SpeechTripleNet: End-to-End Disentangled Speech Representation Learning for Content, Timbre and Prosody

Hui Lu
luhui@se.cuhk.edu.hk
The Chinese University of Hong Kong
Hong Kong SAR, China

Xixin Wu[*]
wuxx@se.cuhk.edu.hk
The Chinese University of Hong Kong
Hong Kong SAR, China

Zhiyong Wu
zywu@sz.tsinghua.edu.cn
Tsinghua University
Shenzhen, China

Helen Meng
hmmeng@se.cuhk.edu.hk
The Chinese University of Hong Kong
Hong Kong SAR, China

ABSTRACT

Disentangled speech representation learning aims to separate different factors of variation from speech into disjoint representations. This paper focuses on disentangling speech into representations for three factors: spoken content, speaker timbre, and speech prosody. Many previous methods for speech disentanglement have focused on separating spoken content and speaker timbre. However, the lack of explicit modeling of prosodic information leads to degraded speech generation performance and uncontrollable prosody leakage into content and/or speaker representations. While some recent methods have utilized explicit speaker labels or pre-trained models to facilitate triple-factor disentanglement, there are no end-to-end methods to simultaneously disentangle three factors using only unsupervised or self-supervised learning objectives. This paper introduces SpeechTripleNet, an end-to-end method to disentangle speech into representations for content, timbre, and prosody. Based on VAE, SpeechTripleNet restricts the structures of the latent variables and the amount of information captured in them to induce disentanglement. It is a pure unsupervised/self-supervised learning method that only requires speech data and no additional labels. Our qualitative and quantitative results demonstrate that SpeechTripleNet is effective in achieving triple-factor speech disentanglement, as well as controllable speech editing concerning different factors.

CCS CONCEPTS

• **Computing methodologies → Learning latent representations**; *Latent variable models*; Learning paradigms.

KEYWORDS

Speech disentanglement, unsupervised representation learning, prosody modeling, VAE

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0108-5/23/10.
https://doi.org/10.1145/3581783.3612485

ACM Reference Format:
Hui Lu, Xixin Wu, Zhiyong Wu, and Helen Meng. 2023. SpeechTripleNet: End-to-End Disentangled Speech Representation Learning for Content, Timbre and Prosody. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3612485

1 INTRODUCTION

Speech is the most natural medium of communication for humans. Linguists view speech communication as involving three types of behavior: linguistic, extralinguistic, and paralinguistic [12]. Linguistic behavior involves transmitting spoken content that is encoded by the units of the corresponding language. The extralinguistic aspect of speech communication identifies individual speakers and includes information such as voice quality, overall pitch range, and loudness. Paralinguistic behavior incorporates the speaker's affective, attitudinal, or emotional state. The information being conveyed through linguistic, extralinguistic, and paralinguistic behaviors corresponds, respectively, to spoken content, speaker timbre, and speech prosody. These are three independent factors of variation in speech. Therefore, our goal of separating speech into content, timbre, and prosody representations is well-founded.

In general, disentangled speech representations learning can benefit many downstream tasks. Disentangled representations provide a feature that is invariant to irrelevant factors for the task, making them useful for discriminative tasks like automatic speech recognition (ASR) and speaker recognition. Moreover, disentangled representations enable us to modify the factors of a given speech independently, which is useful for generative tasks such as speech editing, including speaker identity conversion and prosody modification.

Previous unsupervised speech disentanglement methods have primarily focused on separating speech into content and timbre representations. To achieve this, various restrictions have been imposed on the representation learning of content and speaker to ensure they are decomposed. FHVAE [6], a variational auto-encoder (VAE) [9] designed for sequential data, incorporates different temporal structures for the prior distributions of content and timbre to induce their disentanglement. Additionally, FHVAE introduces a contrastive learning objective to aid in learning the discriminative speaker representation. Numerous advances have been made in one-shot voice conversion, which aims to convert the speaker

[*] Corresponding author

identity of source speech into that of target speech, thus requiring a good disentanglement of content and speaker representations from speech. AdINVC [3] uses instance normalization [17] to normalize speech into speaker-independent content representation, taking the normalization statistics as the speaker representation. AutoVC [15] downsamples the content representation to remove speaker information, while the speaker representation is obtained through a pre-trained speaker verification model. VQVC [22] and VQVC+ [21] use vector quantization to eliminate speaker information and obtain the content representation, with the quantization residual as the speaker representation. In addition to using vector quantization, VQMIVC [18] further incorporates mutual information minimization to encourage disentanglement. β-VAEVC [13] imposes separate weight parameters on the KL divergence terms with respect to content and speaker representations to restrict the amount of information captured by them. This is similar to the idea proposed in β-VAE [5]. Through careful tuning of the two weight parameters, β-VAEVC can restrict the two representations to capture content and speaker information, respectively.

However, the lack of explicit modeling of prosody in the aforementioned content-speaker disentanglement methods could result in unintentional leakage of prosody information into content and/or speaker representation. This can cause unexpected behavior for downstream tasks, especially for speech editing, where undesired prosody changes may occur when we aim to modify only timbre or content.

Several recent works propose to further disentangle prosody from speech and separate it from content and timbre representations. The key is to prevent prosody information from being captured by other representations. SpeechSplit [14] and SpeechSplit 2.0 [2] adopt random resampling and signal processing methods to explicitly corrupt the prosody information before extracting content representation. However, they adopt speaker identity labels as the speaker representation instead of learning it jointly with representations of other factors, making them not purely unsupervised learning methods. Another method utilizes a separate model for learning prosody representations in a self-supervised learning manner [19]. By using signal processing-based data augmentation such as pitch stretch and volume adjustment, the model is trained to predict the permutation strength to learn representations for pitch and energy. The well-trained prosody model is then fixed as a prosody representations extractor equipped to another model to facilitate the disentanglement of content and speaker representations. In this way, the disentanglement of prosody, content, and speaker representations is separated into two phases.

For prosody modeling, SpeechSplit (2.0) [2, 14] model pitch and rhythm as sequential features in the prosody representation. However, manipulating these two features in SpeechSplit (2.0) typically requires a reference speech with a similar phoneme sequence; otherwise, editing pitch and/or rhythm can be problematic. On the other hand, [19] models prosody as global features of an utterance, capturing the overall trends of pitch and energy in two separate vectors. This approach allows for the use of a random reference utterance to provide overall pitch and energy information for modifying these two factors. The disadvantage of the global prosody representation is that it makes it impossible to manipulate prosody locally.

We propose SpeechTripleNet, an end-to-end speech disentanglement method that learns disjoint representations for three generative factors underlying speech: spoken content, speaker timbre, and prosody. Unlike previous methods, SpeechTripleNet does not require speaker identity labels or a pre-trained representation extractor. Instead, it disentangles all three factors end-to-end after one pass of training. SpeechTripleNet achieves triple-factor disentanglement through the design of latent structures, incorporation of easy-to-extract self-supervision features, utilization of information-restricting learning objectives, and VAE's modeling power for latent variables. We utilize a VAE to separate speech into three latent variables and design the latent structures to be suitable for capturing different factors. We incorporate the pitch and energy features and process them into the appropriate self-supervision signals for inducing better prosody representation learning. We further adopt channel capacity restrictions to all three latent variables to limit the amount of information captured and ensure their independence and disentanglement.

SpeechTripleNet provides a novel method for prosody modeling. We represent prosody as sequential features that are quantitatively aligned with pitch and energy. This allows us to directly modify speech pitch and energy by manipulating the prosody representation. This eliminates the need for a reference speech during prosody editing and avoids any mismatching issues between the reference and original speech.

SpeechTripleNet can achieve one-shot voice conversion as many previous methods do. Besides, thanks to the interpretable structure of the prosody representation, SpeechTripleNet can also achieve finegrained prosody editing. For example, we can modify an utterance to emphasize or de-emphasize a certain word, or turn an utterance from a statement into a question. And We believe that with more expert knowledge about prosody, we can realize more applications regarding prosody modification, given the flexibility of the prosody representation. We can achieve editing of timbre and prosody at the same time since we have the disentangled representations of them. Our code and demo are available here[1].

2 RELATED WORKS

2.1 VAE-based disentanglement

Since the proposed SpeechTripleNet is based on VAE, it is closely related to VAE-based disentangled representation learning methods [1, 4, 13]. SpeechTripleNet is similar to β-VAEVC [13] which proposes restricting the information captured by content and speaker representations. β-VAEVC achieves this by imposing two weight parameters on the two KL divergence terms concerning the content and speaker representations. Although we also restrict the KL divergence terms to constrain the amount of information captured by each latent variable, we set the channel capacity for each latent variable instead of imposing weight parameters, which has been proven to yield better sampling quality in general [1]. SpeechTripleNet further disentangles the prosody factor from speech, which is not achieved in β-VAEVC. SpeechTripleNet is related to AnnealVAE [1] and JointVAE [4]. These two methods also set the channel capacity for different latent variables to induce disentanglement. However,

[1] Code and demo: https://github.com/light1726/SpeechTripleNet/

AnnealVAE only disentangles simple image data into a dimension-wise disentangled vector, while JointVAE disentangles simple image data into continuous and discrete vectors. Our method scales the channel capacity restriction method to more complex speech disentanglement while adopting self-supervision to facilitate prosody representation learning.

2.2 Prosody modeling

SpeechTripleNet proposes disentangling prosody representation from speech, which is related to many speech generation methods with prosody modeling. Some popular text-to-speech synthesis (TTS) methods [20, 24] model the prosody as a single vector, which is not suitable for fine-grained locally prosody manipulation. Our method is related to Fastspeech2 [16], which predicts pitch and energy as prosodic features to enable prosody control for TTS. The main difference is that SpeechTripleNet adopts a channel capacity restriction method to avoid the other information being leaked into the prosody representation, which is not a common concern in TTS methods.

3 APPROACH

3.1 Overview

Suppose we have a speech dataset $\mathcal{X} = \{X^i \in \mathbb{R}^{T_i \times D_x} | i = 1, 2, ..., N\}$ consisting of N speech utterances from various speakers and of varying spoken content. To avoid processing much longer speech waveforms, we deal directly with the time-frequency form of speech, such as the Mel-spectrogram. We denote the i-th utterance as X^i, with time length (i.e., number of frames) T_i and dimension D_x. We aim to build a model that disentangles a speech utterance X into three latent representations for three generative factors: spoken content, speaker timbre, and speech prosody, denoted as Z_c, Z_s, and Z_p, respectively. We may also refer to these three latent variables as content latent, speaker latent, and prosody latent. We model the relationship between the speech utterance X and its underlying latent variables with a variant of VAE, which we refer to as SpeechTripleNet.

The overall architecture of SpeechTripleNet is shown in Figure 1, which consists of four neural network modules: the content encoder, speaker encoder, prosody encoder, and decoder. We use the three encoders to model the posterior distributions of content, speaker, and prosody representations, which are respectively denoted as $q_\phi(Z_c|X)$, $q_\phi(Z_s|X)$, and $q_\phi(Z_p|X)$, where ϕ represents the trainable parameters involved in the encoders. In practice, an encoder predicts a set of distribution parameters that define the posterior distribution of the corresponding latent variable. The decoder models the conditional distribution $p_\theta(X|Z_c, Z_s, Z_p)$, where θ denotes the trainable parameters in the decoder. It defines the generation process of speech from the three latent variables. We denote the prior distributions for the three latent representations respectively as $p(Z_c)$, $p(Z_s)$, and $p(Z_p)$, which are referred to as content prior, speaker prior, and prosody prior, respectively.

3.2 Latent structures

Intuitively, each latent variable operates on a specific manifold that is not observable. To define the structure of a latent variable, we specify three aspects: the form of the representation, posterior distribution, and prior distribution. The structure of a latent variable defines a distributional space or manifold in which the latent variable should lie, serving as an initial bottleneck for information flowing from it. In this sense, the structures of latent variables are essential for inducing speech disentanglement.

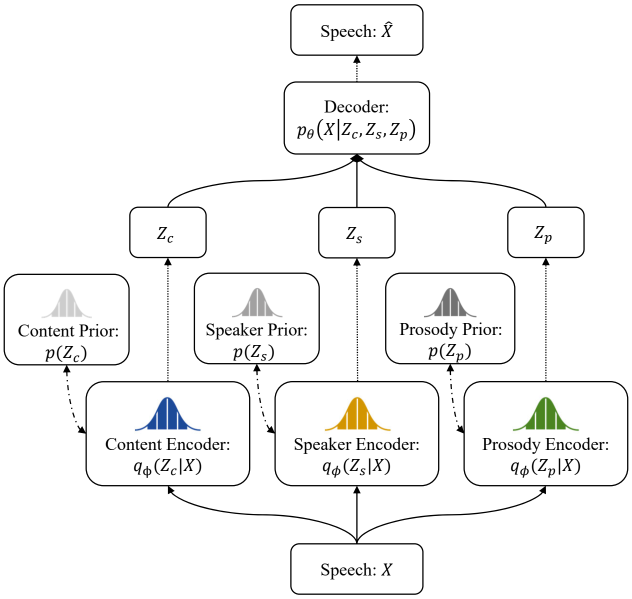
Speech: $\hat{X}$
Decoder: $p_\theta(X|Z_c, Z_s, Z_p)$
Z_c
Z_s
Z_p
Content Prior: $p(Z_c)$
Speaker Prior: $p(Z_s)$
Prosody Prior: $p(Z_p)$
Content Encoder: $q_\phi(Z_c|X)$
Speaker Encoder: $q_\phi(Z_s|X)$
Prosody Encoder: $q_\phi(Z_p|X)$
Speech: X

Figure 1: Model architecture

Content Latent: The spoken content in speech includes information on pronunciation variations across an utterance. To capture the time-variant pronunciation variation, we define the content representation as a sequence of vectors with the same time resolution as the original speech. For a speech utterance $X \in \mathbb{R}^{T \times D_x}$, we have $Z_C \in \mathbb{R}^{T \times D_c}$. We define the prior distribution of Z_C to be a frame-wise multidimensional normal distribution such that $p(Z_c^{(i)}) = \mathcal{N}(Z_c^{(i)}|\mathbf{0}, \mathbf{I})$, $i = 1, ..., T_i$. Likewise, we define the content posterior distribution to be a frame-wise multi-dimensional isotropic Gaussian distribution, such that $q_\phi(Z_c^{(i)}|X) = \mathcal{N}(Z_c^{(i)}|\mu_c(X;\phi), \sigma_c^2(X;\phi))$. Here, i is the frame index, and $i = 1, 2, ..., T$. $\mu_c(\cdot;\phi)$ and $\sigma_c^2(\cdot;\phi)$ respectively denote the mean and variance of the content representation produced by the content encoder.

Speaker latent: In contrast to spoken content, speaker identity remains constant throughout an utterance. We define the speaker representation as a single vector that summarizes the overall speaker information in the utterance and is shared across different time steps. Specifically, we have the speaker representation $Z_S \in \mathbb{R}^{D_s}$ which is a continuous vector, where D_s is the dimension of the speaker representation. We define the prior distribution of the speaker representation as a multi-dimensional normal distribution, $p_S(Z_s) = \mathcal{N}(Z_s|\mathbf{0}, \mathbf{I})$. Thus, we define the speaker posterior distribution as a multi-dimensional isotropic Gaussian distribution, denoted as $q_\phi(Z_s|X) = \mathcal{N}(Z_s^{(i)}|\mu_s(X;\phi), \sigma_s^2(X;\phi))$. Here, $\mu_s(\cdot;\phi)$ and $\sigma_s^2(\cdot;\phi)$ represent the mean and variance, respectively, of the speaker representation predicted by the speaker encoder.

Prosody latent: While some methods [19] model prosody as a global feature of a speech utterance, we propose to model it as a time-variant sequence of vectors. This approach is preferable for several reasons. Firstly, speech prosody is comprised of suprasegmental features such as intonation, intensity, and rhythm, which manifest locally within a speech utterance. While humans define abstract, utterance-level prosodic characteristics such as emotion and style, they can be modified by varying the three prosodic features at the lower level. Secondly, the generally available speech corpora are not explicitly designed for expressiveness modeling, in which global prosody variation is less prominent than local variation. For this reason, there is insufficient support data for modeling prosody as a global feature of an utterance. Furthermore, modeling prosody as a global feature of a speech utterance blocks the possibility of manipulating local prosody.

To this end, we define the prosody representation as a time-variant representation with the same time granularity as the speech. To make it easier for explicit prosody manipulation and to bottleneck other information, we define the prosody representation as a sequence of discrete random variables. We then set the prosody prior distribution of each speech frame as a uniform categorical distribution, denoted as $p_p(Z_p^{(i)}|X) = \text{Cat}(M_p; 1/M_p, ..., 1/M_p)$; we accordingly define the prosody posterior distribution as $q_\phi(Z_p^i|X)$ $= \text{Cat}(M_p; \text{softmax}(\pi(X;\phi)))$, where $\pi(\cdot;\phi)$ is the pre-normalized probability distribution of the i-th frame of prosody representation, M_p refers to the number of possible values of the prosody random variable.

3.3 Learning objective

The proposed model inherits the learning objective of VAEs while imposing additional regularizations to facilitate disentanglement. Assuming that the three latent variables are conditionally independent given the speech data, we can derive the vanilla evidence lower bound (ELBO) of the proposed speech VAE as shown in equation (1). The first term represents the reconstruction loss, while the remaining are the KL divergence terms with respect to the three latent variables.

$$\begin{aligned}\mathcal{L}_{\text{vanilla}} = &- \mathbb{E}_{X,q_\phi(Z_c,Z_s,Z_p|X)}\left[\log p_\theta(X|Z_c,Z_s,Z_p)\right]\\ &+ \mathbb{E}_X\left[D_{\text{KL}}\left[q_\phi(Z_c|X) \parallel p(Z_c)\right]\right]\\ &+ \mathbb{E}_X\left[D_{\text{KL}}\left[q_\phi(Z_s|X) \parallel p(Z_s)\right]\right]\\ &+ \mathbb{E}_X\left[D_{\text{KL}}\left[q_\phi(Z_p|X) \parallel p(Z_p)\right]\right] \qquad (1)\end{aligned}$$

Self-supervision: As previously mentioned, prosody representation can be broken down into intonation, intensity, and rhythm. The first two are closely related to two acoustic features: pitch and energy, which can be easily extracted from speech using off-the-shelf tools. Since pitch and energy are easily perceptible to humans, we aim to model the prosody representation to be explicitly related to these two acoustic features. This leads to better interpretability and fundamental controllability. We do not disentangle rhythm features in prosody representation as they are generally more challenging to extract explicitly and cannot be easily captured under unsupervised and self-supervised learning settings. While we set the content and prosody representations to have the same time resolution as the speech, we implicitly leave the rhythm information (i.e., the duration of each pronunciation unit) as is.

In this work, we propose inducing prosody representation learning using extracted pitch and energy features. The naive approach is to supervise the prosody encoder to predict pitch and energy as the prosody representation. However, these features contain information about both spoken content and speaker timbre, which can cause the content and speaker information to leak into the prosody representation if we use them directly. To mitigate this issue, we adopt min-max normalization to scale the voiced parts of the logarithm pitch contour into the range of [0, 1]. This operation significantly reduces the speaker information. Additionally, we quantize the scaled pitch contour into M_f bins to reduce its discriminativeness about the spoken content. We apply the same process to the energy feature and set the number of quantization bins to be M_e. We refer to the quantized pitch contour and energy as y_f and y_e, respectively.

The preprocessing of pitch and energy aligns with the structure of the prosody latent, whose posterior and prior distributions are frame-wise categorical distributions. Since we expect the prosody representation to explicitly capture pitch and energy for easier controllability, we set a learning objective that encourages the posterior distribution of the prosody latent to be close to the joint distribution of y_p and y_e. The loss function is shown in equation (2), where $M_p = M_f \times M_e$ is the number of classes of the prosody representation. $p(y_f, y_e)$ denotes the joint distribution of y_f and y_e, practically represented by one-hot vectors of M_p dimensions with each entry represents one possible combination of pitch and energy values. We model y_f and y_e jointly because they are initially dependent, which can be easily verified on quantized pitch and energy. Intuitively, $p(y_f = M_f - 1, y_e = 0)$ should be close to zero, since high pitch typically incurs a large volume of speech, while marginally both $y_f = M_f - 1$ and $y_e = 0$ are quite common. Another benefit of modeling the joint distribution is that it involves tuning only one weight parameter instead of two for two separate loss terms.

$$\mathcal{L}_p = D_{\text{KL}}[q_\phi(Z_p|X) \parallel p(y_p, y_e)] \qquad (2)$$

Channel capacity restrictions: When SpeechTripleNet defines three latent variables, it sets up three channels for speech information to flow through. It is important to ensure that content, timbre, and prosody flow through their respective channels exclusively to achieve speech disentanglement. So far, we have defined the latent structures for three latent variables and implemented pitch-and-energy-aware self-supervision for the prosody latent. The defined latent structures make it easier for each channel to capture the desired factor, while the self-supervision informs the prosody representation to be explicitly aware of pitch and energy. Although these two inductive biases help each channel recognize the desirable factor, there is no guarantee that each channel will capture precisely the right factor, so long as it has redundant information capacity. Intuitively, each channel always tends to capture more information, regardless of from which factor, to further reduce the reconstruction loss term.

To address this issue, we must limit the amount of information captured by each channel, which can be achieved by restricting the channel capacity of each channel. It is proved [4] that we can factorize the KL divergence terms in equation (1) as shown in equation (3) :

$$\mathbb{E}_X \left[D_{\mathrm{KL}}[q_\phi(Z|X) \parallel p(Z)]\right] = \mathbb{I}(X,Z) + D_{\mathrm{KL}}[q_\phi(Z) \parallel p(Z)], \quad (3)$$

where Z can be any of Z_c, Z_s, or Z_p. $\mathbb{I}(X,Z)$ denotes the mutual information between speech X and the latent variable Z. $q_\phi(Z)$ is the marginal distribution of Z, defined as $\int_X p_X(X) q_\phi(Z|X) dX$. We can observe that $\mathbb{E}X[D_{\mathrm{KL}}[q_\phi(Z|X) \parallel p(Z)]]$ is an upper bound of $\mathbb{I}(X,Z)$, indicating that we can restrict the KL divergence terms in equation (1) in order to limit the amount of information captured by the respective latent variables.

Based on the results and derivation presented in previous works [1, 4], we set the channel capacities for the three latent variables (Z_c, Z_s, and Z_p) as hyperparameters C_c, C_s, and C_p, respectively. We adopt the learning objective shown in Equation (4), where γ_c, γ_s, and γ_p are also hyperparameters that weight different loss terms. C_c, C_s, and C_p represent the maximum amount of information of speech being captured by Z_c, Z_s, and Z_p, respectively. Ideally, C_c, C_s, and C_p should be approximately equal to the amounts of information in content, timbre, and prosody, respectively. To stabilize convergence, we gradually increase the channel capacities during training, as in previous works [1, 4].

$$\begin{aligned}\mathcal{L}_{\mathrm{Cap}} = &- \mathbb{E}_{X,q_\phi(Z_c,Z_s,Z_p|X)} \left[\log p_\theta(X|Z_c,Z_s,Z_p)\right] \\ &+ \mathbb{E}_X \left[\gamma_c \left|D_{\mathrm{KL}}[q_\phi(Z_c|X) \parallel p(Z_c)] - C_c\right|\right] \\ &+ \mathbb{E}_X \left[\gamma_s \left|D_{\mathrm{KL}}[q_\phi(Z_s|X) \parallel p(Z_s)] - C_s\right|\right] \\ &+ \mathbb{E}_X \left[\gamma_p \left|D_{\mathrm{KL}}[q_\phi(Z_p|X) \parallel p(Z_p)] - C_p\right|\right] \quad (4)\end{aligned}$$

One may notice that we still impose the channel capacity restriction on the prosody representation learning even though we have already introduced a self-supervised learning objective to it. The reasons are two-fold. Firstly, most linearly normalization procedures (including ours introduced in Section 3.3) cannot entirely eliminate the speaker variation from the prosody (especially pitch); one can still notice the difference between the processed pitch for male and female speakers. Secondly, as long as there is extra information capacity in the prosody representation, it can still encode information other than the prosody to decrease the reconstruction loss, since the prediction of pitch and energy cannot be 100% accurate. To this end, the capacity restriction imposed on the prosody representation penalizes possible information leakage. The overall learning objective is shown in equation 5, where γ_{fe} is a hyper-parameter balancing $\mathcal{L}_p$ against other loss terms.

$$\mathcal{L}_{overall} = \mathcal{L}_{\mathrm{Cap}} + \gamma_{fe}\mathcal{L}_p \quad (5)$$

4 IMPLEMENTATION

4.1 Features

To facilitate modeling, we convert speech into its time-frequency representation. Specifically, we transform each speech waveform into its corresponding Mel-spectrogram. The Fast Fourier Transformation (FFT) window size is set to 1024, and the window shift to 256. The number of Mel bins is set to 80. In this way, for a one-second speech waveform with 22,050 samples, we obtain the corresponding Mel-spectrogram matrix with 86 frames and 80 dimensions.

We use pitch and energy contours as self-supervision signals to learn prosody representation. The window length and shift used to extract these features are the same as those for extracting the Mel-spectrogram. We process the pitch and energy contours as described in Section 3, and set the number of quantization bins for both features to 8. This results in a joint one-hot distribution with a dimension of 64.

4.2 Model structure

Content Encoder: The content encoder aims to model the posterior distribution of the content representation given the speech. It consists of a fully-connected layer that projects the Mel-spectrogram into 256 dimensions. Then, we use two layers of 1D convolution, each with a kernel size of 3 and stride of 1. A batch normalization layer follows each convolution layer. Two self-attention layers, each with feed-forward neural networks, are stacked upon the convolutional layers to obtain a more contextualized representation. The output of the self-attention layers is then projected into frame-wise mean and logarithm standard deviation vectors, each with a dimension of 128. We use the re-parameterization trick to sample from the content posterior distribution, resulting in a sequence of 128-dimensional vectors.

Speaker encoder: The speaker encoder is composed of four layers of 1D convolution with kernel sizes of 3, 3, 5, and 5. Each convolutional layer is followed by a temporal pooling layer that downsamples the sequence by a factor of 2. In total, the sequence is downsampled by a factor of 16. A temporally global pooling layer is used to pool the output of the last convolution layer into a single vector. This vector is then projected into the mean and logarithm vectors of the distribution parameters of the speaker posterior distribution. The dimension of the speaker posterior distribution is set to 128. The reparameterization trick is used to sample from the speaker posterior distribution, which yields a 128-dimension speaker latent vector.

Prosody encoder: The prosody encoder has the same structure as the content encoder, except that the last layer's dimension is set to 64, which corresponds to the joint distribution of pitch and energy. This results in a frame-wise 64-dimensional logit, which is normalized to frame-wise probability using softmax. During training, we use Gumbel-Softmax [7] to sample a sequence of discrete codes as the prosody representation. We can easily obtain the marginal discrete codes for pitch and energy, which are then separately embedded into 8-dimension vectors. The concatenation of the pitch and energy embeddings is adopted as the embedding for the prosody.

Decoder: To form the input feature to the decoder, we concatenate three latent representations on the feature axis. The speaker latent vector is repeated to the same length as the content and prosody representations. The decoder has the same structure as the content encoder, except for the input and output dimensions. The input dimension is set to the sum of the three latent representations, which is 272. The output is the predicted Mel-spectrogram with a dimension of 80.

4.3 Training specifics

We compute the reconstruction loss in equation (4) as the log-likelihood of the predicted Mel-spectrogram. We assume that the distribution of the Mel-spectrogram is a multi-dimensional normal distribution with the ground-truth Mel-spectrogram as the mean and unit variance. The log-likelihood is calculated frame by frame, which is normalized over the number of frames. Similarly, for sequential content and prosody representations, we compute their KL divergence terms in equation 4 frame-wise before being normalized by the number of frames.

We train the model on a single NVIDIA Tesla V100 GPU card. We use Adam optimizer [8] with a fixed learning rate of 10^{-4}, for which β_1 and β_2 are set respectively as 0.9 and 0.98. For weight parameters in the learning objective, we set $\gamma_c = 100$, $\gamma_s = \gamma_p = 10$. We set the maximum channel capacities for content, speaker and prosody as 1.3, 60 and 3, respectively. We increase the channel capacity for each latent variable from zero to its maximum values in the first 20,000 training steps. We set the batch size to 64 and train the model for 200k steps.

4.4 Speech editing

SpeechTripleNet can disentangle speech into content, timbre, and prosody representations that are aware of pitch and energy. This allows for three types of speech editing: speaker identity transformation, pitch modification, and energy modification. To transform the speaker identity of a given source utterance X, we select a reference utterance X' with a different speaker identity. We then feed X into the SpeechTripleNet encoders to disentangle it into its three representations: content (Z_c), timbre (Z_s), and prosody (Z_p). We extract the speaker representation Z'_s from the reference speech X' using the speaker encoder of SpeechTripleNet. To achieve speaker identity transformation, we use the SpeechTripleNet decoder to generate speech from Z_c, Z'_s, and Z_p. This results in the converted speech X_s, where $X_s \sim p_\theta(\cdot|Z_c, Z'_s, Z_p)$.

To modify the prosody of a speech utterance locally, we first need to locate the position of the word or syllable that we want to modify. This can be done by using force-alignment tools to align the text with speech, or by observation. Once we have identified the position in speech X, we marginalize the prosody posterior distribution $q_\phi(z_p|X)$ over the pitch and energy axis to respectively obtain the pitch and energy distributions $q_\phi(z^f|x)$ and $q_\phi(z^e|x)$. We can then independently modify these pitch and energy distributions (e.g., by increasing or decreasing one or more bins) before combining them into a single prosody representation. We denote the prosody representation with modified pitch and energy as Z_p^f and Z_p^e, respectively. To generate the modified speech with modified pitch or energy, we decode the speech from the latent variables Z_c, Z_s, and either Z_p^f or Z_p^e. We denote the speech with modified pitch and energy respectively as X_f and X_e. That is, we have $X_f \sim p_\theta(\cdot|Z_c, Z_s, Z_p^f)$ and $X_e \sim p_\theta(\cdot|Z_c, Z_s, Z_p^e)$. It is important to note that while pitch and energy can be modified independently, we still need to ensure that the modification of one does not contradict the other's value. For example, increasing pitch values where energy values are low can result in unnatural-sounding speech.

Prosody modification has many applications. It is especially useful for emphasizing a word that wasn't emphasized in the original speech. This can be achieved by increasing both the pitch and energy of the word. Conversely, a word can be de-emphasized by decreasing its pitch and energy. Additionally, we find that increasing the pitch and energy at the end of an utterance can turn it from a statement into a question.

5 EXPERIMENTS

5.1 Dataset

We used the multi-speaker speech corpus VCTK [23] to evaluate SpeechTripleNet's ability to disentangle triple factors and achieve speech editing. VCTK contains speech data from 110 speakers, each with around 400 utterances. For training, we used data from 88 speakers, and for validation, 8 speakers were used. The remaining 11 speakers were used for testing. The VCTK dataset has variations in speaker identity, content, and local prosody, which support the modeling of content, timbre, and local prosody distributions.

5.2 Speech disentanglement evaluation

An ideal speech disentanglement produces separate representations that capture different factors of variation underlying the speech. The evaluation of disentanglement examines whether each separate representation captures the correct factor and whether different representations are independent. To achieve this, we manipulate different factors in the latent representations and verify whether the desired factor varies while other factors remain constant in the generated speech.

In this evaluation, we utilize the entire test set, which includes speech data from 11 speakers. For each utterance in the test set, we perform three types of speech editing as described in section 4.4: speaker identity conversion, pitch modification, and energy modification. To convert the speaker identity, we randomly select a target speech to provide the target speaker representation. For pitch modification, we adjust the pitch latent variables of each voiced segment in the utterance by randomly increasing or decreasing them. We apply a similar process for energy modification.

We first illustrate the possible speech editing results for an utterance in the test set in Figure 2. Through visualization of the Mel-spectrograms, pitch and energy contours, we demonstrate the speech editing and disentanglement of different factors qualitatively. The first column shows the pitch latent variables, the second column visualizes the energy latent variables, and the third column displays the generated speech from the latent variables shown in the same row. We omit the visualization of the speaker representation and content representation as they are not easy to read. We visualize the modified speech samples with their Mel-spectrograms, pitch (in red), and energy (in yellow) contours. The first row shows the latent pitch and energy latent variables as well as the Mel-spectrogram of the original speech. For each row from the second to fifth, we modify one or two latent variables and compare the generated speech Mel-spectrogram and the pitch and energy contours with the original speech.

In the second row, we replace the original female speaker representation with one from a male speaker, and one can notice a change in the formants of the Mel-spectrogram. However, the pitch

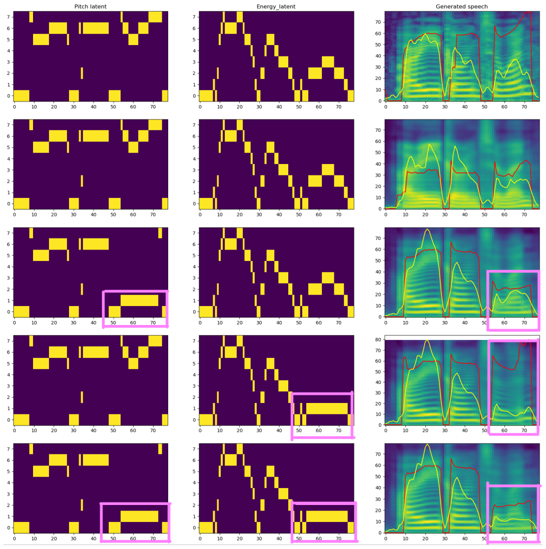

Figure 2: Speech editing visualization

and energy contours remain the same trend as the original speech. In the third row, we modify the pitch latent with respect to the third syllable into the lowest value for voiced speech segments (i.e., 1). One can notice a decrease of the pitch in the modified speech for the third syllable. The fourth row decreases only the energy latent variable concerning the third syllable into 1, resulting in a modified speech with the energy for the third syllable being significantly lower than that of the original speech. The fifth row combines the modifications from the third and fourth rows, which reduces both the pitch and energy latent variables into 1. By comparing the pitch and energy contours in the first and fifth rows, one can observe that both contours decrease on the third syllable as intended. In addition, one may notice the decrease of energy in the third row when we only decrease the pitch latent variable. This is caused by the dependence between the pitch and energy factors.

We then demonstrate the effectiveness of speech disentanglement quantitatively. We denote the original speech as X, and the three edited samples with respect to speaker identity, pitch, and energy as X_s, X_f, and X_e, respectively. We represent the reference speech providing the target speaker identity as X'. Ideally, all three sorts of edited samples should have the same content as X. Furthermore, X_s should contain the same prosody as X and a different speaker identity. Similarly, X_f should contain the same speaker identity as X, while showing a different pitch contour. We expect X_e to have a different energy pattern than X, while other factors remain the same.

We compare different factors of the original speech with those of the edited version using quantitative metrics. For the content factor, we adopt a pre-trained ASR model [11] to transcribe all the edited speech and report the transcription character error rate (ASR-CER). For speaker identification, we utilize a pre-trained speaker verification model[2] to embed speech into a speaker embedding and compute the cosine similarity between the edited speech and the original one, denoted as "CS with X". We also report the cosine similarity between the edited speech and the target speech providing the speaker identity, denoted as "CS with X'". To evaluate pitch and energy, we extract them using signal processing tools and compute the Pearson correlation coefficient (PCC) between the features extracted from the original speech and the edited one.

The results are presented in Table 1, which also shows the evaluation conducted on the reconstructed speech $\hat{X}$. We observe that the ASR-WER does not vary much among the different variants of edited speech, indicating good preservation of the content factor when modifying other factors. Regarding the speaker factor, X_s exhibits the worst speaker similarity with X, while showing very good speaker similarity with the target speech X', indicating successful modification of the speaker factor. Modifying either the pitch (X_f) or energy (X_e) factors results in far less change in speaker identity. In terms of pitch and energy factors, we notice that X_f differs the most from X in terms of pitch PCC, while X_e displays the most significant discrepancy with the source speech in terms of energy PCC. Due to the dependence between pitch and energy, we observe that the energy PCC also degrades when we modify only the pitch factor in X_f. Similarly, we can see the pitch change comprared to the original speech when we only modify the energy factor in X_e. X_s shows far less changes in prosody compared to other variants. These results suggest that the manipulation of latent space is consistent with factor variation in the generated speech, indicating that SpeechTripleNet learns well-disentangled speaker and prosody representations.

Table 1: Factor similarity with source speech X

	$\hat{X}$	X_s	X_f	X_e
ASR-CER	12.32%	13.62%	14.33%	14.97%
CS with X	0.7518	**0.6096**	0.7317	0.7155
CS with X'	0.5026	**0.7453**	0.4822	0.4907
Pitch PCC	0.9126	0.8994	**0.8059**	0.8418
Energy PCC	0.9560	0.9335	0.9221	**0.6847**

5.3 Speech editing evaluation

SpeechTripleNet is capable of modifying speech to alter speaker identity, pitch, and energy. In this section, we evaluate the performance of speech editing using subjective evaluation. For speaker identity transformation, we randomly select 15 pairs of source and reference utterances from the test set and conduct the speaker identity transformation introduced in Section 4.4. We ask 16 users to listen to these samples and rate the naturalness and speaker similarity of each utterance on a five-point scale. We then compare SpeechTripleNet with two state-of-the-art one-shot voice conversion methods: VQMIVC [18] and β-VAEVC [13] with similar modeling constraints, i.e., without using pre-trained feature extractors or speaker identity labels. We use a pre-trained Hifi-GAN [10] vocoder to synthesize the speech waveform from the Mel-spectrogram. As a reference, we include the evaluation results of samples that are re-synthesized using the Hifi-GAN vocoder, denoted as "Copy-synthesis." The results, shown in Table 2 with a 95% confidence interval, indicate that while all three methods achieve

[2] Resemblyzer: https://github.com/resemble-ai/Resemblyzer

comparable performance on speech naturalness, SpeechTripleNet achieves the best speaker similarity. This demonstrates the effectiveness of SpeechTripleNet in achieving good voice conversion performance.

Table 2: Speaker identity conversion user study

Model	Naturalness	Similarity
Copy-synthesis	4.29±0.13	4.58±0.11
VQMIVC	3.64±0.14	3.38±0.11
β-VAEVC	3.62±0.11	3.55±0.13
SpeechTripleNet (ours)	3.63±0.13	**3.67±0.15**

To evaluate the effectiveness of SpeechTripleNet in prosody conversion, we randomly selected 30 utterances from the test set that initially did not have any emphasized words. For each utterance, we edited the prosody to emphasize one word by raising the pitch and energy to their highest levels. We then asked 10 users to listen to the edited utterances and identify the emphasized word without presenting the original utterances. We collected their answers and computed the accuracy of correctly recognizing our prosody modification. On average, 88% of the emphasized words were correctly identified by the users, indicating the effectiveness of SpeechTripleNet in achieving fine-grained local prosody modification.

5.4 Ablation study

In this ablation study, we aim to demonstrate the effect of setting the channel capacity restriction on learning disentangled speech representations. We demonstrate this by varying the channel capacity restriction C_p imposed on the prosody representation learning. As we find that the timbre information can be easily leaked into the pitch representation, we show through experiments that imposing proper C_p can avoid the timbre information being leaked into the pitch representation. We extract pitch representations from all speech utterances from the test set. We then conduct the speaker verification using the pitch representation averaged over the temporal axis and report the equal error rate (EER). Intuitively, the lower the EER score the more timbre information the pitch representation captures.

The results are shown in Table 3, where we vary C_p from 1.5 to 3.0. Note that as the channel capacity for the prosody representation increases, more and more speaker information is captured by the pitch representation, denoted by the decreasing EER along with the increasing C_p. As references, we also show the speaker verification performance using energy and speaker representations. In contrast to the pitch representation, we notice that the energy representation captures little speaker information even when the prosody channel capacity increases; this is because the preprocessing operation stated in section 3.2 can already largely remove the speaker information from the energy. While the channel capacity restriction for the speaker representation remains unchanged during this process, the amount of speaker information captured by the speaker representation does not change much when varying C_p. This indicates the effectiveness of the channel capacity restriction in limiting information captured by a latent representation and avoiding information leakage from other factors.

Table 3: Speaker verification using pitch representation

C_p	1.5	2.0	2.5	3.0
EER (Z_f)	0.3747	0.3477	0.3016	0.2666
EER (Z_p)	0.4551	0.4812	0.4630	0.5006
EER (Z_s)	0.0701	0.0696	0.0685	0.0677

6 CONCLUSION

This paper presents SpeechTripleNet, an end-to-end method for disentangling speech into representations for content, timbre, and prosody with only unsupervised and self-supervised learning objectives. SpeechTripleNet is a VAE that models the three factors in speech as three latent variables. SpeechTripleNet does not require any human labeling or pre-trained representation extractors. It is purely learned from speech and features that can be easily extracted from speech. SpeechTripleNet achieves speech disentanglement by designing the structures of the latent representations to be suitable for capturing the respective factors and imposing channel capacity restrictions to limit the amount of information each factor obtains. Qualitative and quantitative experiments demonstrate that SpeechTripleNet effectively disentangles speech with respect to content, timbre, and prosody. With the disentanglement, speech editing such as voice conversion and prosody modification is possible. Thanks to the self-supervision from the pitch and energy features, the prosody representation is fundamentally interpretable and controllable, which enables prosody modification such as emphasizing or de-emphasizing a word in an utterance or changing the tone of an utterance from statement to question.

7 LIMITATIONS

Although SpeechTripleNet effectively disentangles speech with respect to content, timbre, and prosody, we acknowledge that there is still room for improvement and exploration. Firstly, SpeechTripleNet requires proper channel capacity restrictions to limit the amount of information captured by each factor, which requires parameter tuning to find the best setting. Secondly, SpeechTripleNet learns a prosody representation that can be locally manipulated to achieve fine-grained control over pitch and energy. However, rhythm factor, an important part of prosody, is not disentangled since it involves learning the duration of each language unit, which is difficult without supervision. In future work, we will explore better ways to obtain channel capacities and further disentangle the rhythm factor.

ACKNOWLEDGMENTS

This research was supported by the Center for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission's InnoHK Scheme; and was also supported by National Natural Science Foundation of China (62076144) and Shenzhen Science and Technology Program (WDZC20220816140515001).

REFERENCES

[1] Christopher P. Burgess, Irina Higgins, Arka Pal, Loïc Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding disentangling in β-VAE. *CoRR* abs/1804.03599 (2018). arXiv:1804.03599 http://arxiv.org/abs/1804.03599

[2] Chak Ho Chan, Kaizhi Qian, Yang Zhang, and Mark Hasegawa-Johnson. 2022. Speechsplit2. 0: Unsupervised speech disentanglement for voice conversion without tuning autoencoder bottlenecks. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6332–6336.

[3] Ju-Chieh Chou and Hung-yi Lee. 2019. One-Shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, Gernot Kubin and Zdravko Kacic (Eds.). ISCA, 664–668.

[4] Emilien Dupont. 2018. Learning Disentangled Joint Continuous and Discrete Representations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 708–718.

[5] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[6] Wei-Ning Hsu, Yu Zhang, and James R. Glass. 2017. Unsupervised Learning of Disentangled and Interpretable Representations from Sequential Data. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 1878–1889.

[7] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rkE3y85ee

[8] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[9] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[10] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems* 33 (2020), 17022–17033.

[11] Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, et al. 2019. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577* (2019).

[12] John Laver. 1994. *The semiotic framework.* Cambridge University Press, 13–25. https://doi.org/10.1017/CBO9781139166621.003

[13] Hui Lu, Disong Wang, Xixin Wu, Zhiyong Wu, Xunying Liu, and Helen Meng. 2022. Disentangled Speech Representation Learning for One-Shot Cross-Lingual Voice Conversion Using Beta-VAE. In *IEEE Spoken Language Technology Workshop, SLT 2022, Doha, Qatar, January 9-12, 2023*. IEEE, 814–821. https://doi.org/10.1109/SLT54892.2023.10022787

[14] Kaizhi Qian, Yang Zhang, Shiyu Chang, David Cox, and Mark Hasegawa-Johnson. 2020. Unsupervised speech decomposition via triple information bottleneck. In *37th International Conference on Machine Learning, ICML 2020 (37th International Conference on Machine Learning, ICML 2020)*, Hal Daume and Aarti Singh (Eds.). International Machine Learning Society (IMLS), 7792–7802. Publisher Copyright: Copyright © 2020 by the Authors. All rights reserved.; 37th International Conference on Machine Learning, ICML 2020 ; Conference date: 13-07-2020 Through 18-07-2020.

[15] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. 2019. AutoVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 5210–5219.

[16] 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. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/f63f65b503e22cb970527f23c9ad7db1-Paper.pdf

[17] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022* (2016).

[18] Disong Wang, Liqun Deng, Yu Ting Yeung, Xiao Chen, Xunying Liu, and Helen Meng. 2021. VQMIVC: Vector Quantization and Mutual Information-Based Unsupervised Speech Representation Disentanglement for One-Shot Voice Conversion. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, Hynek Hermansky, Honza Cernocký, Lukás Burget, Lori Lamel, Odette Scharenborg, and Petr Motlícek (Eds.). ISCA, 1344–1348.

[19] Shijun Wang and Damian Borth. 2022. Zero-shot Voice Conversion via Self-supervised Prosody Representation Learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 01–08.

[20] 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*. PMLR, 5180–5189.

[21] Da-Yi Wu, Yen-Hao Chen, and Hung-yi Lee. 2020. VQVC+: One-Shot Voice Conversion by Vector Quantization and U-Net Architecture. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, Helen Meng, Bo Xu, and Thomas Fang Zheng (Eds.). ISCA, 4691–4695.

[22] Da-Yi Wu and Hung-Yi Lee. 2020. One-Shot Voice Conversion by Vector Quantization. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, 7734–7738. https://doi.org/10.1109/ICASSP40776.2020.9053854

[23] Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. 2019. CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92).

[24] Ya-Jie Zhang, Shifeng Pan, Lei He, and Zhen-Hua Ling. 2019. Learning Latent Representations for Style Control and Transfer in End-to-end Speech Synthesis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 6945–6949. https://doi.org/10.1109/ICASSP.2019.8683623