30, 31, 32], we revisit the speech generation process and observe a common structure underlying most tasks: generating intermediate representations from task-specific conditions and synthesizing acoustic representations from these intermediates. The intermediate representations are typically discrete units quantized from self-supervised speech features [33, 34, 35, 36, 37] that encode semantic and prosodic information, while acoustic representations are easily used to reconstruct high-quality waveforms.

Building on this observation, we propose a unified framework that modularizes speech generation tasks as a two-step process, as illustrated in Figure 1a: 1) **Task-Specific Process**: Generating SSL tokens conditioned on task-specific conditions (e.g., text for text-to-speech, noisy speech for speech enhancement, or visual features for lip-to-speech). 2) **Task-Independent Process**: Generating acoustic representations from the SSL tokens. This formulation allows diverse tasks to be addressed by varying input conditions while maintaining a consistent generation mechanism. Previous studies [12, 28, 30, 31, 32, 38, 39, 40] have established that the second stage of acoustic refinement can be achieved effectively using fully self-supervised diffusion models [31, 32, 38, 39, 40] or language models [12, 28, 30]. Therefore, *our objective is simplified to designing a generative pre-training mechanism for the first stage, enabling the pre-trained model to effectively generalize across various speech generation tasks*. In this work, we adopt masked generative models [41] for unconditional pre-training. Masked generative models are trained using a straightforward masked token prediction approach and employ an iterative sampling strategy to generate outputs from a fully masked input. We provide more details about masked generative models in Section 3.1. Once pre-trained, the model can be fine-tuned with limited data and parameters by incorporating task-specific conditions as additional inputs to adapt to various speech generation tasks.

Based on the above discussion, we propose ***Metis***, a foundation model for unified speech generation. Specifically, Metis has the following key features:

Masked Generative Pre-Training: Metis performs masked generative pre-training on SSL tokens using large-scale unlabeled speech data without any task-specific condition, as illustrated in Figure 1b. This pre-training phase establishes a strong foundation, allowing Metis to efficiently adapt to various downstream tasks through fine-tuning with minimal task-specific data.

Efficient Adaption to Various Speech Generation Tasks: Metis can be efficiently adapted to a wide range of speech generation tasks, including zero-shot text-to-speech, voice conversion, speech enhancement, target speaker extraction, and lip-to-speech, as illustrated in Figure 1c. The fine-tuned models achieve state-of-the-art results, even when using fewer than 20M trainable parameters or significantly less training data.

Support for Multimodal Conditional Inputs: Metis supports multimodal conditional inputs during the fine-tuning phase, including text, audio, and video. This capability solidifies Metis as a foundation model for supporting various speech generation tasks. We also explore multi-task fine-tuning, showing that the pre-trained model can be efficiently adapted into a powerful multi-task model with minimal modification, enabling novel applications such as text-guided target speaker extraction with task combinations.

2 Related Work

Masked Generative Models for Speech Masked generative models (MGMs) are a family of generative models that typically employ non-autoregressive transformers [42]. These models have achieved significant success, demonstrating performance comparable to or even surpassing autoregressive and diffusion models in image [41, 43, 44] and video [45, 46] generation, while offering a better balance between quality and speed. In the speech domain, SoundStorm [30] uses the semantic tokens from AudioLM [28] and employs MGMs to generate acoustic tokens from a neural audio codec [47], enabling applications like TTS and voice conversion. NaturalSpeech 3 [10] adopts MGMs to generate disentangled speech tokens. MaskGCT [12] further leverages MGMs for zero-shot generation, eliminating the need for explicit text-speech alignment or phone-level duration prediction in non-autoregressive TTS models. MaskSR [48] applies MGMs to speech enhancement tasks. In this work, we propose a unified speech generation framework based on MGMs.

Unified Speech Generation Developing a unified framework capable of handling various tasks is a key research objective in artificial intelligence. In the field of speech generation, UniAudio [26] employs an LLM for next-token prediction to generate multiple types of audio. Similarly, SpeechX [27] leverages an LLM for unified zero-shot tasks such as TTS, noise suppression, and target speaker extraction. Both models achieve this by concatenating the condition and target speech tokens, followed by AR modeling. However, these models require large amounts of paired training data for each task, failing to leverage the vast amount of unlabeled speech data effectively. VoiceBox [11] employs flow matching to unify tasks such as zero-shot TTS, speech editing, and speech enhancement. However, it has notable limitations, such as requiring text and clean speech as references for speech enhancement and relying on phone durations during training for zero-shot TTS. Its successor, AudioBox [49], extends VoiceBox to unified audio generation with natural language prompt control. Our work is partly inspired by SpeechFlow [50], which uses flow matching [51] to learn infilling during pre-training and fine-tunes with task-specific conditions for various speech generation tasks, such as zero-shot TTS and speech separation. However, it is limited to frame-level conditions, such as requiring a frame-level phoneme sequence for TTS. Additionally, predicting mel-spectrograms directly during pre-training may be suboptimal due to the need to predict extensive acoustic details.

Speech discrete representation is also highly relevant to our work, and we have included this part in Appendix F.

3 Method

3.1 Background: Masked Generative Models

In this section, we provide a brief introduction to masked generative models [12, 41, 46]. Consider a discrete sequence $\boldsymbol{x} = [y_1, y_2, \ldots, y_n]$, where n denotes the length of the sequence. We define $\boldsymbol{x}_t = \boldsymbol{x} \odot \boldsymbol{m}_t$ as the operation of masking a subset of tokens in $\boldsymbol{x}$ using the corresponding binary mask $\boldsymbol{m}_t = [m_{t,1}, m_{t,2}, \ldots, m_{t,n}]$. Specifically, this operation involves replacing x_i with a special `[MASK]` token if $m_{t,i} = 1$, and otherwise leaving x_i unmasked if $m_{t,i} = 0$. Here, each $m_{t,i}$ is independently and identically distributed according to a Bernoulli distribution with parameter $\gamma(t)$, where $\gamma(t) \in (0, 1]$ represents a mask schedule function (for example, $\gamma(t) = \sin(\frac{\pi t}{2T}), t \in (0, T]$). We denote $\boldsymbol{x} = \boldsymbol{x}_0$. The masked generative models are trained to predict the complete sequence (masked tokens) based on the observed tokens (unmasked tokens) and the condition $\boldsymbol{c}$, which can be modeled as $p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_t, \boldsymbol{c})$, and the model parameters θ are trained to optimize the sum of the marginal cross-entropy for each unobserved token:

$$\mathcal{L}_{\text{mask}} = -\mathbb{E}_{\boldsymbol{x},t,\boldsymbol{m}_t} \sum_{i=1}^{n} m_{t,i} \cdot \log p_\theta(y_i \mid \boldsymbol{x}_t, \boldsymbol{c}) \tag{1}$$

Note that $\boldsymbol{c}$ may be empty, for example, during the unconditional pre-training stage of our model. At the inference stage, masked generative models generate tokens in parallel through iterative decoding. The process begins with a fully masked sequence $\boldsymbol{x}_T$. Assuming the total number of decoding steps is S, for each step j from 1 to S, we first sample $\hat{\boldsymbol{x}}_0$ from $p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_{T-(j-1)\cdot\frac{T}{S}}, \boldsymbol{c})$. Then we sample $\lfloor n \cdot \gamma(T - j \cdot \frac{T}{S}) \rfloor$ tokens based on the confidence score to remask, resulting in $\boldsymbol{x}_{T-j\cdot\frac{T}{S}}$, where n is the sequence length of $\boldsymbol{x}$. The confidence score for $\hat{y}_i$ in $\hat{\boldsymbol{x}}_0$ is assigned to $p_\theta(y_i \mid \boldsymbol{x}_{T-(j-1)\cdot\frac{T}{S}}, \boldsymbol{c})$

if $y_{T-(j-1)\cdot\frac{T}{S},i}$ is a `[MASK]` token; otherwise, we set the confidence score of $\hat{y}_i$ to 1, indicating that tokens already unmasked in $\boldsymbol{x}_{T-(j-1)\cdot\frac{T}{S}}$ will not be remasked. In particular, we choose $\lfloor n\cdot\gamma(T-j\cdot\frac{T}{S})\rfloor$ tokens with the lowest confidence scores to be masked. Note that the method for calculating the confidence score is not unique. For example, Lezama et al. [52] introduces Token-Critic, training a critic model to compute confidence scores, aiding the sampling process. Additionally, Xie et al. [44], Lezama et al. [52] suggest that masked generative modeling can be seen as a simplified version of discrete diffusion.

3.2 Overview of Metis

Figure 1 provides an overview of our system, which adopts a two-stage speech generation process for unified speech generation tasks. We first briefly introduce the distinct speech discrete representations used in the two stages in Section 3.3. Then, we present the pre-training (Section 3.4) and fine-tuning (Section 3.5) strategies for the first stage, which form the core of our work. Finally, we provide a brief overview of a unified acoustic decoder for all speech generation tasks in Section 3.6.

3.3 Discrete Representations for Two-Stage Generation

Metis employs two discrete speech representations for the two-stage generation, as illustrated in Figure 2. 1) **SSL tokens**: Derived from SSL features of large-scale speech self-supervised learning models [33, 34, 36, 37], SSL tokens encapsulate both semantic and prosodic information, making them well-suited for conditional generation. To minimize information loss, we employ a vector quantization (VQ) model [53, 54] to quantize SSL features into discrete tokens, following Wang et al. [12]. 2) **Acoustic tokens**: Directly obtained from the waveform via vector quantization. The goal is to preserve all the information from the speech to reconstruct a high-quality waveform. We show more details in Appendix A.

3.4 Masked Generative Pre-training with SSL Tokens

Based on the previous discussion, most speech generation tasks can be generalized into two stages: *conditions to SSL tokens* and *SSL tokens to acoustic tokens*. The primary distinction among tasks lies in the nature of the conditions. To address this, we propose a unified pre-trained model for the first stage, which can be adapted to various tasks.

We use an unconditional masked generative model on SSL tokens for pre-training. Specifically, we randomly mask tokens in the SSL token sequence $\boldsymbol{x}^{ssl}$ using the strategy outlined in Section 3.1 and predict the masked tokens. We introduce a prompt sequence with the probability p to further enhance the in-context learning ability of the model. With this probability, a prefix sequence $\boldsymbol{x}^{ssl}_{prompt}$ from the SSL token sequence is used as a prompt and remains unmasked. This mechanism enables the model to leverage prompt information, thereby improving its adaptability to downstream tasks requiring prompts, such as zero-shot TTS and target speaker extraction. The pre-training objective is to model $p_\theta(\boldsymbol{x}^{ssl}_0 \mid \boldsymbol{x}^{ssl}_t, \boldsymbol{x}^{ssl}_{prompt})$.

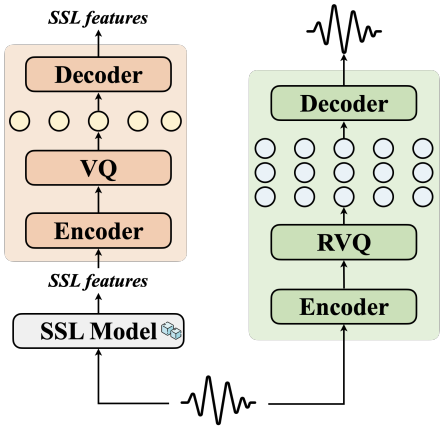


Figure 2: Two discrete speech representations for the two-stage speech generation: SSL tokens (left) and acoustic tokens (right).

Our design is motivated by the observation that models trained on extensive data can recover masked SSL tokens from prompts and unmasked tokens, even without task-specific conditions, as shown in other domains [2, 4]. Empirically, the pre-trained model can generate speech that mimics the prosodic style and timbre of a prompt. However, the generated speech often lacks intelligibility, producing random word concatenations due to the absence of semantic guidance. This highlights the need for task-specific conditions, which can be efficiently incorporated through fine-tuning to adapt the model to various speech generation tasks.

3.5 Efficient Adaptation to Various Generation Tasks

Now, we describe how to efficiently adapt the pre-trained model to various speech generation tasks. We first categorize the conditions for different speech generation tasks into two types: non-frame-level conditions and frame-level conditions. For the former, such as TTS, the condition is a phoneme sequence or a text token sequence. In this case, the model needs to implicitly learn the alignment between the condition and the SSL token sequence. For the latter, such as voice conversion or speech enhancement, the conditions (e.g., the source speech voice conversion or the noisy speech for speech enhancement) can be aligned with the target SSL token sequence at the frame level. Based on these distinctions, during fine-tuning, for non-frame-level conditions, we simply concatenate the condition with the input sequence along the time dimension. For frame-level conditions, we apply a simple interpolation to align the condition with the input sequence in the time dimension and then pass it through an MLP-based adapter before adding it to the input. Then, the fine-tuned model is trained to learn $p_\theta(\boldsymbol{x}_0^{ssl} \mid \boldsymbol{x}_t^{ssl}, \boldsymbol{x}_{prompt}^{ssl}, \boldsymbol{c})$, where $\boldsymbol{c}$ is the task-specific condition. Our experiments demonstrate that the pre-trained model can efficiently adapt to various tasks and achieve remarkable results with minimal data. Additionally, we explored the use of Low-Rank Adaptation (LoRA) [55], which allows fine-tuning with a small number of trainable parameters. More details are provided in Section 4.

3.6 Masked Generative Acoustic Decoder

We train an SSL-to-acoustic model based on masked generative modeling, which serves as a unified acoustic decoder for all speech generation tasks. The model is trained to recover masked tokens from a masked acoustic token sequence $\boldsymbol{x_t}^a$, conditioned on SSL tokens $\boldsymbol{x}_{ssl}$ and prompt acoustic tokens $\boldsymbol{x}_{prompt}^a$. This can be formulated as $p_\theta(\boldsymbol{x}_0^a \mid \boldsymbol{x}_t^a, \boldsymbol{x}_{prompt}^a, \boldsymbol{x}_{ssl})$. During training, we randomly select a layer for masking in the multi-layered acoustic tokens, while the lower-layer tokens remain unmasked and serve as conditional inputs to the model, following [30]. During inference, we generate acoustic tokens layer by layer. Finally, we generate the speech by first predicting SSL tokens and then generating acoustic tokens.

4 Experiments and Results

4.1 Setup

Model Architecture We adopt the same model architecture as described in Wang et al. [12], except removing the text embedding during the pre-training phase. The model follows the standard Llama-style architecture [42, 56], but replaces causal attention with bidirectional attention.

Dataset We use a dataset consisting of 300K hours of speech for pre-training, including 100K hours from the Emilia [57] dataset and an additional 200K hours self-collected through the Emilia pipeline, to train our pre-trained model. The dataset contains a diverse range of in-the-wild multilingual speech data, including 87K hours in Chinese, 185K hours in English, 7K hours in German, 8K hours in French, 2.5K hours in Japanese, and 7.5K hours in Korean. The dataset used for fine-tuning is sampled from the pre-trained dataset.

Training We pre-train our model on 8 GPUs for a total of 1200K steps. We use the AdamW [58] optimizer with a learning rate of 1e-4 and 32K warmup steps. We employ a dynamic batch size, where each batch contains 10K tokens (200 seconds) per GPU. During training, we randomly select a prefix of the sequence as a prompt that is not masked with a probability $p = 0.8$. The length of the prompt is uniformly sampled from the range $[0\%, 40\%]$ of the total sequence length.

Inference We follow the inference strategy outlined in Wang et al. [12], with task-specific step adjustments for the fine-tuned model to generate SSL tokens. Additionally, we use classifier-free guidance [59] for tasks that involve prompts.

Evaluation Metrics We use multiple evaluation metrics to assess different aspects of the generated speech, including similarity (SIM), intelligibility (WER), and audio quality (DNSMOS, SIG, BAK, OVRL, NISQA). For the subjective metrics, quality mean opinion score (CMOS) and similarity mean

opinion score (SMOS) are used to evaluate speech quality and similarity. The results of the subjective evaluation are presented in Appendix D.3. Details about these metrics are provided in Appendix C.

4.2 Results on Different Speech Generation Tasks

We present the fine-tuning results of Metis on various speech generation tasks. These tasks include zero-shot TTS (without frame-level phoneme condition or duration prediction), voice conversion, speech enhancement, target speaker extraction, and the multimodal task of lip-to-speech.

4.2.1 Zero-Shot TTS

Implementation Details For zero-shot TTS, we fine-tune the pre-trained model on three different datasets: 1K hours and 10K hours randomly sampled from the Emilia dataset, and the LibriTTS [62] dataset, which contains 0.58K hours of English speech data. We use two fine-tuning methods: full-scale fine-tuning and LoRA fine-tuning. For LoRA fine-tuning, we set the rank $r = 32$, resulting in only **32M trainable parameters**, including the newly added text embedding module. All fine-tuned models are trained using 4 GPUs for 200K steps.

Table 1: Results on zero-shot TTS task.

Model	Training Data	WER(↓)	SIM(↑)	DNSMOS(↑)
		SeedTTS test-en		
Ground Truth	-	2.14	0.73	3.53
VALL-E [8]	45K EN	6.13	0.43	3.39
VoiceCraft [60]	9K EN	7.55	0.47	3.37
CosyVoice [32]	170K Multi.	4.08	0.64	3.64
XTTS-v2 [61]	27K Multi.	3.25	0.46	3.45
MaskGCT [12]	100K Multi.	2.47	0.72	3.51
Metis-TTS LoRA 32	1K Multi.	4.78	0.72	3.47
Metis-TTS LoRA 32	10K Multi.	**4.55**	0.72	3.45
Metis-TTS LoRA 32	0.58K[1] EN.	4.63	0.70	3.51
Metis-TTS fine-tune	0.58K[1] EN.	**3.04**	0.68	3.47
Metis-TTS fine-tune	1K Multi.	3.86	0.71	3.46
Metis-TTS fine-tune	10K Multi.	**2.28**	0.72	3.47
Metis-TTS w.o. pre-train	10K Multi.	4.91	0.69	3.42
		SeedTTS test-zh		
Ground Truth	-	1.25	0.75	3.51
CosyVoice [32]	170K Multi.	4.09	0.75	3.71
XTTS-v2 [61]	27K Multi.	2.88	0.63	3.44
MaskGCT [12]	100K Multi.	2.18	0.77	3.58
Metis-TTS LoRA 32	1K Multi.	5.21	0.77	3.57
Metis-TTS LoRA 32	10K Multi.	**4.48**	0.77	3.55
Metis-TTS fine-tune	1K Multi.	4.23	0.77	3.54
Metis-TTS fine-tune	10K Multi.	**2.30**	0.77	3.55
Metis-TTS w.o. pre-train	10K Multi.	4.98	0.73	3.51

[1] This version of the model is trained on LibriTTS [62].

Evaluation and Baseline We evaluate our zero-shot TTS models using three test sets: 1) SeedTTS *test-en*, a test set introduced in Seed-TTS [31] comprising 1,000 samples extracted from English public corpora, including the Common Voice dataset [63]. 2) SeedTTS *test-zh*, a test set introduced in Seed-TTS consisting of 2,000 samples extracted from Chinese public corpora, including the DiDiSpeech dataset [64]. 3) LibriSpeech *test-clean* [65], a widely used test set for TTS. We compare our models with several recent zero-shot TTS models, including AR-based models: VALL-E [8], VoiceCraft [60], XTTS-v2 [61], and CosyVoice [32]; NAR-based models: VoiceBox [11], and MaskGCT [12]. Specifically, MaskGCT can be seen as Metis without pre-training. For VoiceBox and NaturalSpeech 3, we only compare with them on LibriSpeech *test-clean* since they are not open-source. See more details about these baselines in Appendix B.1.

Result The results are presented in Table 1, with additional results for LibriSpeech *test-clean* set provided in Appendix D.1. The table shows that Metis-TTS LoRA 32, fine-tuned with only 1K hours of data and 32M trainable parameters, achieves performance on metrics such as WER, SIM, and DNSMOS that is comparable to or exceeds that of some baselines trained on datasets 10 to 100 times larger. Furthermore, when scaled with full fine-tuning on 10K hours of data, Metis demonstrates an improved WER on SeedTTS *test-en*, outperforming MaskGCT, which was trained on 100K hours of data, while maintaining comparable WER performance on SeedTTS *test-zh*. When fine-tuned with only 580 hours of LibriTTS data, Metis achieves competitive WER and SIM performance, surpassing all baselines except MaskGCT on SeedTTS *test-en*. We also compare with models that are not pre-trained but trained for the same number of steps. The results show that the fine-tuned models converge faster and achieve better performance. In addition, all fine-tuned models exhibit excellent SIM and DNSMOS, indicating high speech similarity and quality.

4.2.2 Voice Conversion

Implementation Details Previous voice conversion systems typically extract timbre-independent features from input signals using methods such as information bottlenecks [14, 66, 67], timbre perturbation [15, 68, 69], or specialized loss functions [10, 67]. However, these approaches often introduce complexity, risk the loss of semantic information, and may still inadvertently retain timbre-

related details [70, 71]. In this work, we use a simpler yet effective solution inspired by Anastassiou et al. [31], Neekhara et al. [72]. Specifically, we leverage a lightweight voice conversion model[1] [73] to perform real-time voice conversion on the target speech using a randomly sampled prompt speech, thereby achieving timbre perturbation. The perturbed speech features are then used as input to predict the target speech based on the prompt. Specifically, we employ the `w2v-bert-2.0` features of the perturbed speech as conditioning inputs and randomly extract a prefix of the target speech as the prompt. Our experiments demonstrate that with minimal data and training steps, our pre-trained model can effectively and rapidly adapt to the voice conversion task. We use both full-scale fine-tuning and LoRA fine-tuning. For LoRA fine-tuning, we set the rank $r = 16$, resulting in only **9M trainable parameters** excluding those of the MLP-based adapter. Additionally, we observe that for the voice conversion task, our pre-trained model converges on a single A100 GPU after just 10K steps of LoRA fine-tuning and **5K steps** of full-scale fine-tuning, using randomly sampled **0.4K hours** of training data.

Evaluation and Baseline We evaluate our models on the VCTK [75] dataset. we randomly select 200 samples from the dataset as source speech, and for each sample, we randomly select another sample from the same speaker as the prompt speech. We also evaluate the models on other test sets, the results are shown in Appendix D.2. We compare our models with several recent voice conversion models, including HireSpeech++ [21], LM-VC [68], UniAudio [26], and Vevo [74]. More details about these baselines can be found in Appendix B.2.

Table 2: Results on voice conversion task.

Model	Training Data	WER(↓)	SIM(↑)	DNSMOS(↑)	NISQA(↑)
		VCTK			
HierSpeech++ [21]	2.8K	4.87	0.38	3.40	3.79
LM-VC [68]	1.4K	8.35	0.29	3.46	3.93
UniAudio [26]	60K	9.00	0.25	3.47	4.28
Vevo [74]	60K	**3.48**	0.38	3.47	4.30
Metis-VC LoRA 16, cfg = 0.0	0.4K	4.49	0.50	3.48	4.46
Metis-VC LoRA 16, cfg = 2.0	0.4K	7.90	**0.55**	3.46	4.42
Metis-VC fine-tune, cfg = 0.0	0.4K	6.65	0.48	**3.49**	**4.47**

[1] For LoRA 16, we train on one A100 GPU for 10K steps. For fine-tuning, we train on one A100 GPU for 5K steps.
[2] The best and the second best result is shown in **bold** and by underlined.

Result The results are presented in Table 2. The table shows 1) For similarity, our model outperforms all baseline systems, achieving a SIM score of 0.55, significantly higher than the best baseline result of 0.38. 2) For WER, Metis-VC LoRA 16, cfg = 0.0 achieves a competitive result of 4.49, second only to Vevo. 3) For audio quality, as measured by DNSMOS and NISQA, our model shows competitive performance, with slight improvements over most baseline systems. In addition, our model requires only a significantly small amount of data to fine-tune for voice conversion.

4.2.3 Target Speaker Extraction

Implementation Details For target speaker extraction, we randomly sample 10K hours of data from the pre-training dataset to create the fine-tuning training set without any filtering. During training, samples are dynamically mixed to simulate multi-speaker speech data. Specifically, a random prefix from a sample is extracted as the prompt, and the remaining portion is mixed with another randomly selected sample by directly adding their waveforms. The `w2v-bert-2.0` features of the mixed speech are then used as conditions. We use two fine-tuning methods: full- scale fine-tuning and LoRA fine-tuning. For LoRA fine-tuning, we explore different LoRA ranks $r = 4, 16, 32$, resulting in LoRA modules with trainable parameters of 2M, 9M, and 18M, respectively.

Table 3: Results on target speaker extraction task.

Model	WER(↓)	SIG(↑)	BAK(↑)	OVRL(↑)	NISQA(↑)	SIM(↑)
			LibriMix test			
Ground Truth	4.27	3.62	4.03	3.32	4.11	0.76
UniAudio [26]	20.08	3.64	**4.15**	3.33	4.32	0.66
VoiceFilter [22]	20.10	3.27	3.77	2.91	2.97	0.68
WeSep [23]	**6.19**	3.56	3.93	3.23	4.04	0.73
TSELM [76]	9.20	3.55	4.08	3.29	4.03	0.27
Metis-TSE LoRA 4	13.57	**3.66**	4.02	3.34	4.40	**0.75**
Metis-TSE LoRA 16	12.52	**3.66**	4.02	**3.35**	**4.41**	**0.75**
Metis-TSE LoRA 32	9.65	**3.66**	4.02	**3.35**	4.38	**0.75**
Metis-TSE fine-tune	6.31	3.65	4.02	3.34	4.36	0.74
			EmiliaMix test			
Ground Truth	0.00	3.57	4.01	3.27	3.83	0.86
UniAudio [26]	32.51	3.55	4.03	3.20	4.01	0.55
VoiceFilter [22]	24.10	3.21	3.64	2.78	2.15	0.71
WeSep [23]	**5.58**	3.50	3.85	3.12	3.86	**0.81**
TSELM [76]	52.12	3.48	**4.05**	3.20	3.87	0.26
Metis-TSE LoRA 4	11.55	3.59	3.84	**3.21**	4.04	0.77
Metis-TSE LoRA 16	10.48	**3.60**	3.85	**3.21**	**4.06**	0.77
Metis-TSE LoRA 32	8.72	**3.60**	3.86	**3.21**	4.05	0.78
Metis-TSE fine-tune	6.87	**3.60**	3.85	**3.21**	**4.06**	0.78

[1] The best and the second best result is shown in **bold** and by underlined.
[2] For LibriMix, we use the original text for computing WER. For EmiliaMix, we use the ASR-transcribed text of the ground truth speech for WER calculation.

Evaluation and Baseline We evaluate our models on the LibriMix [77] test set. LibriMix is used exclusively for evaluation and is not included in the training set, thus highlighting the generalization

[1] https://github.com/myshell-ai/OpenVoice

capabilities of our models. In addition, we construct a test set of 1K samples from the Emilia dataset, which features greater diversity in both speakers and acoustic environments. We denote it by EmiliaMix test. We compare our models with several open-source expert models for target speaker extraction: WeSep [23], VoiceFilter [22], and TSELM [76]. We also compare with UniAudio [26]. See more details about these baselines in Appendix B.3.

Result The results are presented in Table 3. The Table shows: 1) All fine-tuned models exhibit similar performance on audio quality metrics, while the WER decreases as the number of trainable parameters increases. 2) Our models demonstrate significant improvements across two test sets in audio quality metrics, DNSMOS and NISQA, compared to the baselines. Metis-TSE LoRA 16 achieves state-of-the-art NISQA scores of 4.41 on LibriMix test and 4.46 on EmiliaMix test, surpassing the ground truth scores of 4.11 and 3.83, respectively. 3) Metis-TSE fine-tune achieves a competitive WER, although it is slightly lower than WeSep.

4.2.4 Speech Enhancement

Implementation Details For speech enhancement, we simulate noisy speech following previous works [24, 48, 78, 79]. We utilize noise datasets such as WHAM! [80] and DEMAND [81], along with room impulse response (RIR) datasets [82], OpenSLR26 and OpenSLR28, in accordance with the 2020 DNS-Challenge[2]. For clean speech data, we randomly sample 10K hours from our pre-trained dataset without any filtering. Inspired by prior works [48, 78], we simulate speech degradation by probabilistically applying various distortions to the audio signals. These include adding noise within an SNR range of -5 to 20 dB with probability $p = 0.9$, introducing reverberation with $p = 0.35$, and limiting the speech signal bandwidth (randomly selected from 2 kHz, 4 kHz, or 8 kHz) with a probability of with $p = 0.25$. The `w2v-bert-2.0` features extracted from the degraded speech serve as input conditions for our models. We fine-tune the models using both full-scale fine-tuning and LoRA fine-tuning. For LoRA fine-tuning, we also experiment with $r = 4, 16, 32$, corresponding to LoRA modules with trainable parameter counts of 2M, 9M, and 18M, respectively.

Evaluation and Baseline We evaluate our models using the 2020 DNS-Challenge [83] test sets, which consist of three categories: 1) synthetic data with reverb, 2) synthetic data without reverb, and 3) real recordings. For comparison, we downsample the output of our models to 16 kHz. We compare our models with several recent speech enhancement models, including TF-GridNet [24], VoiceFixer [78], SELM [79] and MaskSR [48]. Notably, MaskSR is a model that also uses masked generative modeling but directly generates acoustic tokens for speech enhancement. See more details about these baselines in Appendix B.4.

Table 4: Results on speech enhancement task.

Model	SIG(↑)	BAK(↑)	OVRL(↑)	NISQA(↑)	SIM(↑)
DNS2020 with reverb					
TF-GridNet [24]	3.11	3.23	2.51	2.61	0.69
VoiceFixer [78]	3.43	4.02	3.13	3.82	0.91
SELM [79]	3.16	3.58	2.70	-	-
MaskSR [48]	3.53	4.07	3.25	-	-
Metis-SE LoRA 4	3.67	4.13	3.43	4.54	0.93
Metis-SE LoRA 16	3.67	4.13	3.43	**4.57**	**0.94**
Metis-SE LoRA 32	**3.68**	**4.14**	3.43	4.48	**0.94**
Metis-SE fine-tune	**3.68**	**4.14**	**3.44**	4.56	**0.94**
DNS2020 no reverb					
TF-GridNet [24]	3.54	4.05	3.27	4.35	0.68
VoiceFixer [78]	3.50	4.11	3.25	4.27	0.96
SELM [79]	3.51	4.10	3.26	-	-
MaskSR [48]	3.60	4.15	3.37	-	-
Metis-SE LoRA 4	3.65	4.15	3.43	**4.82**	**0.97**
Metis-SE LoRA 16	3.65	4.16	3.43	4.81	**0.97**
Metis-SE LoRA 32	**3.66**	**4.17**	**3.44**	4.77	**0.97**
Metis-SE fine-tune	3.64	**4.17**	3.43	4.76	**0.97**
DNS2020 Real Recording					
VoiceFixer [19]	3.31	3.93	3.00	3.66	-
SELM [79]	3.59	3.44	3.12	-	-
MaskSR [48]	3.43	4.03	3.14	-	-
Metis-SE LoRA 4	**3.60**	4.02	3.29	3.94	-
Metis-SE LoRA 16	**3.60**	**4.04**	**3.30**	**3.97**	-
Metis-SE LoRA 32	3.59	3.99	3.26	3.92	-
Metis-SE fine-tune	3.59	4.01	3.27	3.95	-

[1] The best and the second best result is shown in **bold** and by underlined.

Result The results are presented in Table 4. The table shows that our models achieve state-of-the-art performance across all benchmarks. 1) Metis-SE achieves state-of-the-art results across all three test sets with the highest SIG, BAK, OVRL, and NISQA scores, showing significant improvements over previous baselines. 2) Results among different fine-tuned versions are comparable, even for Metis-SE LoRA 4. 3) Notably, Metis-SE performs exceptionally well on the real recording dataset, further validating its practical applicability.

[2] https://github.com/microsoft/DNS-Challenge

Table 5: Results on lip-to-speech task.

Model	WER(↓)	DNSMOS(↑)	NISQA(↑)	SIM(↑)
LRS2				
Lip2Speech-Unit [25]	33.64	3.01	2.70	29.34
Metis-L2S fine-tune	**32.28**	**3.23**	**3.71**	**59.73**
LRS3				
Lip2Speech-Unit [25]	38.34	2.28	1.92	32.05
Metis-L2S fine-tune	**31.03**	**3.09**	**3.75**	**56.74**

Table 6: Results of Metis-Omni on target speaker extraction.

Model	WER(↓)	SIG(↑)	BAK(↑)	OVRL(↑)	NISQA(↑)	SIM(↑)
LibriMix test						
Ground Truth	4.27	3.62	4.03	3.32	4.11	0.76
WeSep [23]	6.19	3.56	3.93	3.23	4.04	0.73
TSELM [76]	9.20	3.55	**4.08**	3.29	4.03	0.27
Metis-TSE fine-tune	6.31	3.65	4.02	<u>3.34</u>	4.36	<u>0.74</u>
Metis-Omni fine-tune	<u>5.90</u>	<u>3.66</u>	<u>4.03</u>	**3.36**	**4.40**	**0.75**
Metis-Omni fine-tune, text-guided	**2.70**	**3.67**	<u>4.03</u>	**3.36**	<u>4.39</u>	**0.75**

[1] The best and the second best result is shown in **bold** and by <u>underlined</u>.

4.2.5 Lip-to-Speech

Implementation Details We use a combined data set comprising the training and sets of LRW[3], LRS2[4], and LRS3[5] as the training data. We use the same methodology in Ma et al. [84] to pre-process the videos and use the visual speech recognition encoder in Ma et al. [84] to extract visual features that served as the conditions and randomly extract a prefix of the target speech as the prompt.

Evaluation and Baseline We compare our models with Lip2Speech-Unit [85], an expert lip-to-speech model. We use the test sets of LRS2 and LRS3 to evaluate our models following previous works [25, 85]. Notably, over 60% of the samples in the two test sets have durations of less than 2 seconds. Utilizing excessively short audio as a prompt may degrade the sound quality of the inference results, while directly using part of the target audio as a prompt may lead to information leakage. We use speech randomly selected from SeedTTS *test-en* as prompts for each test case.

Result The results are presented in Table 5. The table shows that Metis-L2S outperforms the baseline Lip2Speech-Unit across all metrics. Specifically, the audio quality metrics show substantial improvements and the speaker similarity nearly doubles on both datasets (29.34 → 59.73 on LRS2, 32.05 → 56.74 on LRS3). The model also achieves a lower WER than the baseline.

4.3 Multi-Task Fine-Tuning

In addition to fine-tuning the pre-trained model separately for different tasks, we explore the potential of jointly fine-tuning it on multiple tasks, resulting in a multi-task model, which we refer to as Metis-Omni. For this study, we select four tasks: zero-shot TTS, voice conversion, target speaker extraction, and speech enhancement. Further details and experimental results are provided in Appendix E and Table 11.

We take target speaker extraction as an example to illustrate the effectiveness of Metis-Omni in enabling novel applications through task combinations. As shown in Table 6, Metis-Omni outperforms baseline methods in terms of WER. Furthermore, by integrating text-to-speech and target speaker extraction tasks, the text-guided version of Metis-Omni achieves a remarkable WER reduction to 2.70, demonstrating a substantial improvement over all other models. Notably, despite the model not being explicitly trained on text-guided target speaker extraction, it generalizes well to this novel setting, highlighting the advantage of our system in leveraging multimodal conditional inputs to enable flexible and efficient task adaptation.

5 Conclusion

In this work, we propose ***Metis***, a foundation model for unified speech generation that leverages large-scale unlabeled speech data for pre-training and can be effectively adapted to diverse speech generation tasks through fine-tuning. Our experiments demonstrate that Metis outperforms state-of-the-art task-specific and multi-task systems on zero-shot TTS, voice conversion, target speaker enhancement, speech enhancement, and lip-to-speech after fine-tuning, even with fewer than 20M trainable parameters or up to 300 times less training data for certain tasks while supporting multimodal conditional inputs. In addition, we propose Metis-Omni, a version of our pre-trained model fine-tuned on multiple tasks, which demonstrates further improvements.

[3] https://www.robots.ox.ac.uk/~vgg/data/lip_reading/lrw1.html
[4] https://www.robots.ox.ac.uk/~vgg/data/lip_reading/lrs2.html
[5] https://mmai.io/datasets/lip_reading/

6 Acknowledgment

The authors gratefully acknowledge the funding support from the National Natural Science Foundation of China (Grant No. 62376237), the Shenzhen Science and Technology Program (Grant No. ZDSYS20230626091302006), the Shenzhen Research Institute of Big Data (Internal Project Fund, Grant No. T00120230002), and the 2023 Shenzhen Stability Science Program. We would also like to thank the anonymous reviewers and the Area Chair for their insightful comments and valuable suggestions, which helped improve our paper.

References

[1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[2] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[5] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[7] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, 2020.

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

[9] Kai Shen, Zeqian Ju, Xu Tan, Yanqing Liu, Yichong Leng, Lei He, Tao Qin, Sheng Zhao, and Jiang Bian. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *arXiv preprint arXiv:2304.09116*, 2023.

[10] Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, et al. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. *arXiv preprint arXiv:2403.03100*, 2024.

[11] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36, 2024.

[12] Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Shunsi Zhang, and Zhizheng Wu. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *arXiv preprint arXiv:2409.00750*, 2024.

[13] Seyed Hamidreza Mohammadi and Alexander Kain. An overview of voice conversion systems. *Speech Communication*, 88:65–82, 2017.

[14] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *International Conference on Machine Learning*, pages 5210–5219. PMLR, 2019.

[15] Jingyi Li, Weiping Tu, and Li Xiao. Freevc: Towards high-quality text-free one-shot voice conversion. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[16] Ha-Yeong Choi, Sang-Hoon Lee, and Seong-Whan Lee. Diff-hiervc: Diffusion-based hierarchical voice conversion with robust pitch generation and masked prior for zero-shot speaker adaptation. *International Speech Communication Association*, pages 2283–2287, 2023.

[17] Santiago Pascual, Antonio Bonafonte, and Joan Serra. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

[18] Szu-Wei Fu, Cheng Yu, Tsun-An Hsieh, Peter Plantinga, Mirco Ravanelli, Xugang Lu, and Yu Tsao. Metricgan+: An improved version of metricgan for speech enhancement. *arXiv preprint arXiv:2104.03538*, 2021.

[19] Haohe Liu, Qiuqiang Kong, Qiao Tian, Yan Zhao, DeLiang Wang, Chuanzeng Huang, and Yuxuan Wang. Voicefixer: Toward general speech restoration with neural vocoder. *arXiv preprint arXiv:2109.13731*, 2021.

[20] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech: Fast, robust and controllable text to speech. *Advances in neural information processing systems*, 32, 2019.

[21] Sang-Hoon Lee, Ha-Yeong Choi, Seung-Bin Kim, and Seong-Whan Lee. Hierspeech++: Bridging the gap between semantic and acoustic representation of speech by hierarchical variational inference for zero-shot speech synthesis. *arXiv preprint arXiv:2311.12454*, 2023.

[22] Quan Wang, Hannah Muckenhirn, Kevin Wilson, Prashant Sridhar, Zelin Wu, John Hershey, Rif A Saurous, Ron J Weiss, Ye Jia, and Ignacio Lopez Moreno. Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking. *arXiv preprint arXiv:1810.04826*, 2018.

[23] Shuai Wang, Ke Zhang, Shaoxiong Lin, Junjie Li, Xuefei Wang, Meng Ge, Jianwei Yu, Yanmin Qian, and Haizhou Li. Wesep: A scalable and flexible toolkit towards generalizable target speaker extraction. *arXiv preprint arXiv:2409.15799*, 2024.

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

[25] Minsu Kim, Joanna Hong, and Yong Man Ro. Lip-to-speech synthesis in the wild with multi-task learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

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

[27] Xiaofei Wang, Manthan Thakker, Zhuo Chen, Naoyuki Kanda, Sefik Emre Eskimez, Sanyuan Chen, Min Tang, Shujie Liu, Jinyu Li, and Takuya Yoshioka. Speechx: Neural codec language model as a versatile speech transformer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[28] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2523–2533, 2023.

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

[30] Zalán Borsos, Matt Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, and Marco Tagliasacchi. Soundstorm: Efficient parallel audio generation. *arXiv preprint arXiv:2305.09636*, 2023.

[31] Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, et al. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*, 2024.

[32] Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, et al. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*, 2024.

[33] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 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.

[34] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE, 2021.

[35] Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, et al. Google usm: Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037*, 2023.

[36] Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. Self-supervised learning with random-projection quantizer for speech recognition. In *International Conference on Machine Learning*, pages 3915–3924. PMLR, 2022.

[37] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[38] Hao-Han Guo, Kun Liu, Fei-Yu Shen, Yi-Chen Wu, Feng-Long Xie, Kun Xie, and Kai-Tuo Xu. Fireredtts: A foundation text-to-speech framework for industry-level generative speech applications. *arXiv preprint arXiv:2409.03283*, 2024.

[39] Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[40] Haohe Liu, Xuenan Xu, Yi Yuan, Mengyue Wu, Wenwu Wang, and Mark D Plumbley. Semanticodec: An ultra low bitrate semantic audio codec for general sound. *arXiv preprint arXiv:2405.00233*, 2024.

[41] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[42] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[43] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

[44] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[45] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023.

[46] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[47] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

[48] Xu Li, Qirui Wang, and Xiaoyu Liu. Masksr: Masked language model for full-band speech restoration. *arXiv preprint arXiv:2406.02092*, 2024.

[49] Apoorv Vyas, Bowen Shi, Matthew Le, Andros Tjandra, Yi-Chiao Wu, Baishan Guo, Jiemin Zhang, Xinyue Zhang, Robert Adkins, William Ngan, et al. Audiobox: Unified audio generation with natural language prompts. *arXiv preprint arXiv:2312.15821*, 2023.

[50] Alexander H Liu, Matt Le, Apoorv Vyas, Bowen Shi, Andros Tjandra, and Wei-Ning Hsu. Generative pre-training for speech with flow matching. *arXiv preprint arXiv:2310.16338*, 2023.

[51] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[52] José Lezama, Huiwen Chang, Lu Jiang, and Irfan Essa. Improved masked image generation with token-critic. In *European Conference on Computer Vision*, pages 70–86. Springer, 2022.

[53] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[54] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[55] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[56] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[57] Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, et al. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. *arXiv preprint arXiv:2407.05361*, 2024.

[58] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[59] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[60] Puyuan Peng, Po-Yao Huang, Daniel Li, Abdelrahman Mohamed, and David Harwath. Voicecraft: Zero-shot speech editing and text-to-speech in the wild. *arXiv preprint arXiv:2403.16973*, 2024.

[61] Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, et al. Xtts: a massively multilingual zero-shot text-to-speech model. *arXiv preprint arXiv:2406.04904*, 2024.

[62] Heiga Zen, Rob Clark, Ron J. Weiss, Viet Dang, Ye Jia, Yonghui Wu, Yu Zhang, and Zhifeng Chen. Libritts: A corpus derived from librispeech for text-to-speech. In *Interspeech*, 2019. URL https://arxiv.org/abs/1904.02882.

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

[64] Tingwei Guo, Cheng Wen, Dongwei Jiang, Ne Luo, Ruixiong Zhang, Shuaijiang Zhao, Wubo Li, Cheng Gong, Wei Zou, Kun Han, et al. Didispeech: A large scale mandarin speech corpus. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6968–6972. IEEE, 2021.

[65] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[66] Yuanzhe Chen, Ming Tu, Tang Li, Xin Li, Qiuqiang Kong, Jiaxin Li, Zhichao Wang, Qiao Tian, Yuping Wang, and Yuxuan Wang. Streaming voice conversion via intermediate bottleneck features and non-streaming teacher guidance. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[67] Dongya Jia, Qiao Tian, Kainan Peng, Jiaxin Li, Yuanzhe Chen, Mingbo Ma, Yuping Wang, and Yuxuan Wang. Zero-shot accent conversion using pseudo siamese disentanglement network. *arXiv preprint arXiv:2212.05751*, 2022.

[68] Zhichao Wang, Yuanzhe Chen, Lei Xie, Qiao Tian, and Yuping Wang. Lm-vc: Zero-shot voice conversion via speech generation based on language models. *IEEE Signal Processing Letters*, 2023.

[69] Philip Anastassiou, Zhenyu Tang, Kainan Peng, Dongya Jia, Jiaxin Li, Ming Tu, Yuping Wang, Yuxuan Wang, and Mingbo Ma. Voiceshop: A unified speech-to-speech framework for identity-preserving zero-shot voice editing. *arXiv preprint arXiv:2404.06674*, 2024.

[70] Mateusz Łajszczak, Guillermo Cámbara, Yang Li, Fatih Beyhan, Arent van Korlaar, Fan Yang, Arnaud Joly, Álvaro Martín-Cortinas, Ammar Abbas, Adam Michalski, et al. Base tts: Lessons from building a billion-parameter text-to-speech model on 100k hours of data. *arXiv preprint arXiv:2402.08093*, 2024.

[71] Matthew Baas, Benjamin van Niekerk, and Herman Kamper. Voice conversion with just nearest neighbors. *arXiv preprint arXiv:2305.18975*, 2023.

[72] Paarth Neekhara, Shehzeen Hussain, Rafael Valle, Boris Ginsburg, Rishabh Ranjan, Shlomo Dubnov, Farinaz Koushanfar, and Julian McAuley. Selfvc: Voice conversion with iterative refinement using self transformations. *arXiv preprint arXiv:2310.09653*, 2023.

[73] Zengyi Qin, Wenliang Zhao, Xumin Yu, and Xin Sun. Openvoice: Versatile instant voice cloning. *arXiv preprint arXiv:2312.01479*, 2023.

[74] Xueyao Zhang, Xiaohui Zhang, Kainan Peng, Zhenyu Tang, Vimal Manohar, Yingru Liu, Jeff Hwang, Dangna Li, Yuhao Wang, Julian Chan, Yuan Huang, Zhizheng Wu, and Mingbo Ma. Vevo: Controllable zero-shot voice imitation with self-supervised disentanglement. *OpenReview*, 2024.

[75] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, 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.

[76] Beilong Tang, Bang Zeng, and Ming Li. Tselm: Target speaker extraction using discrete tokens and language models. *arXiv preprint arXiv:2409.07841*, 2024.

[77] Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent. Librimix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*, 2020.

[78] Haohe Liu, Xubo Liu, Qiuqiang Kong, Qiao Tian, Yan Zhao, DeLiang Wang, Chuanzeng Huang, and Yuxuan Wang. Voicefixer: A unified framework for high-fidelity speech restoration. *arXiv preprint arXiv:2204.05841*, 2022.

[79] Ziqian Wang, Xinfa Zhu, Zihan Zhang, YuanJun Lv, Ning Jiang, Guoqing Zhao, and Lei Xie. Selm: Speech enhancement using discrete tokens and language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11561–11565. IEEE, 2024.

[80] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. Wham!: Extending speech separation to noisy environments. *arXiv preprint arXiv:1907.01160*, 2019.

[81] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent. The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings. In *Proceedings of Meetings on Acoustics*, volume 19. AIP Publishing, 2013.

[82] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5220–5224. IEEE, 2017.

[83] Harishchandra Dubey, Ashkan Aazami, Vishak Gopal, Babak Naderi, Sebastian Braun, Ross Cutler, Alex Ju, Mehdi Zohourian, Min Tang, Mehrsa Golestaneh, et al. Icassp 2023 deep noise suppression challenge. *IEEE Open Journal of Signal Processing*, 2024.

[84] P. Ma, A. Haliassos, A. Fernandez-Lopez, H. Chen, S. Petridis, and M. Pantic. Auto-avsr: Audio-visual speech recognition with automatic labels. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[85] Jeongsoo Choi, Minsu Kim, and Yong Man Ro. Intelligible lip-to-speech synthesis with speech units. *arXiv preprint arXiv:2305.19603*, 2023.

[86] James Betker. Better speech synthesis through scaling. *arXiv preprint arXiv:2305.07243*, 2023.

[87] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36, 2024.

[88] Hubert Siuzdak. Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis. *arXiv preprint arXiv:2306.00814*, 2023.

[89] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

[90] Xueyao Zhang, Liumeng Xue, Yuancheng Wang, Yicheng Gu, Xi Chen, Zihao Fang, Haopeng Chen, Lexiao Zou, Chaoren Wang, Jun Han, et al. Amphion: An open-source audio, music and speech generation toolkit. *arXiv preprint arXiv:2312.09911*, 2023.

[91] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*, 2020.

[92] Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*, 2021.

[93] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR, 2021.

[94] Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, 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)*, pages 7669–7673. IEEE, 2020.

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

[96] Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey. Libriheavy: a 50,000 hours asr corpus with punctuation casing and context. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10991–10995. IEEE, 2024.

[97] Thai-Binh Nguyen and Alexander Waibel. Convoifilter: A case study of doing cocktail party speech recognition. In *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pages 565–569. IEEE, 2024.

[98] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[99] Foivos I Diakogiannis, François Waldner, Peter Caccetta, and Chen Wu. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162:94–114, 2020.

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

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

[102] Zhifu Gao, Zerui Li, Jiaming Wang, Haoneng Luo, Xian Shi, Mengzhe Chen, Yabin Li, Lingyun Zuo, Zhihao Du, Zhangyu Xiao, et al. Funasr: A fundamental end-to-end speech recognition toolkit. *arXiv preprint arXiv:2305.11013*, 2023.

[103] Chandan KA Reddy, Vishak Gopal, and Ross Cutler. Dnsmos: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6493–6497. IEEE, 2021.

[104] Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller. Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. *arXiv preprint arXiv:2104.09494*, 2021.

A Details of Two Types of Speech Discrete Representations

Metis employs two discrete speech representations for the two-stage generation, following the approach of Wang et al. [12].

1) **SSL tokens**: Derived from SSL features of large-scale speech self-supervised learning models [33, 34, 36, 37], SSL tokens encapsulate both semantic and prosodic information, making them well-suited for conditional generation. To minimize information loss, we employ a vector quantization (VQ) model [53, 54] to quantize SSL features into discrete tokens, in contrast to the k-means approach used in previous works [28, 29, 86]. SSL features are extracted from the 17th layer of `w2v-bert-2.0`[6] [34]. The VQ model has a codebook of 8,192 and a codebook dimension of 8. The model is trained using only reconstruction loss and VQ loss.

2) **Acoustic tokens**: Directly obtained from the waveform via vector quantization. The goal is to preserve all the information from the speech to reconstruct a high-quality waveform. We follow the recipe of DAC codec [87] for model architecture, discriminators, and training losses, with modifications to adopt the Vocos [88] decoder for more efficient training and inference. The 24 kHz speech waveform is compressed into discrete tokens using residual vector quantization (RVQ) across 12 layers, each with a codebook size of 1,024 and a codebook dimension of 8.

B Baselines

B.1 Zero-Shot TTS

VALL-E [8] It uses an AR transformer to predict codes from the first layer of EnCodec [89] and a NAR transformer to predict codes from the remaining layers of EnCodec. We use the released checkpoint in Amphion [90][7] which is pre-trained on 45K hours of the MLS [91] English set.

VoiceCraft [60] A token-filling neural codec language model for text editing and text-to-speech. It predicts multi-layer tokens in a delay pattern. We use the official code and checkpoint[8] which is pre-trained on 9K hours of GigaSpeech [92] dataset.

CosyVoice [32] A two-stage large-scale TTS system. The first stage is an autoregressive model and the second stage is a diffusion model. It is trained on 171K hours of multilingual speech data. We use the official code and checkpoint[9].

XTTS-v2 [61] An open-source multilingual TTS model that supports 16 languages. It is also based on an autoregressive model. We use the official code and checkpoint[10].

MaskGCT [12] An open-source large-scale NAR TTS system that eliminates the need for explicit alignment information between text and speech supervision, as well as phone-level duration prediction. It employs masked generative models for two-stage modeling and is trained on 100K hours of multilingual speech data. We use the official code and checkpoint[11].

NaturalSpeech 3 [10] A large-scale NAR TTS system featuring a factorized speech codec for speech decoupling representation and factorized diffusion models for speech generation. It achieves human-level naturalness on the LibriSpeech test set. We report the scores of LibriSpeech *test-clean* obtained from the original paper.

VoiceBox [11] A large-scale NAR multi-task speech generation model based on flow matching [51]. We report the scores of LibriSpeech *test-clean* obtained from the original paper.

[6] https://huggingface.co/facebook/w2v-bert-2.0
[7] https://github.com/open-mmlab/Amphion/tree/main/egs/tts/VALLE_V2
[8] https://huggingface.co/pyp1/VoiceCraft/blob/main/830M_TTSEnhanced.pth
[9] https://huggingface.co/model-scope/CosyVoice-300M
[10] https://huggingface.co/coqui/XTTS-v2
[11] https://github.com/open-mmlab/Amphion/blob/main/models/tts/maskgct

B.2 Voice Conversion

HireSpeech++ [21] A speech generation system designed based on the VITS architecture [93]. It is trained on 2.8K hours sourced from Libri-light [94] and LibriTTS [95]. We use the official code and checkpoint[12].

LM-VC [68] It uses an AR hierarchical transformer to predict SoundStream [47] codes from soft units similar to HuBERT [33] k-means tokens, trained on the Libri-light dataset [94]. It is not open-source, we report the scores obtained from [74].

UniAudio [26] An AR model system can perform multiple audio generation tasks. It uses 500-cluster k-means tokens from HuBERT [33] to predict their proposed acoustic codes for voice conversion. We use the official code and checkpoint[13]. It uses 60K hours of Libri-light [94] for training voice conversion.

Vevo [74] It is a versatile zero-shot voice imitation model which can control both timbre and style. It is trained on 60K hours of Libri-heavy [96]. We obtain the generated samples from the authors.

B.3 Target Speaker Extraction

UniAudio [26] An AR model system can perform multiple audio generation tasks including target speaker extraction. We use the official code and checkpoint.

VoiceFilter [22] A system that isolates a target speaker's voice from multi-speaker audio using a reference signal and neural networks for speaker embedding and spectrogram masking. We use the checkpoint [14] provided from Nguyen and Waibel [97].

WeSep [23] A target speaker extraction model trained on LibriMix [77] and VoxCeleb [98]. We use the official code and checkpoint[15].

TSELM [76] A target speaker extraction model that leverages a NAR transformer to predict discrete speech tokens driven from WavLM [37]. We use the official code and checkpoint[16].

B.4 Speech Enhancement

TF-GridNet [24] A deep neural network for speech separation integrating full- and sub-band modeling in the time-frequency (T-F) domain. It can also used for speech enhancement. We use the checkpoint[17] provided from Interspeech URGENT 2025 Challenge[18].

VoiceFixer [78] A generative framework to address the speech enhancement task. It consists of an analysis stage and a synthesis stage and employs a ResUNet [99] to model the analysis stage and a neural vocoder to model the synthesis stage. We use the official code and checkpoint[19].

SELM [79] A speech enhancement model that leverages an AR transformer to predict discrete speech tokens driven from WavLM [37]. We report the scores obtained from [48].

MaskSR [48] It uses masked generative models to predict acoustic tokens from DAC [87] codec for speech enhancement. We report the scores obtained from the original paper.

[12] https://github.com/sh-lee-prml/HierSpeechpp
[13] https://github.com/yangdongchao/UniAudio
[14] https://huggingface.co/nguyenvulebinh/voice-filter
[15] https://huggingface.co/spaces/wenet-e2e/wesep-tse-2speaker-demo/tree/main
[16] https://huggingface.co/Beilong/TSELM
[17] https://huggingface.co/kohei0209/tfgridnet_urgent25/tree/main
[18] https://urgent-challenge.github.io/urgent2025/
[19] https://github.com/haoheliu/voicefixer_main

B.5 Lip-to-Speech

Lip2Speech-Unit [25] A lip-to-speech model trained to generate multiple targets, mel-spectrograms and quantized self-supervised speech representations. We use the official code and checkpoint[20].

C Evaluation Metrics

SIM We evaluate speaker similarity between the prompt speech and the generated speech by computing the cosine similarity of the WavLM TDNN[21] [37] speaker embeddings between the generated sample and the prompt. SIM is reported for tasks involving prompt speech, including zero-shot TTS, voice conversion, target speaker extraction, and lip-to-speech. For speech enhancement, we compute SIM between the generated speech and ground truth using a separate checkpoint[22].

WER Word Error Rate measures the intelligibility of the generated speech. We use `whisper-large-v3`[23] [100] for all languages except Chinese, where we use `paraformer-zh`[24] [101, 102] as the ASR model to calculate WER. WER is reported for zero-shot TTS, voice conversion, target speaker extraction, and lip-to-speech.

DNSMOS [103] DNSMOS is a neural network-based mean opinion score estimator that correlates strongly with human quality ratings. It comprises three components: 1) speech quality (**SIG**), 2) background noise quality (**BAK**), and 3) overall quality (**OVRL**). We report DNSMOS scores for all tasks while providing the average scores for zero-shot TTS and VC tasks, and all three metrics for the remaining tasks.

NISQA [104] NISQA is a deep learning framework for speech quality prediction. We use the public checkpoint[25]. We report NISQA for voice conversion, speech enhancement, target speaker extraction, and lip-to-speech.

QMOS We use the Quality Mean Opinion Score (QMOS) for subjective listening tests to evaluate speech quality. QMOS is rated on a 5-point scale: 5 (Excellent), 4 (Good), 3 (Fair), 2 (Poor), and 1 (Bad).

SMOS We use the Speaker Mean Opinion Score (SMOS) for subjective listening tests to assess speaker similarity between the generated speech and the prompt. SMOS is rated on a 5-point scale: 5 (Excellent), 4 (Good), 3 (Fair), 2 (Poor), and 1 (Bad). SMOS is reported for zero-shot TTS.

[20] https://github.com/choijeongsoo/lip2speech-unitn

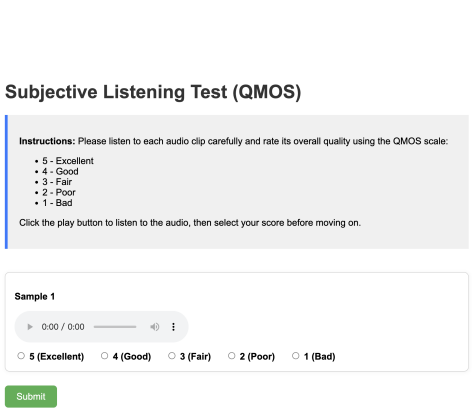
[21] https://github.com/microsoft/UniSpeech/tree/main/downstreams/speaker_verification

[22] https://huggingface.co/microsoft/wavlm-base-plus-sv

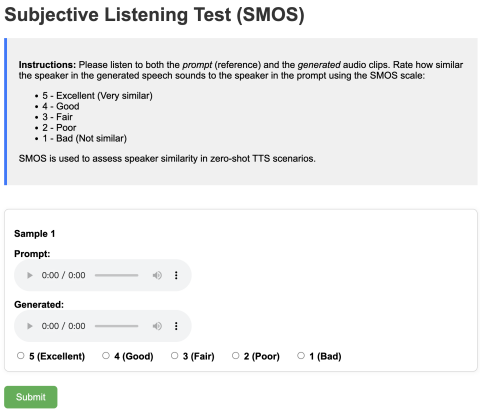
[23] https://huggingface.co/openai/whisper-large-v3

[24] https://huggingface.co/funasr/paraformer-zh

[25] https://github.com/gabrielmittag/NISQA/blob/master/weights/nisqa.tar



(a) A screenshot of the QMOS evaluation interface.



(b) A screenshot of the CMOS evaluation interface.

Figure 3: Screenshots of the subjective evaluation webpages.

D Additional Experimental Results

D.1 Zero-Shot TTS Results on LibriSpeech

Table 7 shows zero-shot TTS results on LibriSpeech *test-clean*. Metis-TTS achieves the highest SIM despite using significantly less training data, while maintaining a competitive WER.

Table 7: Zero-shot TTS results on LibriSpeech.

Model	Training Data	WER(↓)	SIM(↑)
LibriSpeech test-clean			
VALL-E [10]	45K EN	5.90	0.50
VoiceCraft [60]	9K EN	4.68	0.45
NaturalSpeech 3 [10]	60K EN	**1.94**	0.67
VoiceBox [10]	60K EN	2.03	0.64
Metis-TTS	10K Multi.	4.33	**0.70**

D.2 Voice Conversion Results on SeedTTS-VC

We also provide voice conversion results on SeedTTS-VC test sets in Table 8. The test sets are provided by [31][26]. Despite being fine-tuned on significantly less training data (0.4K hours compared to 2.8K hours used by HierSpeech++), Metis-VC achieves higher SIM scores on both SeedTTS-VC-en (0.76 vs. 0.56) and SeedTTS-VC-zh (0.63 vs. 0.39), demonstrating its strong capability in preserving speaker identity. Additionally, Metis-VC achieves a higher NISQA score than the baseline model, indicating improved speech quality.

Table 8: Voice conversion results on SeedTTS-VC test sets.

Model	Training Data	WER(↓)	SIM(↑)	DNSMOS(↑)	NISQA(↑)
SeedTTS-VC-en					
HierSpeech++ [21]	2.8K	4.48	0.56	3.57	3.47
Metis-VC fine-tune, cfg = 0.0	0.4K	7.05	**0.76**	3.51	**4.14**
SeedTTS-VC-zh					
HierSpeech++ [21]	2.8K	5.45	0.39	3.50	4.14
Metis-VC fine-tune, cfg = 0.0	0.4K	6.82	**0.63**	3.39	**3.82**

D.3 Results of Subjective Evaluation

We conduct subjective listening tests for two representative tasks: zero-shot TTS and speech enhancement. For zero-shot TTS, we randomly sampled 20 test samples from the results of SeedTTS *test-en*. Both QMOS and SMOS are reported for zero-shot TTS. The results are shown in Table 9. For speech enhancement, we randomly sampled 20 test samples from the results of DNS2020 with reverb. QMOS is reported for speech enhancement. The results are shown in Table 10.

[26] https://github.com/BytedanceSpeech/seed-tts-eval

Table 9: Results of subjective evaluation for zero-shot TTS.

Model	**QMOS**(↑)	**SMOS**(↑)
Ground Truth	4.32 ±0.28	4.01 ±0.30
VALL-E [8]	3.32 ±0.22	3.58 ±0.13
VoiceCraft [60]	3.58 ±0.17	3.65 ±0.19
CosyVoice [32]	4.02 ±0.10	3.97 ±0.21
Metis-TTS	**4.21** ±0.31	**4.19** ±0.18

Table 10: Results of subjective evaluation for speech enhancement.

Model	**QMOS**(↑)
Ground Truth	4.57 ±0.13
TF-GridNet [24]	3.65 ±0.35
VoiceFixer [78]	3.77 ±0.29
Metis-SE	**4.29** ±0.21

E Metis-Omni: Multi-Task Fine-Tuning

In addition to fine-tuning the pre-trained model separately for different tasks, we explore the potential of jointly fine-tuning a pre-trained model on multiple tasks resulting a multi-task model, which we refer to as **Metis-Omni**. For this study, we select four tasks: zero-shot TTS, voice conversion, target speaker extraction, and speech enhancement. We show more details in Appendix E. A subset of 10K hours of data is randomly sampled from the Emilia dataset as the shared training data for these tasks. During training, the task proportions are set as $p = \{0.5, 0.1, 0.2, 0.2\}$ for zero-shot TTS, voice conversion, target speaker extraction, and speech enhancement, respectively. As shown in Table 11, Metis-Omni achieves performance that is either on par with or surpasses task-specific fine-tuned models across all tasks. The only exception is the WER metric in zero-shot TTS, where performance is slightly lower. A possible reason for this is that, under the same number of training steps, multi-task fine-tuning allocates fewer steps to zero-shot TTS, leading to less task-specific optimization.

Table 11: Results of Metis-Omni on four speech generation tasks.

Task	**Dataset**	**Model**	**Performance**	
			Metrics	**Results**
Zero-shot TTS	***SeedTTS-en***	MaskGCT [12]	WER(↓) | SIM(↑) | DNSMOS(↑)	2.47 | 0.72 | 3.52
		Metis-TTS		**2.41** | **0.72** | **3.57**
		Metis-Omni		4.78 | 0.71 | **3.57**
	SeedTTS-zh	MaskGCT [12]	WER(↓) | SIM(↑) | DNSMOS(↑)	2.18 | 0.77 | 3.58
		Metis-TTS		**2.30** | **0.77** | 3.55
		Metis-Omni		4.39 | **0.77** | **3.60**
VC	***VCTK***	LM-VC [68]	WER(↓) | SIM(↑) |DNSMOS(↑)	8.35 | 0.29 | 3.46
		Metis-VC		6.65 | **0.48** | 3.49
		Metis-Omni		**3.52** | 0.34 | **3.51**
SE	***DNS2020 with reverb***	TF-Gridient [24]	SIG(↑) | BAK(↑) | OVRL(↑) | NISQA(↑)	3.11 | 3.23 | 2.51 | 2.61
		Metis-SE		**3.68** | **4.14** | **3.44** | **4.56**
		Metis-Omni		**3.68** | 4.13 | **3.44** | 4.53
	DNS2020 no reverb	TF-Gridient [24]	SIG(↑) | BAK(↑) | OVRL(↑) | NISQA(↑)	3.54 | 4.05 | 3.27 | 4.35
		Metis-SE		3.64 | **4.17** | 3.43 | 4.77
		Metis-Omni		**3.65** | **4.17** | **3.44** | **4.79**
	Real Recording	VoiceFixer [78]	SIG(↑) | BAK(↑) | OVRL(↑) | NISQA(↑)	3.31 | 3.93 | 3.00 | 3.66
		Metis-SE		3.59 | **4.01** | **3.27** | 3.95
		Metis-Omni		**3.60** | 4.00 | **3.27** | **3.96**
TSE	***LibriMix test***	WeSep [23]	SIG(↑) | BAK(↑) | OVRL(↑) | NISQA(↑)	3.56 | 3.39 | 3.23 | 4.04
		Metis-TSE		3.65 | 4.02 | 3.34 | 4.36
		Metis-Omni		**3.66** | **4.03** | **3.36** | **4.40**
	EmiliaMix test	WeSep [23]	SIG(↑) | BAK(↑) | OVRL(↑) | NISQA(↑)	3.50 | 3.85 | 3.12 | 3.86
		Metis-TSE		3.60 | **3.85** | 3.21 | 4.06
		Metis-Omni		**3.62** | 3.82 | **3.24** | **4.11**

F Speech Discrete Representation

Speech representation is a crucial aspect of speech generation. Recently, some speech generation systems [8, 10, 12, 28, 29, 30, 31, 74] have transitioned to using discrete speech representations, which can be broadly categorized into two types: 1) **SSL tokens**: typically derived by quantizing speech self-supervised learning features [33, 34, 35, 36, 37]. Unlike acoustic tokens, SSL tokens are designed not for directly reconstructing waveform but for encoding essential semantic and prosody information in speech, making them more suitable for prediction in conditional generation models. 2) **Acoustic tokens**: typically obtained by training a VQGAN [53, 54] model for waveform reconstruction, as used in speech codecs [47, 87, 89]. Acoustic tokens are effective for reconstructing high-quality waveforms. Currently, some speech generation systems [12, 29, 30, 39] utilize both types of representations for speech generation. In this work, we also adopt this two-stage paradigm, employing an MGM to generate SSL tokens from any condition and another MGM to generate acoustic tokens from SSL tokens.

G Limitation and Future Work

There are still some limitations that can be studied in the future.

Unified Audio Representation In this work, we utilize two distinct discrete speech representations, SSL tokens and acoustic tokens, for two-stage modeling, with SSL tokens specifically designed for the speech domain. Developing a unified discrete representation for all audio types, such as speech, music, and sound effects, that can seamlessly integrate with conditional generation models is an important direction for future research. Additionally, unifying the characteristics of SSL tokens and acoustic tokens to enable both high-quality waveform reconstruction and effective conditional modeling is equally significant.

Few-Shot Task Learning In this work, we adapt our pre-trained model to different characters through fine-tuning. In the field of NLP, large language models exhibit the remarkable ability to learn new tasks in a zero-shot manner without requiring additional training. This capability merits further investigation in future research.

H Impact Statement

Given that Metis is a powerful foundation model capable of generating high-quality speech across multiple tasks, it also presents potential risks of misuse, such as voice spoofing, speaker impersonation, and unauthorized content generation. To mitigate these risks, it is essential to develop robust detection mechanisms for synthetic speech, establish safeguards to prevent malicious use, and implement a responsible reporting system for identifying and addressing misuse cases.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clarify the scope and contribution of our paper in the abstract and the last two paragraphs of the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We discuss the limitations in Appendix G.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the implementation in Section 4 and Appendix D. And we will release our code and model checkpoints.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release our code and model checkpoints. We use open-source datasets.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide details about experimental setting and evaluation in Section 4 and Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For subjective evalution, we provide the confidence interval.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the information in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We do conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both potential positive societal impacts and negative societal impacts in Appendix H.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We discuss it in Appendix H.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original papers for all data and models we used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release our code and model checkpoints.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: We provide details about subject evaluation in Appendix C. Figure 3 shows sreenshots of the subjective evaluation webpages.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve potential risks.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [No]

Justification: We only use LLMs for editing.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.