# Prosody-TTS: Improving Prosody with Masked Autoencoder and Conditional Diffusion Model For Expressive Text-to-Speech

**Rongjie Huang**[1]*, **Chunlei Zhang**[2]*, **Yi Ren**[1], **Zhou Zhao**[1]†, **Dong Yu**[2]
Zhejiang University[1], Tencent AI Lab[2]
{rongjiehuang, rayeren, zhaozhou}@zju.edu.cn
{cleizhang, dyu}@global.tencent.com

## Abstract

Expressive text-to-speech aims to generate high-quality samples with rich and diverse prosody, which is hampered by **dual challenges**: 1) prosodic attributes in highly dynamic voices are difficult to capture and model without intonation; and 2) highly multimodal prosodic representations cannot be well learned by simple regression (e.g., MSE) objectives, which causes blurry and over-smoothing predictions. This paper proposes Prosody-TTS, a two-stage pipeline that enhances **prosody modeling and sampling** by introducing several components: 1) a self-supervised masked autoencoder to model the prosodic representation without relying on text transcriptions or local prosody attributes, which ensures to cover diverse speaking voices with superior generalization; and 2) a diffusion model to sample diverse prosodic patterns within the latent space, which prevents TTS models from generating samples with dull prosodic performance. Experimental results show that Prosody-TTS achieves new state-of-the-art in text-to-speech with natural and expressive synthesis. Both subjective and objective evaluation demonstrate that it exhibits superior audio quality and prosody naturalness with rich and diverse prosodic attributes. [1]

## 1 Introduction

Text-to-speech (TTS) (Wang et al., 2017; Ren et al., 2019; Kim et al., 2020; Huang et al., 2023) aims to generate human-like audios using text and auxiliary conditions, which attracts broad interest in the machine learning community. TTS models have been extended to more complex scenarios, requiring more natural and expressive voice generation with improved prosody modeling (Min et al., 2021; Chen et al., 2021; Li et al., 2021). A growing number of applications, such as personalized voice assistants and game commentary, have been actively developed and deployed to real-world applications.

Expressive text-to-speech aims to generate samples with natural, rich, and diverse prosodic attributes (e.g., duration, pitch, and energy), which is challenged by two major obstacles: 1) Prosody patterns (Qian et al., 2021; Wang et al., 2018) in human speech are often very sparse, which are difficult to capture and model without supervision signals (i.e., detailed transcriptions); 2) machine learning models (Li et al., 2018; Wang et al., 2022) usually learn a mean distribution over input data, resulting a dull prediction with prosody learners which fails to produce natural and diverse prosodic styles in the generated speech. Although recent studies (Choi et al., 2021; Kim et al., 2021; Ren et al., 2022) have proposed several ways to enhance prosody for high-fidelity TTS, there still exist **dual challenges**:

- **Prosody capturing and modeling.** Researchers leverage several designs to capture and model prosodic attributes: 1) Local prosody features. Ren et al. (2020) and Choi et al. (2021) introduce the idea of predicting pitch and energy explicitly. However, those signal processing-based prosodic attributes may have inevitable errors, which make the optimization of TTS models difficult and degrade performance. 2) Variational latent representations. A series of works (Sun et al., 2020; Kenter et al., 2019; Liu et al., 2022) utilize conditional variational auto-encoder to model prosody in a latent space, where global, local, or hierarchical features are sampled from a prior distribution. Nevertheless, they generally request speech-text parallel data for modeling prosody, which constrain the learned representation to the paired TTS data.

- **Prosody producing and sampling.** Most works (Wang et al., 2017; Min et al., 2021; Yang et al., 2021a) utilize regression losses (e.g.,

*Equal contributions
†Corresponding author
[1]Audio samples are available at https://improve-prosody.github.io/.



*Findings of the Association for Computational Linguistics: ACL 2023*, pages 8018–8034
July 9-14, 2023 

MSE) for prediction and assume that the latent space follows a unimodal distribution. However, the highly multimodal (a phoneme may be pronounced in various speaking styles) prosodic representations cannot be well modeled by these simple objectives, which causes blurry and over-smoothing predictions.

To address the above **dual challenges** for prosody-enhanced expressive text-to-speech, we propose Prosody-TTS, a two-stage TTS pipeline that improves both prosody modeling and sampling by introducing several novel designs:

- **Self-supervised prosody pre-training.** To handle different acoustic conditions for expressive speech, we propose prosody masked autoencoders (Prosody-MAE), a transformer-based model that captures prosody patterns (e.g., local rises and falls of the pitch and stress) in a self-supervised manner. It is trained on audio-only data, which avoids inevitable errors and ensures to cover diverse speech corpora with superior generalization.
- **Generative diffusion modeling in latent space.** A diffusion model is explored to bridge TTS inputs (i.e., text and target speaker) and speaking prosody in latent space. We formulate the generative process with multiple conditional diffusion steps, and thus we expect our model to exhibit better diversity and prevent generating samples with dull prosodic performance.

Experimental results on LJSpeech and LibriTTS benchmarks demonstrate that our proposed Prosody-TTS achieves new state-of-the-art results for text-to-speech with natural and expressive synthesis. Both subjective and objective evaluations demonstrate that Prosody-TTS exhibits superior audio quality and prosody naturalness with rich and diverse prosodic attributes.

## 2 Related Works

### 2.1 Prosody Modeling in Text-To-Speech

Prosody modeling has been studied for decades in the TTS community. The idea of pitch and energy prediction (Łańcucki, 2021; Ren et al., 2020) represents a popular way to address the one-to-many mapping challenges. Wang et al. (2019) utilize the VQ-VAE framework to learn a latent representation for the F0 contour of each linguistic unit and adopt a second-stage model which maps from linguistic features to the latent features. Choi et al. (2021) further use a new set of analysis features, i.e., the wav2vec and Yingram feature for self-supervised training. However, these signal processing-based prosodic attributes have inevitable errors, which make the optimization of TTS models difficult and result in degraded TTS performance. Instead of relying on local prosody attributes, a series of works (Sun et al., 2020; Kenter et al., 2019; Liu et al., 2022) utilize conditional variational autoencoder to model prosody in a latent space, where global, local, or hierarchical features are sampled from a prior distribution. Nevertheless, they generally request speech-text parallel data for modeling prosody, which constrained the learned representation to the paired TTS data and explicit poor generalization (Wang et al., 2022). Ren et al. (2022) introduces a prosody encoder to disentangle the prosody to latent vectors, while the requirement of a pre-trained TTS model hurts model generalization. In this work, we propose to learn the prosodic distribution given speech-only corpora without relying on pre-trained TTS models or text transcriptions.

### 2.2 Self-Supervised Learning in Speech

Recently, self-supervised learning (SSL) has emerged as a popular solution to many speech processing problems with a massive amount of unlabeled speech data. HuBERT (Hsu et al., 2021) is trained with a masked prediction with masked continuous audio signals. SS-AST (Gong et al., 2022) is a self-supervised learning method that operates over spectrogram patches. Baade et al. (2022) propose a simple yet powerful improvement over the recent audio spectrogram transformer (SSAST) model. Audio-MAE (Xu et al., 2022) is a simple extension of image-based Masked Autoencoders (MAE) (He et al., 2022) for SSL from audio spectrograms. Unlike most of the speech SSL models which capture linguistic content for style-agnostic representation, we focus on learning prosodic representation in expressive speech, which is relatively overlooked.

### 2.3 Diffusion Probabilistic Model

Denoising diffusion probabilistic models (DDPMs) (Ho et al., 2020; Song et al., 2020a) are likelihood-based generative models that have recently advanced the SOTA results in several important domains, including image (Dhariwal and Nichol, 2021; Song et al., 2020a), audio (Huang

et al., 2022c; Liu et al., 2021; Huang et al., 2022d), and 3D point cloud generation (Luo and Hu, 2021). In this work, we investigate generative modeling for latent representations with a conditional diffusion model. Unlike regression-based prediction, it generates realistic results that match the ground-truth distribution and avoid over-smoothing predictions.

## 3 Prosody-TTS

In this section, we first overview the Prosody-TTS framework, introducing several critical designs with prosody masked autoencoder (Prosody-MAE), latent diffusion model, and the vector quantization layer. Finally, we present the pre-training, training, and inference pipeline, which supports high-fidelity speech synthesis with natural, rich, and diverse prosodic attributes.

### 3.1 Problem Formulation

Expressive text-to-speech aims to generate high-fidelity speech samples with natural and diverse prosody (e.g., duration, pitch, and energy). Since the duration attribute has been inherently well-studied in non-autoregressive literature (Ren et al., 2020; Min et al., 2021; Huang et al., 2021, 2022b), we mainly explore prosody on **rises and falls of the pitch and stress** in this work.

### 3.2 Overview

As illustrated in Figure 1, to address the aforementioned **dual challenges** for prosody-enhanced expressive text-to-speech, we introduce a multi-stage pipeline with the following key designs: 1) a prosody masked autoencoder (Prosody-MAE) to **capture and model** prosody feature in a self-supervised manner. 2) a generative diffusion model to **produce and sample** prosody in latent space. Specifically:

1) In the pre-training stage, the Prosody-MAE captures prosodic information from large-scale unpaired speech data without relying on transcriptions or local prosody attributes. The self-supervised training manner ensures Prosody-MAE learns discriminative prosody representations covering diverse speech corpora; 2) In training TTS models, the converged prosody encoder derives style representations $\mathbf{z}$ for optimizing the latent diffusion model (LDM), which bridges the TTS conditions (i.e., textual features and target speaker) and prosody representations via diffusion process $q(\mathbf{z}_t|\mathbf{z}_{t-1})$; 3) In inference time, the LDM samples diverse latent representations within the prosodic space through reverse denoising $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)$. It breaks the generation process into several conditional diffusion steps, which exhibits better diversity and prevents generating dull samples with a constrained prosodic distribution. We describe these designs in detail in the following subsections.

### 3.3 Self-supervised Prosody Pre-training

In this part, we propose Prosody-MAE, a self-supervised autoencoder (AE) consisting of an encoder and decoder that can effectively capture and model prosodic style given speech samples without relying on text annotations. Moreover, we design several techniques to learn prosodic representation in a self-supervised manner:

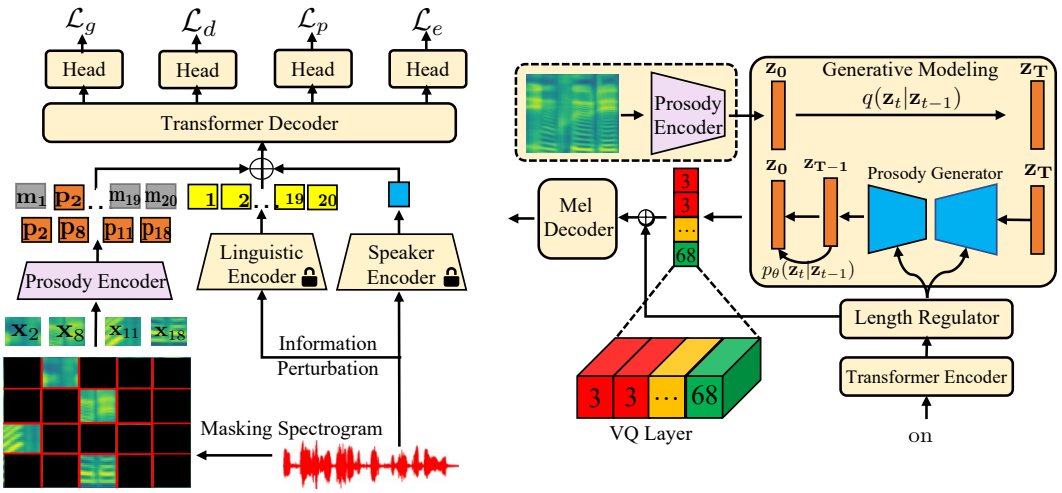
- Information flow. Through analysis of speech attributes, Prosody-MAE enjoys a carefully-crafted bottleneck design to disentangle linguistic and speaker information, ensuring the prosody stream to learn discriminative style-aware representations.
- Multi-task learning. Auxiliary style (i.e., pitch and energy) classifications have been included in training SSL models, and it guarantees to discover style representation aware of the pitch/stress rises and falls.

#### 3.3.1 Information Flow

Most voice reconstruction tasks (Choi et al., 2021; Polyak et al., 2021) can be defined by synthesizing and controlling three aspects of voice, **i.e., linguistic, speaker, and prosody encoder**. It motivates us to develop an autoencoder that can analyze voice into these properties and then synthesize them back into a speech **(transformer decoder)**.

**Linguistic Encoder.** Learning the linguistic content $\mathcal{C}$ from the speech signal is crucial to construct an intelligible speech signal, and we obtain linguistic representation using a pre-trained XLSR-53. Since SSL representation (Choi et al., 2021; Qian et al., 2022; Gat et al., 2022) contain both linguistic and acoustic information, we perturb the speaker and prosody patterns in audios by randomly shifting pitch and shaping energy values, ensuring it only provides the linguistic-related (i.e., prosodic-agnostic) information. More details have been included in Appendix E.

**Speaker Encoder.** Speaker $\mathcal{S}$ is perceived as the timbre characteristic of a voice. It has been




(a) Prosody-MAE (b) Prosody-TTS

Figure 1: In subfigure (a), the publicly-available pre-trained modules are printed with a *lock*. In subfigure (b), the converged prosody encoder from Prosody-MAE is denoted with dotted lines and included only for training TTS models.

reported that (Choi et al., 2021) the features from the first layer of XLSR-53 perform as clusters representation for each speaker.

**Prosody Encoder.** Prosody is a vital part of the domain style, where different emotions or styles have distinctive prosody patterns. In the multi-layer transformer prosody encoder, 1) speech is first transformed and embedded into spectrogram patches, and 2) the encoders $f : \mathcal{X} \mapsto \mathcal{P}$ take patches $\mathcal{X}$ as input and effectively capture prosodic latent representations $\mathbf{p}_1, \ldots, \mathbf{p}_T$ for $T$ time-steps. 3) Some tokens are masked by randomly replacing them with a learned masking token ($\mathbf{m}_i$ illustrated in Figure 1(a)). In practice, we mask by shuffling the input patches and keeping the first $1 - p$ proportion of tokens.

**Transformer Decoder.** As illustrated in Figure 1(a), we conduct the element-wise addition operation between the linguistic content $\mathcal{C}$, speaker $\mathcal{S}$ and the prosody $\mathcal{P}$ representations before passing through the transformer decoder with a series of transformer blocks. To this end, the carefully crafted bottleneck design in Prosody-MAE disentangles linguistic, speaker, and prosody attributes and then synthesizes them back into a speech with a transformer decoder, ensuring the prosody stream to learn discriminative prosody-aware representations agnostic to linguistic content and speaker.

#### 3.3.2 Multi-task Learning

For training autoencoders, reconstruction loss $\mathcal{L}_g$ is calculated as a mean squared error between the output of the linear reconstruction head and the input patches. Contrastive head (Gong et al., 2022) creates an output vector $\mathbf{v}_i$ similar to the masked input patch $\mathbf{x}_i$ but dissimilar to other masked inputs, where we consider different masked inputs as negative samples and implement the InfoNCE (Oord et al., 2018) as a criterion.

Moreover, to enhance the model in deriving style attributes, we explore the frame-level style (i.e., pitch $\mathcal{L}_p$, energy $\mathcal{L}_e$) classification with cross-entropy criterion (Oord et al., 2018) as the complemental tasks. To formulate the classification target, we respectively 1) quantize the fundamental frequency (f0) of each frame to 256 possible values $\mathbf{p}_i$ in log-scale; and 2) compute the L2-norm of the amplitude of each short-time Fourier transform (STFT) and then quantize to 256 possible values $\mathbf{e}_i$ uniformly. On that account, Prosody-MAE better discovers prosodic representations which are aware of the pitch/stress rises and falls.

### 3.4 Generative Modeling of Prosodic Representations

To produce and sample prosodic representation $\mathbf{z}$ within the latent space learned in Prosody-MAE, we implement our prosody generator over Latent Diffusion Models (LDMs) (Rombach et al., 2022; Gal et al., 2022), a recently introduced class of Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020) that operate in the latent space. As illustrated in Figure 1(c), the denoising WaveNet $\theta$ conditions on phonetic representation, breaking the generation process into several conditional diffusion steps. The training loss is defined as the mean squared error in the noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ space, and efficient training is optimizing a random

term of $t$ with stochastic gradient descent:

$$\mathcal{L}_\theta = \left\| \boldsymbol{\epsilon}_\theta \left( \alpha_t \mathbf{z}_0 + \sqrt{1-\alpha_t^2}\boldsymbol{\epsilon} \right) - \boldsymbol{\epsilon} \right\|_2^2 \quad (1)$$

To this end, our prosody generator produces and samples prosody faithfully, which strongly matches the ground-truth distribution and exhibits better diversity. It avoids incorrect unimodal distribution assumptions by regression objectives (e.g., MSE) and prevents generating samples with dull prosodic performance. We refer the reader to Section 4.2 for summary of our findings.

### 3.5 Vector Quantization

It has been reported (Rombach et al., 2022) that due to the expressiveness of diffusion models, the produced latent spaces $z$ could be highly variant and diverse. To avoid instability, we impose a vector quantization (VQ) layer after the latent diffusion for regularization.

Denote the latent space $e \in R^{K \times D}$ where $K$ is the size of the discrete latent space (i.e., a $K$-way categorical), and $D$ is the dimensionality of each latent embedding vector $e_i$. Note that there are $K$ embedding vectors $e_i \in \mathbb{R}^D, i \in 1, 2, \ldots, K$. To make sure the representation sequence commits to an embedding and its output does not grow, we add a commitment loss following previous work (van den Oord et al., 2017):

$$\mathcal{L}_c = \|\mathbf{z} - \mathrm{sg}[e]\|_2^2, \quad (2)$$

where sg stands for stop gradient.

### 3.6 Pre-training, Training and Inference Procedures

#### 3.6.1 Pre-training and Training

We pre-train the Prosody-MAE to derive prosodic representation in a self-supervised manner with the following objectives: 1) reconstruction loss $\mathcal{L}_g$: the MSE between the estimated and ground-truth sample; 2) contrastive loss $\mathcal{L}_d$: the discriminative gradient to pick the correct patch for each masked position from all patches being masked, and 3) frame-level style (i.e., pitch, energy) classification losses $\mathcal{L}_p, \mathcal{L}_e$: the cross entropy error between the estimated and ground-truth style attributes.

In training Prosody-TTS, the final loss terms consist of the following parts: 1) duration loss $\mathcal{L}_{dur}$: MSE between the predicted and the GT phoneme-level duration in log scale; 2) diffusion losses in prosody generator $\mathcal{L}_{ldm}$ and mel decoder $\mathcal{L}_{dec}$: calculating between the estimated and gaussian noise according to Equation 1; 3) commitment loss $\mathcal{L}_c$: regularizing vector quantization layer according to Equation 2.

#### 3.6.2 Inference

As illustrated in Figure 1, Prosody-TTS generates expressive speech with natural, rich, and diverse prosody in the following pipeline: 1) The text encoder takes the phoneme sequence as input, which is expanded according to the predicted durations; 2) conditioning on linguistic and speaker information, the prosody generator randomly samples a noise latent and iteratively denoises to produce a new prosodic representation in latent space, and 3) the mel decoder converts randomly sampled noise latent and iteratively decodes to expressive mel-spectrograms.

## 4 Experiments

### 4.1 Experimental Setup

#### 4.1.1 Pre-training Prosody-MAE

In the pre-training stage, we utilize the commonly-used LibriSpeech (Panayotov et al., 2015) dataset with labels discarded, which provides 960 hours of audiobook data in English, read by over 1,000 speakers. We convert the 16kHz waveforms into 128-dimensional log-Melfilterbank features with a frame length of 25 ms and frame shift of 10 ms. The spectrogram is then split into 16×16 patches.

By default, we use an encoder with 6 layers and a decoder of 2 layers, both using 12 heads and a width of 768. We train Prosody-MAE for up to 400k iterations on 8 NVIDIA V100 GPUs using the publicly-available *fairseq* framework (Ott et al., 2019), and the pre-training takes about 5 days. For downstream evaluation, we use the standard SUPERB (Yang et al., 2021b) training and testing framework. More detailed information has been attached in Appendix C.

#### 4.1.2 Training Prosody-TTS

**Dataset.** For a fair and reproducible comparison against other competing methods, we use the benchmark LJSpeech dataset (Ito, 2017), which consists of 13,100 audio clips from a female speaker for about 24 hours in total. For the multi-speaker scenario, we utilize the 300-hour LibriTTS (Zen et al., 2019) dataset derived from LibriSpeech. We convert the text sequence into the phoneme sequence with an open-source grapheme-to-phoneme conver-

| Method | LJSpeech | | | | | LibriTTS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MOS-P | MOS-Q | MCD | NDB | JSD | MOS-P | MOS-Q | MCD | NDB | JSD |
| GT | 4.36±0.05 | 4.39±0.06 | / | / | / | 4.38±0.05 | 4.42±0.06 | / | / | |
| GT(voc.) | 4.31±0.06 | 4.25±0.06 | 1.67 | 19 | 0.02 | 4.35±0.04 | 4.22±0.05 | 1.52 | 41 | 0.01 |
| FastSpeech 2 | 3.92±0.07 | 3.84±0.06 | 3.88 | 45 | 0.05 | 3.89±0.06 | 3.81±0.07 | 4.35 | 74 | 0.04 |
| StyleSpeech | 3.94±0.06 | 3.88±0.05 | 5.54 | 41 | 0.07 | 3.95±0.07 | 3.91±0.08 | 3.78 | 58 | **0.01** |
| Glow-TTS | 3.88±0.06 | 3.91±0.06 | 3.54 | 34 | **0.03** | 3.91±0.08 | 3.86±0.08 | 5.38 | 61 | 0.03 |
| Grad-TTS | 3.91±0.07 | 3.92±0.06 | 5.01 | 49 | 0.13 | 3.96±0.06 | 3.97±0.05 | 3.93 | 71 | 0.05 |
| YourTTS | 3.97±0.06 | 3.96±0.06 | 5.09 | 47 | 0.08 | 3.99±0.07 | 3.99±0.06 | 4.61 | 73 | 0.06 |
| Prosody-TTS | **4.10±0.06** | **4.03±0.05** | **3.52** | **30** | 0.04 | **4.12±0.07** | **4.09±0.06** | **3.39** | **52** | **0.01** |

Table 1: **Performance (audio quality and prosody naturalness) comparison with other models.** We report the evaluation metrics including MOS(↑), MCD(↓), NDB(↓), and JSD(↓). The mel-spectrograms are converted to waveforms using Hifi-GAN (V1).

| Model | Prosody Capturing | Prosody Sampling |
|---|---|---|
| FastSpeech 2 | Local Prosody | Regression |
| StyleSpeech | Local Prosody | Regression |
| Glow-TTS | Local Prosody | Generative |
| Grad-TTS | Local Prosody | Generative |
| YourTTS | Variational | Generative |
| Prosody-TTS | Self-Supervised | Generative |

Table 2: Prosody modeling and sampling approaches comparison with other models.

sion tool (Sun et al., 2019) [2].

Following the common practice (Chen et al., 2021; Min et al., 2021), we conduct preprocessing on the speech and text data: 1) convert the sampling rate of all speech data to 16kHz; 2) extract the spectrogram with the FFT size of 1024, hop size of 256, and window size of 1024 samples; 3) convert it to a mel-spectrogram with 80 frequency bins.

**Model Configurations.** Prosody-TTS consists of 4 feed-forward transformer blocks for the phoneme encoder. We add a linear layer to transform the 768-dimensional prosody latent representation from Prosody-MAE to 256 dimensions. The default size of the codebook in the vector quantization layer is set to 1000. The diffusion model comprises a 1x1 convolution layer and $N$ convolution blocks with residual connections to project the input hidden sequence with 256 channels. For any step $t$, we use the cosine schedule $\beta_t = \cos(0.5\pi t)$. More detailed information has been attached in Appendix A.

**Training and Evaluation.** We train Prosody-TTS for 200,000 steps using 4 NVIDIA V100 GPUs with a batch size of 64 sentences. Adam optimizer is used with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$. We utilize HiFi-GAN (Kong et al., 2020) as the vocoder to synthesize waveform from the mel-spectrogram in our experiments.

We conduct crowd-sourced human evaluations on the testing set via Amazon Mechanical Turk, which is reported with 95% confidence intervals (CI). We analyze the MOS in two aspects: prosody (naturalness of pitch, energy, and duration) and audio quality (clarity, high-frequency and original timbre reconstruction), respectively scoring MOS-P and MOS-Q. We further include objective evaluation metrics: MCD (Kubichek, 1993) measures the audio and prosody quality, NDB and JSD (Richardson and Weiss, 2018) explore the diversity of generated mel-spectrograms. More details have been attached in Appendix F.

**Baseline Models.** We compare the quality of generated audio samples with other systems, including 1) GT, the ground-truth audio; 2) GT (voc.), we first convert the ground-truth audio into mel-spectrograms and then convert them back to audio using HiFi-GAN (V1) (Kong et al., 2020); 3) FastSpeech 2 (Ren et al., 2020): a model that predicts local prosody attributes; 4) Meta-StyleSpeech (Kim et al., 2020): the finetuned multi-speaker model with meta-learning; 5) Glow-TTS (Kim et al., 2020): a flow-based TTS model trained with monotonic alignment search; 6) Grad-TTS (Popov et al., 2021): a denoising diffusion probabilistic models for speech synthesis. 7) YourTTS (Casanova et al., 2022): an expressive model for zero-shot multi-speaker synthesis which is built upon VITS (Kim et al., 2021). We list the prosody modeling and sampling approaches in baseline models in Table 2.

### 4.2 Quantitative Results

Both objective and subjective evaluation results are presented in Table 1, and we have the following observations: 1) In terms of **audio quality**, Prosody-TTS achieves the highest perceptual quality with MOS-Q of 4.03 (LJSpeech) and 4.09

[2] https://github.com/Kyubyong/g2p

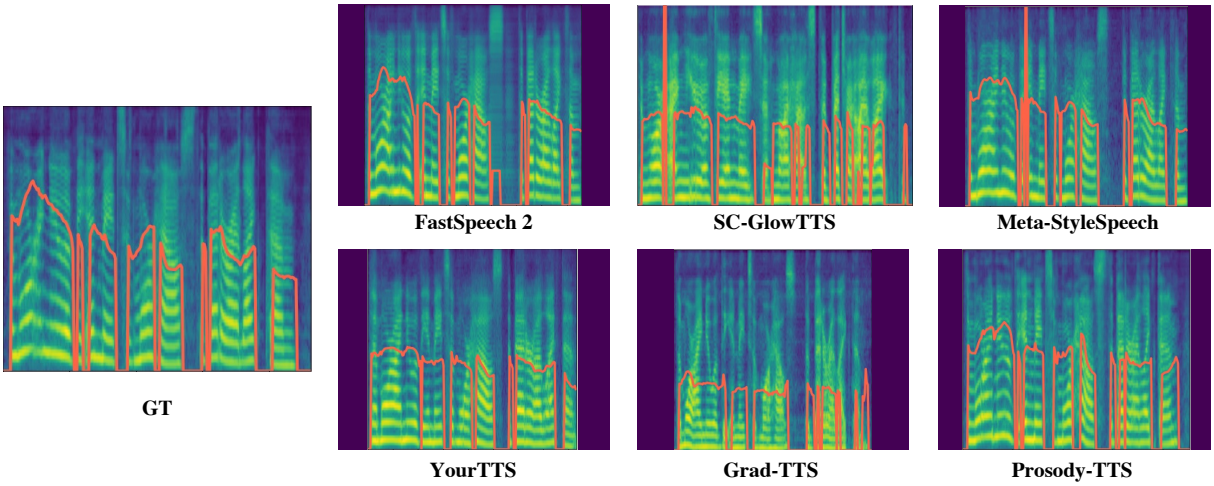



Figure 2: Visualizations of the generated mel-spectrograms. The corresponding text of generated speech samples is "*there was not a worse vagabond in Shrewsbury than old Barney the piper.*".

(LibriTTS). For objective evaluation, Prosody-TTS also demonstrates the outperformed performance in MCD, superior to all baseline models. 2) For **prosody diversity and naturalness**, Prosody-TTS scores the highest overall MOS-P with a gap of 0.21 (LJSpeech) and 0.23 (LibriTTS) compared to the ground truth audio. Prosody-TTS scores the superior NDB with scores of 30 (LJSpeech) and 52 (LibriTTS), producing samples covering diverse prosodic patterns (e.g., local rises and falls of the pitch and stress). Informally, by breaking the generation process into several conditional diffusion steps, generative latent modeling prevents TTS from synthesizing samples with dull prosodic performance.

The evaluation of the TTS models is very challenging due to its subjective nature in perceptual quality, and thus we include a site-by-site AXY test in Table 3. For each reference (A), the listeners are asked to choose a preferred one among the samples synthesized by baseline models (X) and proposed Prosody-TTS (Y), from which AXY preference rates are calculated. It indicates that raters prefer our model synthesis against baselines in terms of prosody naturalness and expressiveness. Without relying on text transcriptions or local prosody attributes, Prosody-TTS is trained on an audio-only corpus in a self-supervised manner, covering diverse speaking styles and avoiding dull synthesis with similar patterns.

### 4.3 Qualitative Findings

As illustrated in Figure 2, we plot the mel-spectrograms and corresponding pitch tracks generated by the TTS systems and have the following observations: 1) Prosody-TTS can generate mel-spectrograms with rich details in frequency bins between two adjacent harmonics, unvoiced frames, and high-frequency parts, which results in more natural sounds. 2) Prosody-TTS demonstrates its ability to generate samples with diverse prosodic styles. In contrast, some baseline models have difficulties addressing the dual challenges of prosody modeling and sampling: some of them learn a mean pitch contour (YourTTS, Grad-TTS) or incomplete sampling (FastSpeech 2), others suffer from a perturbed distribution with acute contour (SC-GlowTTS, Meta-StyleSpeech).

| Baseline | 7-point score | X | Neutral | Y |
|---|---|---|---|---|
| FastSpeech 2 | 1.13 ±0.19 | 21% | 10% | 69% |
| StyleSpeech | 1.50±0.11 | 33% | 12% | 55% |
| Glow-TTS | 1.11±0.11 | 13% | 22% | 65% |
| Grad-TTS | 1.20±0.08 | 19 % | 21% | 60% |
| YourTTS | 1.42±0.10 | 28% | 13% | 59% |

Table 3: The AXY preference test results. The scale ranges of 7-point are from "X is much closer" to "Both are about the same distance" to "Y is much closer", and can naturally be mapped on the integers from -3 to 3.

### 4.4 Ablation Studies and Model Properties

In this section, we conduct ablation studies to demonstrate the effectiveness of several designs to alleviate the dual challenges in prosody-enhanced text-to-speech:

- For **prosody capturing and modeling**, we explore Prosody-MAE with different model properties in the style-aware downstream challenges, including the frame-level pitch and energy recognition on the commonly-used

| Objective | PA | PM | EM |
|---|---|---|---|
| $\mathcal{L}_g$ | 70.0 | 8.35 | 4.63 |
| $+\mathcal{L}_d$ | 73.1 | 7.50 | 3.13 |
| $+\mathcal{L}_d + \mathcal{L}_p + \mathcal{L}_e$ | **75.2** | **7.22** | **1.76** |

(a) **Pretext Task**

| IF | IP | PA | PM | EM |
|---|---|---|---|---|
| ✗ | ✗ | 73.0 | 7.50 | 3.13 |
| ✓ | ✗ | 74.9 | 7.76 | 6.26 |
| ✓ | ✓ | **75.2** | **7.22** | **1.76** |

(b) **Information Flow**

| Model | CMOS-P | CMOS-Q |
|---|---|---|
| Prosody-MAE | 0.00 | 0.00 |
| w/o LDM | -0.11 | -0.04 |
| w/o VQ | -0.04 | -0.08 |
| Local Prosody | -0.12 | -0.02 |
| Variational Inference | -0.10 | -0.03 |

(c) **Prosody Sampling**

Table 4: **Ablation studies and model propagates**. For **prosody capturing and modeling**, we report accuracy (PA↑), mean absolute error (PM↓) in pitch recognition, and mean absolute error (EM↓) in energy recognition. For **prosody producing and sampling**, we evaluate through human ratings with CMOS-P/CMOS-Q (↑). In table (b), we use IF and IP to denote the carefully-crafted information flow design and the perturbation. More ablations have been attached in Appendix B, and details on evaluation metrics are included in Appendix F.

dataset IEMOCAP (Busso et al., 2008).

- For **prosody producing and sampling**, we investigate the generative modeling in Prosody-TTS with diffusion prosody generator and vector quantization modulewith through CMOS evaluation.

#### 4.4.1 Prosody capturing and modeling

**Pretext task.** We investigate the impact of different pretext tasks for pre-training the Prosody-MAE, and find that 1) the additional contrastive objective $\mathcal{L}_d$ leads to better performance for all tasks, and 2) the joint multi-task learning with frame-level style classification $\mathcal{L}_p, \mathcal{L}_e$ has witnessed a distinct promotion of downstream accuracy, demonstrating its efficiency in learning style-aware prosody representations.

**Information flow.** We conduct ablation studies to demonstrate the effectiveness of the carefully-crafted information flow in learning prosodic style attributes: 1) Dropping the linguistic and speaker encoder has witnessed a distinct degradation of downstream performance, proving that they disentangle the linguistic and speaker information, ensuring the prosody stream to learn style-aware representations; and 2) Removing the information perturbation also decreases accuracy, demonstrating that the perturbation assists to selectively provide only the linguistic (i.e., prosodic-agnostic) and eliminate undesired information.

More ablations on **masking strategies**, **network architecture**, and further **comparision with other state-of-the-art** have been attached in Appendix B

#### 4.4.2 Prosody producing and sampling

To verify the effectiveness of prosody producing and sampling in Prosody-TTS, we respectively replace the latent diffusion model and remove the vector quantization module. The CMOS evaluation results have been presented in Table 4(c), and we have the following observations: 1) Replacing the diffusion prosody generator with regression-based predictor results in decrease prosody naturalness, suggesting that generative latent diffusion avoids producing blurry and over-smoothing results. 2) Removing the vector quantization layer has witnessed a distinct drop in audio quality, verifying that the VQ compression layer is efficient in regularizing latent spaces and preventing arbitrarily high-variance predictions. 3) Since baseline models with local attributes have inevitable errors, and variational inference requires parallel speech-text data which constrains learned representation, they both lead to the degradation in prosody naturalness.

## 5 Conclusion

In this work, we propose Prosody-TTS, improving prosody with masked autoencoder and conditional diffusion model for expressive text-to-speech. To tackle **dual challenges of prosody modeling and sampling**, we design a two-stage pipeline to enhance high-quality synthesis with prosperous and diverse prosody: 1) Prosody-MAE was introduced to pre-train on large-scale unpaired speech datasets to capture prosodic representations without relying on text transcriptions. It ensured that the model covered diverse speaking voices and avoided inevitable error. 2) The latent diffusion model was adopted to produce diverse patterns within the learned prosody space. It broke the generation process into several conditional diffusion steps, avoiding generating samples with dull prosodic performance. Experimental results demonstrated that Prosody-TTS promoted prosody modeling and synthesized high-fidelity speech samples, achieving new state-of-the-art results with outperformed audio quality and

prosody expressiveness. For future work, we will further extend Prosody-TTS to more challenging scenarios such as multilingual prosody learning. We envisage that our work serve as a basis for future prosody-aware TTS studies.

## 6 Limitation

Prosody-TTS adopts generative diffusion models for high-quality synthesis, and thus it inherently requires multiple iterative refinements for better results. Besides, latent diffusion models require typically require more computational resources, and degradation could be witnessed with decreased training data. One of our future directions is to develop lightweight and fast diffusion models for accelerating sampling.

## 7 Ethics Statement

Prosody-TTS lowers the requirements for high-quality and expressive text-to-speech synthesis, which may cause unemployment for people with related occupations, such as broadcasters and radio hosts. In addition, there is the potential for harm from non-consensual voice cloning or the generation of fake media, and the voices of the speakers in the recordings might be overused than they expect.

## Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant No.2022ZD0162000, National Natural Science Foundation of China under Grant No.62222211, Grant No.61836002 and Grant No.62072397.

## References

Alan Baade, Puyuan Peng, and David Harwath. 2022. Mae-ast: Masked autoencoding audio spectrogram transformer. *arXiv preprint arXiv:2203.16691*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359.

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

Mingjian Chen, Xu Tan, Bohan Li, Yanqing Liu, Tao Qin, Sheng Zhao, and Tie-Yan Liu. 2021. Adaspeech: Adaptive text to speech for custom voice. *arXiv preprint arXiv:2103.00993*.

Hyeong-Seok Choi, Juheon Lee, Wansoo Kim, Jie Lee, Hoon Heo, and Kyogu Lee. 2021. Neural analysis and synthesis: Reconstructing speech from self-supervised representations. *Advances in Neural Information Processing Systems*, 34:16251–16265.

Prafulla Dhariwal and Alex Nichol. 2021. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*.

Itai Gat, Felix Kreuk, Ann Lee, Jade Copet, Gabriel Synnaeve, Emmanuel Dupoux, and Yossi Adi. 2022. On the robustness of self-supervised representations for spoken language modeling. *arXiv preprint arXiv:2209.15483*.

Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. 2022. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10699–10709.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Kuan Po Huang, Yu-Kuan Fu, Yu Zhang, and Hung-yi Lee. 2022a. Improving distortion robustness of self-supervised speech processing tasks with domain adaptation. *arXiv preprint arXiv:2203.16104*.

Rongjie Huang, Feiyang Chen, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. 2021. Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3945–3954.

Rongjie Huang, Chenye Cui, Feiyang Chen, Yi Ren, Jinglin Liu, Zhou Zhao, Baoxing Huai, and Zhefeng Wang. 2022b. Singgan: Generative adversarial network for high-fidelity singing voice generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2525–2535.

Rongjie Huang, Max WY Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao. 2022c. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. *arXiv preprint arXiv:2204.09934*.

Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, et al. 2023. Audiogpt: Understanding and generating speech, music, sound, and talking head. *arXiv preprint arXiv:2304.12995*.

Rongjie Huang, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. Generspeech: Towards style transfer for generalizable out-of-domain text-to-speech. In *Advances in Neural Information Processing Systems*.

Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. 2022d. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2595–2605.

Keith Ito. 2017. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/.

Tom Kenter, Vincent Wan, Chun-An Chan, Rob Clark, and Jakub Vit. 2019. Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network. In *International Conference on Machine Learning*, pages 3331–3340. PMLR.

Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. 2020. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems*, 33:8067–8077.

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

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Proc. of NeurIPS*.

Robert Kubichek. 1993. Mel-cepstral distance measure for objective speech quality assessment. 1:125–128.

Max W. Y. Lam, Jun Wang, Rongjie Huang, Dan Su, and Dong Yu. 2021. Bilateral denoising diffusion models.

Adrian Łańcucki. 2021. Fastpitch: Parallel text-to-speech with pitch prediction. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6588–6592. IEEE.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. 2018. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409.

Xiang Li, Changhe Song, Jingbei Li, Zhiyong Wu, Jia Jia, and Helen Meng. 2021. Towards multi-scale style control for expressive speech synthesis. *arXiv preprint arXiv:2104.03521*.

Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, Peng Liu, and Zhou Zhao. 2021. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. *arXiv preprint arXiv:2105.02446*, 2.

Zhengxi Liu, Qiao Tian, Chenxu Hu, Xudong Liu, Menglin Wu, Yuping Wang, Hang Zhao, and Yuxuan Wang. 2022. Controllable and lossless non-autoregressive end-to-end text-to-speech. *arXiv preprint arXiv:2207.06088*.

Shitong Luo and Wei Hu. 2021. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845.

Dongchan Min, Dong Bok Lee, Eunho Yang, and Sung Ju Hwang. 2021. Meta-stylespeech: Multi-speaker adaptive text-to-speech generation. pages 7748–7759.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. 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.

Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. Speech resynthesis from discrete disentangled self-supervised representations. *arXiv preprint arXiv:2104.00355*.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. 2021. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, pages 8599–8608. PMLR.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Mark Hasegawa-Johnson, and David Cox. 2020. Unsupervised speech decomposition via triple information bottleneck. In *International Conference on Machine Learning*, pages 7836–7846. PMLR.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Jinjun Xiong, Chuang Gan, David Cox, and Mark Hasegawa-Johnson. 2021. Global rhythm style transfer without text transcriptions. *arXiv preprint arXiv:2106.08519*.

Kaizhi Qian, Yang Zhang, Heting Gao, Junrui Ni, Cheng-I Lai, David Cox, Mark Hasegawa-Johnson, and Shiyu Chang. 2022. Contentvec: An improved self-supervised speech representation by disentangling speakers. In *International Conference on Machine Learning*, pages 18003–18017. PMLR.

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

Yi Ren, Ming Lei, Zhiying Huang, Shiliang Zhang, Qian Chen, Zhijie Yan, and Zhou Zhao. 2022. Prosospeech: Enhancing prosody with quantized vector pre-training in text-to-speech. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7577–7581. IEEE.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. *Advances in Neural Information Processing Systems*, 32.

Eitan Richardson and Yair Weiss. 2018. On gans and gmms. In *Proc. of ICONIP*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020a. Denoising diffusion implicit models. In *Proc. of ICLR*.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020b. Score-based generative modeling through stochastic differential equations. In *Proc. of ICLR*.

Guangzhi Sun, Yu Zhang, Ron J Weiss, Yuan Cao, Heiga Zen, and Yonghui Wu. 2020. Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6264–6268. IEEE.

Hao Sun, Xu Tan, Jun-Wei Gan, Hongzhi Liu, Sheng Zhao, Tao Qin, and Tie-Yan Liu. 2019. Token-level ensemble distillation for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1904.03446*.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6309–6318.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. 2022. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*.

Xin Wang, Shinji Takaki, Junichi Yamagishi, Simon King, and Keiichi Tokuda. 2019. A vector quantized variational autoencoder (vq-vae) autoregressive neural $f\_0$ model for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:157–170.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*.

Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous. 2018. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, pages 5180–5189. PMLR.

Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, Christoph Feichtenhofer, et al. 2022. Masked autoencoders that listen. *arXiv preprint arXiv:2207.06405*.

Jinhyeok Yang, Jae-Sung Bae, Taejun Bak, Youngik Kim, and Hoon-Young Cho. 2021a. Ganspeech: Adversarial training for high-fidelity multi-speaker speech synthesis. *arXiv preprint arXiv:2106.15153*.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al. 2021b. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*.

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

## A Details of Models

In this section, we describe hyper-parameters and details of several modules.

### A.1 Model Configurations

We list the model hyper-parameters of Prosody-TTS in Table 5.

### A.2 Diffusion mechanism

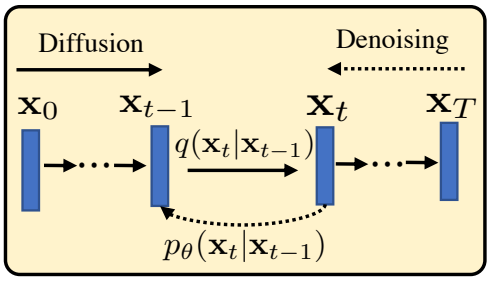



Figure 3: Graph for Diffusion.

For the training prosody latent diffusion model, the clean prosodic representation derived by Prosody-MAE passes through the vector quantification layer, which is also adopted to optimize the latent diffusion model (LDM) via the forward diffusion process. In inference time, the LDM samples diverse latent representations within the prosodic space through reverse backward denoising. According to the spectrogram denoiser, sampling from the Gaussian prior distribution is regarded as a common assumption. The diffusion decoder receives the textual hidden representation as a conditional signal and iteratively denoises Gaussian noise to reconstruct the target distribution by reverse sampling.

## B Downstream Evaluation on Model Properties

In the fine-tuning phase, we remove the decoder and only fine-tune the encoder on the commonly-used dataset IEMOCAP (Busso et al., 2008) that contains about 12 hours of emotional speech. We use a fixed learning rate of 1e-4 and max iteration of 10k and fine-tune on 4 V100 GPUs for 60 epochs using the SUPERB (Yang et al., 2021b) framework. We further evaluate the architecture and masking strategies designs in Prosody-MAE:

**Network architecture.** Similar to the MAE paper demonstrated for the visual domain, increasing the decoder depth only provides minor improvements if any, indicating that the decoder depth can be small relative to the encoder.

**Masking strategies.** We compare different masking ratios for pre-training Prosody-MAE, and observe that a high masking ratio (70% in our case) is optimal for audio spectrograms. Due to the fact that audio spectrograms and images are continuous signals with significant redundancy, and thus SSL models still could reconstruct results given most tokens dropped, which is consistent with the masked autoencoders (He et al., 2022) in the visual domain.

**Comparision with other state-of-the-art.** We compare our proposed Prosody-MAE with prior state-of-the-art SSL models, including: 1) wav2vec 2.0 (Baevski et al., 2020), 2) hubert (Hsu et al., 2021), 3) robust hubert (Huang et al., 2022a), and 4) mae-ast (Baade et al., 2022) and find that our proposed Prosody-MAE achieves the best performance across all tasks compared to other systems. Specifically, the majority of the speech SSL models focus on learning the linguistic content information, which try to disentangle unwanted variations (e.g. acoustic variations) from the content. In contrast, we hope to capture prosodic information from speech, and thus Prosody-MAE exhibits outperformed capability in capturing style attributes.

## C Details of Pre-training and Fine-tuning

We list the pre-training and fine-tuning settings in Table 7.

| Settings | | Values |
|---|---|---|
| Pre-training | Optimizer | Adam |
| | Base Learning Rate | 0.0001 |
| | Batch Size | 900 |
| | Optimizer Momentum | 0.9,0.98 |
| | Weight Decay | 0.01 |
| | Warmup Updates | 32000 |
| Fine-tuning | Optimizer | Adam |
| | Base Learning Rate | 0.0001 |
| | Batch Size | 4 |

Table 7: Pre-training and fine-tuning settings.

## D Diffusion Probabilistic models

Given i.i.d. samples $\{\mathbf{x}_0 \in \mathbb{R}^D\}$ from an unknown data distribution $p_{data}(\mathbf{x}_0)$. In this section, we introduce the theory of diffusion probabilistic model (Ho et al., 2020; Lam et al., 2021; Song et al., 2020a,b), and present diffusion and reverse process given by denoising diffusion probabilistic models (DDPMs), which could be used to learn a model distribution $p_\theta(\mathbf{x}_0)$ that approximates $p_{data}(\mathbf{x}_0)$.

| Hyperparameter | | Prosody-TTS |
|---|---|---|
| Text Encoder | Phoneme Embedding | 192 |
| | Encoder Layers | 4 |
| | Encoder Hidden | 256 |
| | Encoder Conv1D Kernel | 9 |
| | Encoder Conv1D Filter Size | 1024 |
| | Encoder Attention Heads | 2 |
| | Encoder Dropout | 0.1 |
| Duration Predictor | Duration Predictor Conv1D Kernel | 3 |
| | Duration Predictor Conv1D Filter Size | 256 |
| | Duration Predictor Dropout | 0.5 |
| Prosody Generator | VQ Codebook Size | 1000 |
| | Latent Diffusion Residual Layers | 30 |
| | Latent Diffusion Residual Channels | 256 |
| | Latent Diffusion WaveNet Conv1d Kernel | 3 |
| | Latent Diffusion WaveNet Conv1d Filter | 512 |
| Diffusion Decoder | Diffusion Embedding | 256 |
| | Residual Layers | 20 |
| | Residual Channels | 256 |
| | WaveNet Conv1d Kernel | 3 |
| | WaveNet Conv1d Filter | 512 |
| Total Number of Parameters | | 53M |

Table 5: Hyperparameters of Prosody-TTS models.

| Layers | PA | PM | EM |
|---|---|---|---|
| 2 | 75.2 | **7.22** | **1.76** |
| 4 | 75.3 | 7.41 | 2.01 |
| 6 | **75.5** | 7.73 | 2.25 |
| 8 | 74.6 | 7.85 | 2.52 |

(a) **Network Architecture**

| Mask Ratio | PA | PM | EM |
|---|---|---|---|
| 80% | **75.2** | 7.22 | 1.76 |
| 70% | **75.2** | 7.11 | **1.65** |
| 60% | 74.9 | **7.05** | 2.11 |
| 50% | 74.6 | 7.34 | 2.82 |

(b) **Masking Strategies**

| Model | PA | PM | EM |
|---|---|---|---|
| wav2vec 2.0 | 70.7 | 7.34 | 3.21 |
| HuBERT | 69.9 | 8.00 | 5.63 |
| Robust HuBERT | 69.5 | 7.95 | 5.37 |
| MAE-AST | 73.1 | 8.17 | 5.43 |
| Prosody-MAE | **75.2** | **7.22** | **1.76** |

(c) **Comparision with other state-of-the-art**

Table 6: **Ablations and model properties**. We report the evaluation metrics including accuracy (PA↑), mean absolute error (PM↓) in pitch recognition, and mean absolute error (EM↓) in energy recognition to evaluate model properties.

**Diffusion process** Similar as previous work (Ho et al., 2020; Song et al., 2020a), we define the data distribution as $q(\mathbf{x}_0)$. The diffusion process is defined by a fixed Markov chain from data $x_0$ to the latent variable $x_T$:

$$q(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T|x_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \quad (3)$$

For a small positive constant $\beta_t$, a small Gaussian noise is added from $x_t$ to the distribution of $x_{t-1}$ under the function of $q(x_t|x_{t-1})$.

The whole process gradually converts data $x_0$ to whitened latent $x_T$ according to the fixed noise schedule $\beta_1, \cdots, \beta_T$.

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) := \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t I) \quad (4)$$

Efficient training is optimizing a random term of $t$ with stochastic gradient descent:

$$\mathcal{L}_\theta = \left\| \boldsymbol{\epsilon}_\theta \left( \alpha_t \mathbf{x}_0 + \sqrt{1-\alpha_t^2}\boldsymbol{\epsilon} \right) - \boldsymbol{\epsilon} \right\|_2^2 \quad (5)$$

**Reverse process** Unlike the diffusion process, reverse process is to recover samples from Gaussian noises. The reverse process is a Markov chain from $x_T$ to $x_0$ parameterized by shared $\theta$:

$$p_\theta(\boldsymbol{x}_0, \cdots, \boldsymbol{x}_{T-1}|x_T) = \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t), \quad (6)$$

where each iteration eliminates the Gaussian noise added in the diffusion process:

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) := \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \sigma_\theta(\boldsymbol{x}_t, t)^2 I) \quad (7)$$

## E Information Perturbation

XLSR-53 is pre-trained on 56k hours of speech in 53 languages, to provide linguistic information.

We apply the following functions (Qian et al., 2020; Choi et al., 2021) on acoustic features (i.e., pitch, and energy) to create acoustic-perturbed speech samples $\hat{S}$, while the linguistic content remains unchanged, including 1) formant shifting $fs$, 2) pitch randomization $pr$, and 3) random frequency shaping using a parametric equalizer $peq$.

- For $fs$, a formant shifting ratio is sampled uniformly from $\mathrm{Unif}(1, 1.4)$. After sampling the ratio, we again randomly decided whether to take the reciprocal of the sampled ratio or not.
- In $pr$, a pitch shift ratio and pitch range ratio are sampled uniformly from $\mathrm{Unif}(1, 2)$ and $\mathrm{Unif}(1, 1.5)$, respectively. Again, we randomly decide whether to take the reciprocal of the sampled ratios or not. For more details for formant shifting and pitch randomization, please refer to Parselmouth `https://github.com/YannickJadoul/Parselmouth`.
- $peq$ represents a serial composition of low-shelving, peaking, and high-shelving filters. We use one low-shelving HLS, one high-shelving HHS, and eight peaking filters HPeak.

## F Evaluation

### F.1 Subjective Evaluation

For MOS tests, the testers present and rate the samples, and each tester is asked to evaluate the subjective naturalness on a 1-5 Likert scale. For CMOS, listeners are asked to compare pairs of audio generated by systems A and B and indicate which of the two audio they prefer, and choose one of the following scores: 0 indicating no difference, 1 indicating a small difference, 2 indicating a large difference and 3 indicating a very large difference.

For quality evaluation, we explicitly instruct the raters to "*(focus on examining the audio quality and naturalness, and ignore the differences of style (timbre, emotion and prosody).)*". For prosody evaluation, we explicitly instruct the raters to "*(focus on the naturalness of the prosody and style, and ignore the differences of content, grammar, or audio quality.)*".

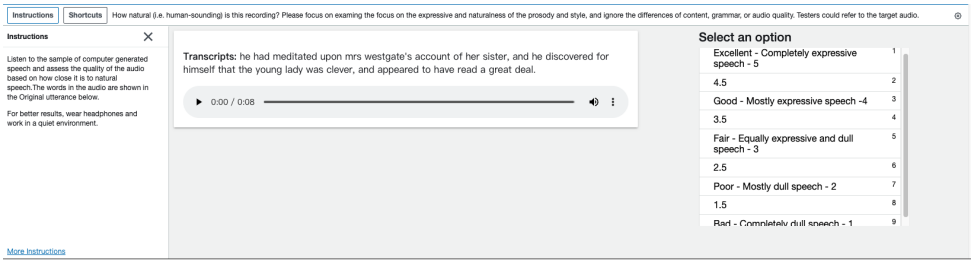
Our subjective evaluation tests are crowdsourced and conducted by 25 native speakers via Amazon Mechanical Turk. The screenshots of instructions for testers have been shown in Figure 4. We paid $8 to participants hourly and totally spent about $800 on participant compensation. A small subset of speech samples used in the test is available at `https://Prosody-TTS.github.io/`.

### F.2 Objective Evaluation

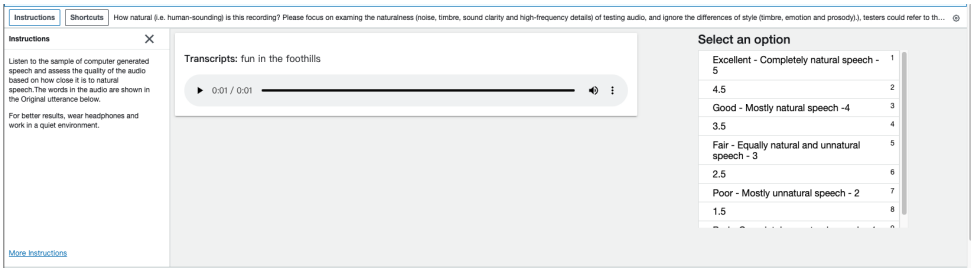
Mel-cepstral distortion (MCD) (Kubichek, 1993) measures the spectral distance between the synthesized and reference mel-spectrum features.

F0 Frame Error (FFE) combines voicing decision error and F0 error metrics to capture F0 information.

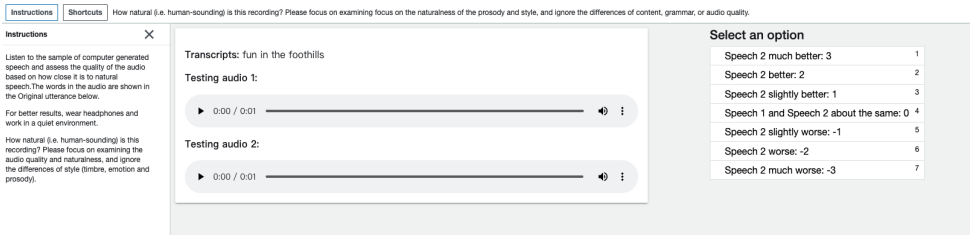
Number of Statistically-Different Bins (NDB) and Jensen-Shannon divergence (JSD) (Richardson and Weiss, 2018). They measure diversity by 1) clustering the training data into several clusters, and 2) measuring how well the generated samples fit into those clusters.



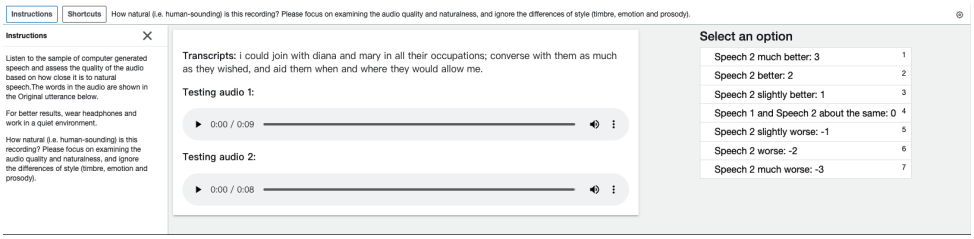

(a) Screenshot of MOS-P testing.




(b) Screenshot of MOS-Q testing.




(c) Screenshot of CMOS-P testing.




(d) Screenshot of CMOS-Q testing.

Figure 4: Screenshots of subjective evaluations.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*See section 6*

☑ A2. Did you discuss any potential risks of your work?
*See section 7*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*See section 1*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☒ Did you use or create scientific artifacts?

*Left blank.*

☐ B1. Did you cite the creators of artifacts you used?
*No response.*

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*No response.*

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*No response.*

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*No response.*

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*No response.*

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*No response.*

### C ☑ Did you run computational experiments?

*See section 4*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*See section 4.1*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*See section 4.1*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*See section 4.1*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*Left blank.*

**D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*See section 4.1 and Appendix F*

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*See section 4.1 and Appendix F*

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*See section 4.1 and Appendix F*

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*Not applicable. Left blank.*

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*Not applicable. Left blank.*

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*Not applicable. Left blank.*