

A UNIFIED ONE-SHOT PROSODY AND SPEAKER CONVERSION SYSTEM WITH SELF-SUPERVISED DISCRETE SPEECH UNITS

Li-Wei Chen, Shinji Watanabe, Alexander Rudnicky

Language Technologies Institute, Carnegie Mellon University

ABSTRACT

We present a unified system to realize one-shot voice conversion (VC) on the pitch, rhythm, and speaker attributes. Existing works generally ignore the correlation between prosody and language content, leading to the degradation of naturalness in converted speech. Additionally, the lack of proper language features prevents these systems from accurately preserving language content after conversion. To address these issues, we devise a cascaded modular system leveraging self-supervised discrete speech units as language representation. These discrete units provide duration information essential for rhythm modeling. Our system first extracts utterance-level prosody and speaker representations from the raw waveform. Given the prosody representation, a prosody predictor estimates pitch, energy, and duration for each discrete unit in the utterance. A synthesizer further reconstructs speech based on the predicted prosody, speaker representation, and discrete units. Experiments show that our system outperforms previous approaches in naturalness, intelligibility, speaker transferability, and prosody transferability. Code and samples are publicly available.¹

Index Terms— voice conversion, one-shot, prosody transfer, disentangled speech representation, self-supervised representations

1. INTRODUCTION

Human speech carries different aspects of information, including prosody, speaker traits, and language content. The objective of voice conversion (VC) is to control individual speech attributes with language content unchanged. In this paper, we focus on the conversion of three main attributes: pitch-energy², speaker traits, and rhythm.

One-shot voice conversion is challenging as the model can only access source and target speech without speaker identities given. Existing works mostly learned a speaker encoder jointly to isolate speaker information from prosody and language content. AutoVC [1] attempted to disentangle speaker traits from language by a carefully designed autoencoder. To separate speaker timbre from prosody, works [2] provided pitch contours explicitly to the system. Several works [3, 4] further improved content separation by learning representations with vector quantization. Additionally, VQMVC [5] proposed to minimize mutual information between content, speaker, and pitch representations for better disentanglement. The above methods, however, focused largely on speaker conversion. In applications such as emotion style transfer, separate control for prosody is desirable. Several works built upon AutoVC attempted to control prosodic attributes of speech. AutoPST [6] modeled rhythm by similarity-based re-sampling. SpeechSplit [7, 8]

achieved rhythm and pitch conversion with multiple carefully designed autoencoders. Leveraging these works, SRDVC [9] presented a unified one-shot VC system that allows control over both prosody and speaker attributes.

Despite their success, we found that improvements could be made. Previous approaches often suffered from intelligibility degradation after conversion due to the lack of disentangled language representations. To address this, recent approaches began to explore self-supervised speech representation [10, 11] (S3R) as a source of language information. However, these works [12, 13] generally focus on continuous S3R and are limited to speaker conversion. In contrast, we explore the use of discrete self-supervised speech units on both speaker and prosody conversion. Compared to continuous S3R, these discrete units formed from clustering naturally encode duration via repeated tokens, which is crucial for rhythm modeling.

Furthermore, we adopt a different modeling approach for prosodic features. In SRDVC, the pitch representation is directly extracted from a given pitch contour without explicit access to language information. However, as prosody is correlated with language [14], it causes naturalness degradation of prosody-converted samples (see Section 4.2). Energy and duration, although important prosodic features, are also not explicitly modeled. To address these issues, we propose a cascaded modular system that leverages discrete speech units for language information. First, our system extracts prosody and speaker representations from the raw waveform. Given the prosody representation, a prosody predictor estimates the pitch, energy, and duration of each speech unit. In combination with the predicted prosody, a synthesizer reconstructs speech based on speaker representation and discrete units. Empirical results demonstrate that our system outperforms previous approaches in intelligibility, naturalness, speaker and prosody transferability.

2. METHOD

2.1. Problem Formulation

Figure 1 presents an overview of the system. We now describe its corresponding formulation and provide details for each component. We use the notation $(\cdot)_{i=1}^I$ to denote a sequence of length I and $\{\cdot\}_{i=1}^I$ to denote a set of I elements. Given a speech waveform $\mathbf{W} = (w_t \in \mathbb{R})_{t=1}^T$ and its log-scale mel-spectrogram $\mathbf{X} = (\mathbf{x}_n \in \mathbb{R}^{d_x})_{n=1}^N$, we aim to extract different speech attribute representations $\mathbf{a}_i \in \mathbb{R}^{d_a}$ from \mathbf{W} , and re-synthesis \mathbf{X} from \mathbf{a}_i . T and N are the lengths of the waveform and log mel-spectrogram. d_x is the number of mel-frequency bands, and d_a is a hyper-parameter we choose as the dimension of all \mathbf{a}_i . We focus on 3 attributes: $i \in \{p, r, s\}$ corresponding to pitch-energy (\mathbf{a}_p), rhythm (\mathbf{a}_r), and speaker (\mathbf{a}_s). We define speech units in \mathbf{W} as $\mathbf{U} = (u_k \in \mathcal{U})_{k=1}^K$, and corresponding duration as $\mathbf{L} = (l_k \in \mathbb{N})_{k=1}^K$ (l_k is the number of frames each u_k spans). K is the total number of units for a given utterance, and \mathcal{U} is

¹<https://github.com/b04901014/UUVC>

²Here we use the term “pitch-energy” to refer to only the pitch and energy variations, which is one aspect of prosody.

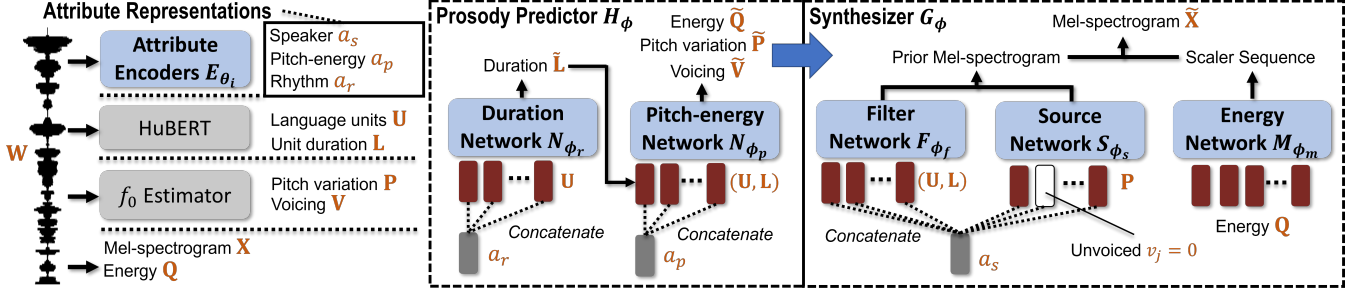


Fig. 1: Overview of our system. Attribute Encoders E_{θ_i} extract attribute representations $\mathbf{a}_r, \mathbf{a}_p, \mathbf{a}_s$. Prosody predictor H_ϕ estimates prosodic features based on \mathbf{a}_r and \mathbf{a}_p . Synthesizer G_ϕ reconstructs speech from \mathbf{a}_s and the estimated prosody.

the set of all possible speech units. We use (\mathbf{U}, \mathbf{L}) to denote a new unit sequence formed with duplicating each u_k by l_k across time. We use learnable embeddings to represent each speech unit in \mathcal{U} as a dense vector.

We further introduce three prosodic features $\mathbf{P}, \mathbf{V}, \mathbf{Q}$ that can be directly inferred from the waveform \mathbf{W} . The pitch variation sequence $\mathbf{P} = (p_j \in \mathbb{R})_{j=1}^J$ is mean-normalized pitch in Hertz. For the voicing sequence $\mathbf{V} = (v_j \in \{0, 1\})_{j=1}^J$, 0 represents unvoiced and 1 represents voiced. J is the total number of frames. We can obtain \mathbf{P} and \mathbf{V} from pitch estimator such as CREPE [15]. We obtain energy $\mathbf{Q} = (q_n \in \mathbb{R})_{n=1}^N$ from the frame-wise \mathcal{L}_2 norm of the linear spectrogram (the same resolution as \mathbf{X}). Given these, our framework can be initially represented by:

$$\mathbf{a}_i = E_{\theta_i}(\mathbf{W}) \quad (1)$$

$$\{\tilde{\mathbf{P}}, \tilde{\mathbf{V}}, \tilde{\mathbf{Q}}, \tilde{\mathbf{L}}\} = H_\phi(\mathbf{U}, \mathbf{a}_p, \mathbf{a}_r) \quad (2)$$

$$\tilde{\mathbf{X}} = G_\phi(\tilde{\mathbf{P}}, \tilde{\mathbf{V}}, \tilde{\mathbf{Q}}, \mathbf{U}, \tilde{\mathbf{L}}, \mathbf{a}_s) \quad (3)$$

where E_{θ_i} is the attribute encoder for each speech attribute, as shown in the upper-left part of Figure 1. H_ϕ is our learnable prosody predictor used to estimate prosodic features given units \mathbf{U} and prosodic attributes $\mathbf{a}_r, \mathbf{a}_p$ (corresponding to the middle part of Figure 1). We use $\tilde{\mathbf{P}}, \tilde{\mathbf{V}}, \tilde{\mathbf{Q}}, \tilde{\mathbf{L}}$ to denote the network’s estimation of $\mathbf{P}, \mathbf{V}, \mathbf{Q}, \mathbf{L}$. G_ϕ is the synthesizer that reconstructs the speech $\tilde{\mathbf{X}}$ given the units \mathbf{U} , the predicted prosodic features, and the speaker attribute representation \mathbf{a}_s . Our objective is designing G_ϕ and H_ϕ to make \mathbf{a}_i (equivalently E_{θ_i}) capture the desired speech attributes. As a consequence, by changing \mathbf{a}_i we can manipulate different aspect of $\tilde{\mathbf{X}}$.

2.2. Synthesizer G_ϕ and Speaker Attribute \mathbf{a}_s

Inspired by [12], we follow the source-filter speech production model [16] to decompose G_ϕ in Eq.3 into the addition of two networks. We further model the energy separately³:

$$\begin{aligned} G_\phi(\mathbf{P}, \mathbf{V}, \mathbf{Q}, \mathbf{U}, \mathbf{L}, \mathbf{a}_s) \\ = S_{\phi_s}(\mathbf{P}, \mathbf{V}, \mathbf{a}_s) + F_{\phi_f}(\mathbf{U}, \mathbf{L}, \mathbf{a}_s) \oplus M_{\phi_m}(\mathbf{Q}) \end{aligned} \quad (4)$$

where S_{ϕ_s}, F_{ϕ_f} are source and filter networks that map the input to the same shape as the mel-spectrogram \mathbf{X} (sequence of length N with dimension d_x). M_{ϕ_m} is the energy network that outputs a scalar sequence of length N . We use \oplus to denote the broadcast addition across d_x . We now introduce each module in detail.

Filter Network. Speaker timbre is characterized by the speaker’s static articulator shapes (e.g., vocal tract length). These articulator

shapes further influence the phonation of linguistic units \mathbf{U} . To model this process, we fuse each embedded linguistic unit in (\mathbf{U}, \mathbf{L}) with \mathbf{a}_s before passing it to F_{ϕ_f} , as illustrated in the middle part of Figure 1. The fusion is done simply with channel-wise concatenation followed by a linear layer. For the network architecture, we adopted repeated residual blocks which consists of 1d-CNN, ReLU activation, linear layer, residual connection and layer normalization [17]. The same structure is also used for all networks $F_{\phi_f}, S_{\phi_s}, M_{\phi_m}, N_{\phi_p}, N_{\phi_r}$. Since the length of (\mathbf{U}, \mathbf{L}) is different from that of \mathbf{X} (length N), we use nearest interpolation on the output of an intermediate block to align two time sequences.

Source Network. The source network S_{ϕ_s} processes the pitch variation sequence \mathbf{P} and voicing sequence \mathbf{V} into an excitation spectrogram. We first follow a similar procedure in [15] to process pitch. Specifically, we first define B equal frequency bins $\mathbf{C}^p = \{c_i = i\ell_b + p_{\min}\}_{i=1}^B$. We choose the minimum normalized pitch $p_{\min} = -250$, and the bin width $\ell_b = 2.5$. We then calculate $\mathbf{b}(p_j) = \{b_i(p_j)\}_{i=1}^B$, the bin weight of c_i for p_j by applying Gaussian blur:

$$b_i(p_j) = \exp\left(-\frac{(p_j - c_i)^2}{2\sigma^2}\right) \quad (5)$$

We choose the blur standard deviation $\sigma = 4$, and $B = 200$. Finally, we compute the dense representation of p_j : $\mathbf{o}(\mathbf{b}(p_j), \mathbf{E}^p)$ by the weighted sum of a set of randomly initialized learnable embeddings $\mathbf{E}^p = \{\mathbf{e}_i^p \in \mathbb{R}^{d_e}\}_{i=1}^B$ with the bin weights $\mathbf{b}(p_j)$:

$$\mathbf{o}(\mathbf{b}(p_j), \mathbf{E}^p) = \frac{\sum_{i=1}^B b_i(p_j) \mathbf{e}_i^p}{\sum_{i=1}^B b_i(p_j)} \quad (6)$$

d_e is the dimension of the embedding. To include voicing information \mathbf{V} , we replace the unvoiced frames (j where $v_j = 0$) of $\mathbf{o}(\mathbf{b}(p_j), \mathbf{E}^p)$ with another learnable embedding before further processing. Similar to the filter network, we provide \mathbf{a}_s to each time frame of $\mathbf{o}(p_j)$. However, \mathbf{a}_s is now responsible for recovering the average f_0 discarded in mean normalization of \mathbf{P} . Finally, we add the output of the filter and source network (both the same shape as \mathbf{X}) and term the output as the prior mel-spectrogram. It carries most of the information $(\mathbf{P}, \mathbf{U}, \mathbf{L}, \mathbf{a}_s)$ needed to reconstruct the original speech, except for energy.

Energy Network. Another attribute unrelated to speaker timbre is energy. While directly adding energy \mathbf{Q} to the prior mel-spectrogram $(S_{\phi_s} + F_{\phi_f})$ is feasible, this restricts the prior mel-spectrogram to have equal spectral energy across time. Instead, we train another network M_{ϕ_m} to process energy, as shown in Eq.4 and the right part of Figure 1. We adopt a similar procedure of encoding

³During inference, we pass $\tilde{\mathbf{P}}, \tilde{\mathbf{V}}, \tilde{\mathbf{Q}}, \tilde{\mathbf{L}}$ instead for Eq.4.

pitch \mathbf{P} mentioned in the previous paragraph to encode energy \mathbf{Q} .⁴ We denote the corresponding outcome of Eq.5 and Eq.6 as $b_i(q_n)$ and $\mathbf{o}(\mathbf{b}(q_n), \mathbf{E}^q)$, where $\mathbf{E}^q = \{\mathbf{e}_i^q \in \mathbb{R}^{d_e}\}_{i=1}^B$ is the learnable embedding for energy bins. M_{ϕ_m} then maps the encoded energy sequence $\{\mathbf{o}(\mathbf{b}(q_n), \mathbf{E}^q)\}_{n=1}^N$ to a scalar sequence of length N .

2.3. Prosody Predictor H_ϕ and Prosodic Attributes $\mathbf{a}_p, \mathbf{a}_r$

We now introduce the prosody predictor H_ϕ in Eq.2. We decompose H_ϕ into the cascade of duration network N_{ϕ_r} and pitch-energy network N_{ϕ_p} . We first model rhythm by duration prediction:

$$\tilde{\mathbf{L}} = N_{\phi_r}(\mathbf{U}, \mathbf{a}_r) \quad (7)$$

The rhythm representation \mathbf{a}_r is trained to encode information useful for recovering the duration \mathbf{L} from unit \mathbf{U} . For the prediction of pitch, we condition on (\mathbf{U}, \mathbf{L}) and \mathbf{a}_p :

$$(\tilde{\mathbf{P}}, \tilde{\mathbf{V}}, \tilde{\mathbf{Q}}) = N_{\phi_p}(\mathbf{U}, \mathbf{L}, \mathbf{a}_p) \quad (8)$$

We predict $\tilde{\mathbf{P}}, \tilde{\mathbf{V}}, \tilde{\mathbf{Q}}$ jointly with \mathbf{a}_p as they are highly correlated. We condition on (\mathbf{U}, \mathbf{L}) as prosody is correlated with language content [14]. For $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{Q}}$, instead of predicting the scalar values, we follow [15] to predict bin weights. We use $\tilde{\mathbf{b}}_j^p, \tilde{\mathbf{b}}_n^q$ to denote the network estimation of bin weights $\mathbf{b}(p_j), \mathbf{b}(q_n)$ calculated by Eq.5. Note that for the source and energy network, it is sufficient to provide bin weights to calculate encoded pitch and energy with Eq.6.

2.4. Attribute Encoders E_{θ_i}

Studies [18, 12] have found Wav2Vec 2.0 [10] useful for learning utterance-level representation. Here we adopt pretrained⁵ Wav2Vec 2.0 as our attribute encoders E_{θ_i} in Eq.1. Wav2Vec 2.0 consists of a cascade of CNN feature extractor and transformer encoder blocks. We only use its CNN feature extractor and the first layer of its transformer encoder. Specifically, we fixed the CNN feature extractor untrained, and fine-tune the 1-layer transformer separately for each E_{θ_i} . We then apply global average pooling on the output to collapse the time dimension, followed by a linear layer to obtain \mathbf{a}_i .

2.5. Optimization

We adopt \mathcal{L}_1 loss for the reconstruction of $\tilde{\mathbf{X}}$ and \mathbf{X} . We further apply commonly used adversarial loss to prevent over-smoothness of $\tilde{\mathbf{X}}$. We use least-squared adversarial loss [19] with the discriminator architecture following [12]. To predict prosodic features with H_ϕ , we minimize the binary cross entropy (BCE) between \mathbf{V} and $\tilde{\mathbf{V}}$. We follow [20] to use mean squared error (MSE) as loss function on log-scale duration for $\tilde{\mathbf{L}}$. For $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{Q}}$, we follow [15] to minimize BCE between predicted and ground truth bin weights (e.g. $\tilde{\mathbf{b}}_j^p$ and $\mathbf{b}(p_j)$). During training, we use the ground truth \mathbf{L}, \mathbf{V} as the input to all networks. During inference, we pass the predicted $\tilde{\mathbf{L}}, \tilde{\mathbf{V}}$.

Joint optimization. During training, a reasonable choice is to use the ground truth \mathbf{P}, \mathbf{Q} as the input to the synthesizer G_ϕ in Eq.3. However, this leaves no training signal between the synthesizer G_ϕ and the prosody predictor H_ϕ in Eq.2. One would imagine that the reconstruction and adversarial loss of $\tilde{\mathbf{X}}$ can guide the pitch-energy network N_{ϕ_p} to generate a perceptually natural pitch and energy contour. One straightforward solution is to pass the estimated pitch

($\tilde{\mathbf{b}}_j^p$) and energy ($\tilde{\mathbf{b}}_n^q$) to G_ϕ . Empirically, we found that this leads to G_ϕ ignoring the given prosodic information. We attribute this to the poor estimation in the early stage of training, which discourages G_ϕ to consider prosodic information. We instead pass the average of predicted and ground truth bin weights (e.g., $0.5(\mathbf{b}(p_j) + \tilde{\mathbf{b}}_j^p)$) to G_ϕ during training. Additionally, we minimize the MSE loss between the predicted and ground truth encoded pitch ($\mathbf{o}(\mathbf{b}(p_j), \mathbf{E}^p)$ and $\mathbf{o}(\tilde{\mathbf{b}}_j^p, \mathbf{E}^p)$). This explicitly encourages \mathbf{E}^p to have consistent encoding between $\mathbf{b}(p_j)$ and $\tilde{\mathbf{b}}_j^p$. The same loss is applied to energy.

3. EXPERIMENTAL SETUP

We use 16 blocks for F_{ϕ_f}, S_{ϕ_s} , 4 blocks for M_{ϕ_m} , 2 blocks for N_{ϕ_r} and 6 blocks for N_{ϕ_p} . The block is the residual block we introduced in Section 2.2. We apply k-means clustering on the final layer of HuBERT [21] with 200 clusters ($|\mathcal{U}| = 200$) as our self-supervised speech units. We use the textless-lib [22] to extract pitch \mathbf{P} , voicing \mathbf{V} and the HuBERT units (\mathbf{U}, \mathbf{L}) . We follow [23] to deduplicate consecutive HuBERT units as \mathbf{U} , and the number of duplicated frames form \mathbf{L} . We adopt pretrained HiFi-GAN [24] for mel-spectrogram inversions. We include detailed implementation in the released code. Given a source speech and a target speech, we evaluate our system with two settings: speaker conversion (transfer \mathbf{a}_s from target) and prosody conversion (transfer $\mathbf{a}_r, \mathbf{a}_p$ from target).

Datasets. We follow previous approaches [1, 5, 9] to train on the VCTK corpus [25]. VCTK consists of reading English with 400 sentences each from 110 speakers. We randomly sample 5% utterances for validation and the rest for training. To fully evaluate the one-shot capability, we test on utterances from LibriTTS [26]. LibriTTS is another reading English corpus with larger speaker and language content coverage (over 2000 speakers), providing a more unbiased evaluation. We randomly sample 1000 utterances each as the source and target speech respectively for both speaker and prosody transfer.

Comparing Methods. AutoVC [1], SRDVC [9] are chosen as competing methods for speaker conversion. We used the officially released checkpoints and HiFi-GAN vocoder for these methods. For prosody conversion, we compare our method with SRDVC. We further evaluate two variants of our system. First, we replace the pretrained Wav2Vec 2.0 with random initialization (-W2V2 in the tables). Second, we evaluate the system without the joint optimization losses we introduced in Section 2.5 (-Joint. Opt. in the tables).

Metrics. We report the character error rate (CER) of the syntheses to measure intelligibility [27]. We use google ASR API for the transcription. For objective metrics of speaker conversion, we report the cosine similarity between speaker embeddings of target and converted speech. We term this speaker embedding similarity (SES); its range is between $[-1, 1]$. Speaker embeddings are extracted with a pretrained speaker identification network⁶. Similarly, we extract emotion embeddings from a pretrained dimensional emotion classifier [28]. We again report the cosine similarity of emotion embeddings (EES). For subjective measures, human evaluations are conducted via Amazon Mechanical Turk. We randomly sampled 25 utterances from each method and assigned them to 10 workers. We report both the 5-scale mean opinion score (MOS) and 95% confidence interval (CI) on naturalness, speaker and prosody similarity.

⁴We choose the minimum energy to be 0 and the energy bin width to be 1 for the calculation of energy bins \mathbf{C}^q .

⁵<https://huggingface.co/facebook/wav2vec2-base>

Table 1: Evaluation results for **unseen speaker transfer**. The right columns are naturalness and speaker similarity MOS with 95% CI. *GT* stands for the ground truth speech.

Method	CER ↓	SES ↑	Naturalness ↑	Similarity ↑
<i>GT</i>	5.5%	<i>n/a</i>	3.95 ± 0.11	<i>n/a</i>
<i>AutoVC</i>	88.4%	0.14	2.58 ± 0.17	2.62 ± 0.15
<i>SRDVC</i>	34.7%	0.17	3.25 ± 0.15	2.56 ± 0.14
<i>Ours</i>	7.5%	0.34	3.50 ± 0.14	2.80 ± 0.14
(-W2V2)	7.8%	0.27	3.47 ± 0.13	2.62 ± 0.15
(-Joint Opt.)	7.3%	0.32	3.55 ± 0.14	2.68 ± 0.14

Table 2: Average Pearson correlation coefficients (PCC) of pitch and energy between target and speaker-converted speech.

PCC	<i>AutoVC</i>	<i>SRDVC</i>	<i>Ours</i> ($\tilde{\mathbf{P}}, \tilde{\mathbf{Q}}$)	<i>Ours</i> (\mathbf{P}, \mathbf{Q})
$\log f_0$	0.09	0.49	0.30	0.51
Energy	0.05	0.82	0.78	0.91

4. RESULTS

4.1. Speaker Conversion

Table 1 presents our experiment result on unseen speaker transfer. First, compared to AutoVC and SRDVC, our system achieves much higher intelligibility (lower CER) and naturalness. Apparently, the discrete self-supervised units provide better language information. For speaker transferability, our system achieves the highest similarity MOS and SES. The similarity MOS noticeably drops if we do not use pretrained Wav2Vec 2.0, suggesting that pretrained SSL models can capture more generalizable speaker characteristics. Without the joint optimization losses, the system obtains distinctively worse speaker similarity. For the synthesizer G_ϕ , joint optimization can be interpreted as a means of data augmentation for the given prosody. We conjecture that this encourages G_ϕ to learn more generalized \mathbf{a}_s for a wider variety of prosody patterns.

Disentanglement of speech attributes. Table 2 presents the average Pearson correlation coefficients (PCC) of prosodic features between target and speaker-converted speech. Higher PCC suggests that the corresponding attribute is less affected following speaker conversion. We analyze our system under two different conditions: passing reconstructed $\tilde{\mathbf{P}}, \tilde{\mathbf{Q}}$, and passing the ground truth \mathbf{P}, \mathbf{Q} . With the ground truth \mathbf{P}, \mathbf{Q} , our system achieves comparable PCC on pitch and distinctively higher PCC on energy compared to SRDVC.⁷ This shows the effectiveness of our energy modeling approach to disentangle \mathbf{a}_s and the energy contour (speaker conversion will not affect energy). On the other hand, using $\tilde{\mathbf{P}}, \tilde{\mathbf{Q}}$ leads to a lower PCC, suggesting that the pitch-energy reconstruction from $\mathbf{a}_r, \mathbf{a}_p$ is not perfect. Note that this is both reasonable and desirable; our goal for \mathbf{a}_p is not to memorize the exact pitch (energy) contour but to model the high-level speaking style of the utterance. The mapping from speaking style to the pitch contour is one-to-many, which naturally results in lower PCC.

Table 3: Evaluation results for **prosody transfer**. The right columns are naturalness and prosody similarity MOS with 95% CI.

Method	CER ↓	EES ↑	Naturalness ↑	Similarity ↑
<i>GT</i>	5.5%	<i>n/a</i>	3.95 ± 0.11	<i>n/a</i>
<i>SRDVC</i>	49.9%	0.28	3.06 ± 0.16	2.58 ± 0.15
<i>Ours</i>	8.9%	0.42	3.49 ± 0.13	2.69 ± 0.14
(-W2V2)	10.1%	0.35	3.39 ± 0.14	2.57 ± 0.15
(-Joint Opt.)	9.0%	0.38	3.36 ± 0.14	2.50 ± 0.14

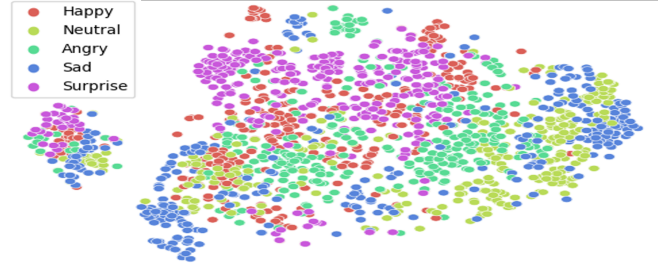


Fig. 2: Visualization of \mathbf{a}_p (pitch-energy) by running t-SNE on ESD.

4.2. Prosody Conversion

Table 3 shows the result for prosody conversion. We first observe that SRDVC suffers from much higher CER and lower naturalness MOS compared to its own performance in speaker conversion, suggesting that it failed to generate natural prosody. We observe that in many samples SRDVC directly transfers the pitch contour from the target speech without considering phonetic content. Its high similarity MOS (even higher than our ablated versions) but significantly lower naturalness further supports this claim. On the other hand, compared to speaker conversion, our system shows a small increase in CER and almost no difference in naturalness MOS. Since our prosody prediction is conditioned on discrete speech units, consistency between language content and prosody can be learned naturally. Additionally, our system better transfers prosody, judging from the highest EES score and similarity MOS. Table 3 also shows that joint optimization noticeably increases prosody naturalness and similarity. It suggests that the adversarial loss and reconstruction loss indeed provide useful guidance to the prosody predictor H_ϕ . All measures noticeably degrade without pretrained SSL models, indicating its usefulness for extracting speaking styles.

4.3. Visualization of Prosody Representations.

We further visualize \mathbf{a}_p by running t-SNE [29] on the Emotional Speech Database (ESD) [30]. ESD contains 350 English utterances spoken by 10 speakers with 5 emotion categories. From Figure 2, we observe that despite being trained on only reading speech (VCTK), \mathbf{a}_p still forms clusters of emotions. In particular, sad and neutral mostly covers the lower part while happy and surprise concentrate on the upper part of Figure 2. This validates that \mathbf{a}_p indeed captures high-level speaking style information.

⁶<https://github.com/pyannote/pyannote-audio>

⁷Note that SRDVC also accepts ground truth pitch contour as input.

5. CONCLUSION

We describe a unified system for one-shot prosody and speaker conversion trained in an unsupervised manner. We evaluate the intelligibility, naturalness, speaker and prosody transferability of synthetic speech and show the superior performance of our approach. Our work potentially benefits various downstream tasks including voice conversion, emotion analysis, speech data augmentation, and expressive speech synthesis. Based on this work, we intend to extend our system to real-world speech, where background noise and recording environments are additional attributes to consider. We also plan to investigate the potential downstream applications of learned attribute representations.

6. ACKNOWLEDGEMENTS

We are grateful to Amazon for the support of this research.

7. REFERENCES

- [1] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "AutoVC: Zero-shot voice style transfer with only autoencoder loss," in *Proc. ICML*, 09–15 Jun 2019, vol. 97, pp. 5210–5219, PMLR.
- [2] K. Qian, Z. Jin, M. Hasegawa-Johnson, and G. J. Mysore, "F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder," in *Proc. ICASSP*, 2020, pp. 6284–6288.
- [3] A. T. Liu, P. chun Hsu, and H.-Y. Lee, "Unsupervised End-to-End Learning of Discrete Linguistic Units for Voice Conversion," in *Proc. Interspeech*, 2019, pp. 1108–1112.
- [4] D.-Y. Wu, Y.-H. Chen, and H. yi Lee, "VQVC+: One-Shot Voice Conversion by Vector Quantization and U-Net Architecture," in *Proc. Interspeech*, 2020, pp. 4691–4695.
- [5] D. Wang, L. Deng, Y. T. Yeung, X. Chen, X. Liu, and H. Meng, "VQMVC: Vector Quantization and Mutual Information-Based Unsupervised Speech Representation Disentanglement for One-Shot Voice Conversion," in *Proc. Interspeech*, 2021, pp. 1344–1348.
- [6] K. Qian, Y. Zhang, S. Chang, J. Xiong, C. Gan, D. Cox, and M. Hasegawa-Johnson, "Global prosody style transfer without text transcriptions," in *Proc. ICML*, 18–24 Jul 2021, vol. 139, pp. 8650–8660, PMLR.
- [7] K. Qian, Y. Zhang, S. Chang, M. Hasegawa-Johnson, and D. Cox, "Unsupervised speech decomposition via triple information bottleneck," in *Proc. ICML*, 13–18 Jul 2020, vol. 119, pp. 7836–7846, PMLR.
- [8] C. Ho Chan, K. Qian, Y. Zhang, and M. Hasegawa-Johnson, "SpeechSplit2.0: Unsupervised speech disentanglement for voice conversion without tuning autoencoder bottlenecks," in *Proc. ICASSP*, 2022, pp. 6332–6336.
- [9] S. Yang, M. Tantrawenith, H. Zhuang, Z. Wu, A. Sun, J. Wang, N. Cheng, H. Tang, X. Zhao, J. Wang, and H. Meng, "Speech Representation Disentanglement with Adversarial Mutual Information Learning for One-shot Voice Conversion," in *Proc. Interspeech*, 2022, pp. 2553–2557.
- [10] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020, vol. 33, pp. 12449–12460, Curran Associates, Inc.
- [11] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech*, 2019, pp. 3465–3469.
- [12] H.-S. Choi, J. Lee, W. Kim, J. Lee, H. Heo, and K. Lee, "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," in *Proc. NeurIPS*, 2021, vol. 34, pp. 16251–16265, Curran Associates, Inc.
- [13] W.-C. Huang, S.-W. Yang, T. Hayashi, H.-Y. Lee, S. Watanabe, and T. Toda, "S3PRL-VC: Open-source voice conversion framework with self-supervised speech representations," in *Proc. ICASSP*, 2022, pp. 6552–6556.
- [14] A. Cutler, D. Dahan, and W. van Donselaar, "Prosody in the comprehension of spoken language: A literature review," *Language and Speech*, vol. 40, no. 2, pp. 141–201, 1997, PMID: 9509577.
- [15] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *Proc. ICASSP*, 2018, pp. 161–165.
- [16] G. Fant, "Acoustic theory of speech production," in *The Hague, The Netherlands, Mouton*, 1960.
- [17] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.
- [18] L.-W. Chen and A. Rudnicky, "Fine-grained style control in transformer-based text-to-speech synthesis," in *Proc. ICASSP*, 2022, pp. 7907–7911.
- [19] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proc. ICCV*, Oct 2017.
- [20] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, robust and controllable text to speech," in *Proc. NeurIPS*, 2019, vol. 32, Curran Associates, Inc.
- [21] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhota, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [22] E. Kharitonov, J. Copet, K. Lakhota, T. A. Nguyen, P. Tomasello, A. Lee, A. Elkahky, W.-N. Hsu, A. Mohamed, E. Dupoux, and Y. Adi, "textless-lib: a library for textless spoken language processing," in *Proc. NAACL*, July 2022, pp. 1–9, Association for Computational Linguistics.
- [23] E. Kharitonov, A. Lee, A. Polyak, Y. Adi, J. Copet, K. Lakhota, T. A. Nguyen, M. Riviere, A. Mohamed, E. Dupoux, and W.-N. Hsu, "Text-free prosody-aware generative spoken language modeling," in *Proc. ACL*, Dublin, Ireland, May 2022, pp. 8666–8681, Association for Computational Linguistics.
- [24] J. Kong, J. Kim, and J. Bae, "HiFi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeurIPS*, 2020, vol. 33, pp. 17022–17033, Curran Associates, Inc.
- [25] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)," 2019.
- [26] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech," in *Proc. Interspeech*, 2019, pp. 1526–1530.

- [27] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, “Espnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit,” in *Proc. ICASSP*, 2020, pp. 7654–7658.
- [28] J. Wagner, A. Triantafyllopoulos, H. Wierstorf, M. Schmitt, F. Burkhardt, F. Eyben, and B. W. Schuller, “Dawn of the transformer era in speech emotion recognition: closing the valence gap,” *CoRR*, vol. abs/2203.07378, 2022.
- [29] L. van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.
- [30] K. Zhou, B. Sisman, R. Liu, and H. Li, “Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset,” in *Proc. ICASSP*, 2021, pp. 920–924.