DisC-VC: Disentangled and $F_0$-Controllable Neural Voice Conversion

Chihiro Watanabe* and Hirokazu Kameoka†

* NTT Communication Science Laboratories, Japan
E-mail: ch.watanabe@ntt.com Tel/Fax: +81-46240458
† NTT Communication Science Laboratories, Japan
E-mail: hirokazu.kameoka@ntt.com Tel/Fax: +81-462403645

Abstract—Voice conversion is a task to convert a non-linguistic feature of a given utterance. Since nuance of speech strongly depends on its pitch pattern, in some applications, it would be desirable to keep the original rise/fall pitch pattern while changing the speaker identity. Some of the existing methods address this problem by either using a source-filter model or developing a neural network that takes an $F_0$ pattern as input to the model. Although the latter approach can achieve relatively high sound quality compared to the former one, there is no consideration for discrepancy between the target and generated $F_0$ patterns in its training process. In this paper, we propose a new variational-autoencoder-based voice conversion model accompanied by an auxiliary network, which ensures that the conversion result correctly reflects the specified $F_0$/timbre information. We show the effectiveness of the proposed method by objective and subjective evaluations.

I. INTRODUCTION

Voice conversion (VC) is a task to convert a non-linguistic feature of a given utterance and it can be used for various applications [1] including personalizing or changing emotion of text-to-speech systems [2]–[4]. In this paper, we focus on the conversion of $F_0$ pattern and timbre (i.e., speaker identity).

A basic VC framework consists of feature extraction from a speech signal, feature conversion, and speech synthesis based on a converted feature. With regard to feature conversion, some of the existing methods require parallel data set (e.g., Gaussian-mixture-model-based [5]–[7], deep-neural-network-based [8], [9], and sequence-to-sequence-based [10], [11] methods), while others do not (e.g., recognition-synthesis-based [12]–[14], phonetic-posteriorgram-based [15], variational-autoencoder (VAE) [16]–[19], generative-adversarial-network-based [20]–[22], flow-based [23], [24], and score-based [25], [26] methods). As for feature extraction and speech synthesis, the existing methods can be classified roughly into two approaches. One is the parametric-vocoder-based (PV) approach [6], [8], [17], where we first apply timbre conversion based on mel-spectral coefficients (MCCs) and $F_0$ conversion (e.g., linear transformation) separately and then synthesize a waveform with a parametric vocoder such as WORLD [27]. The other is the neural-vocoder-based (NV) approach [18], [28], [29], where we use a neural network for speech synthesis. While early neural vocoders have required spectral envelope- and excitation-related features as input [30], [31], the recent mainstream is to synthesize a waveform from a mel-spectrogram [32], [33].

In some applications such as online communication, it would be desirable to keep the original rise/fall pitch pattern, since it strongly affects the nuance of speech. Inappropriate modification of the relative pitch pattern would cause miscommunication in regard to the speaker’s intent and/or emotion. One way to securely keep the relative pitch pattern of input speech would be to take the PV approach, which allows us to explicitly specify the $F_0$ pattern of the converted speech. However, one disadvantage is that it cannot generally achieve as high sound quality as the NV approach. Recently, some $F_0$-controllable NV methods have been proposed, with which we can specify both $F_0$ and timbre conditionings explicitly [34]–[36]. However, even with these models, a generated speech does not necessarily have the same $F_0$ pattern as the input conditioning. This is because the relationship between the specified $F_0$ pattern and the conversion result has not been considered in the training criteria, and thus the decoder may disregard the specified $F_0$ pattern and not reproduce it faithfully in the reconstruction process.

In this paper, we develop a disentangled and $F_0$-controllable neural VC (DisC-VC). DisC consists of the VAE-based generator and auxiliary network. The auxiliary network, which is composed of $F_0$ extractor and speaker classifier, ensures that the conversion result correctly reflects the specified $F_0$/timbre information. We show the effectiveness of DisC by both objective and subjective evaluations.

II. PROPOSED METHOD

A. Preprocessing of training samples

Fig. 1 shows the architecture of the DisC model. During the training phase, we use the standardized log-mel-spectrogram $x \in \mathbb{R}^{T \times T}$, the log $F_0$ pattern $\lambda \in \mathbb{R}^{T}$, and the speaker index $s \in \{1, 2, \ldots, S\}$ as the input features, where $T$, $F_0$, and $S$ indicate the numbers of time and frequency bins and speakers, respectively. To define $x$, we first compute the log-mel-spectrogram $x^{(0)}$. Then, each entry of the column vectors of $x^{(0)}$ is standardized according to the mean and standard deviation for all the time bins and training samples. To compute $\lambda$, we extract the $F_0$ pattern from an input
speech signal with the WORLD $F_0$ estimator [27] and apply logarithmic transformation to its non-zero entries.

### B. Generator

The generator consists of content encoder $ENC_C$ and decoder $DEC$. To reflect the discrete nature of the phoneme units of speech, we adopt the encoder of Vector Quantised-VAE [40] as $ENC_C$. In the generator, we first extract the content feature $c = ENC_C(x) \in \mathbb{R}^{NC \times T}$ from $x$, where we set $NC = 16$ in the experiment. Then, the decoder reconstructs the mean and variance of the log mel-spectrogram $(\hat{\mu}, \hat{\sigma}) = DEC(c, \lambda, s)$, where $\hat{\mu}, \hat{\sigma} \in \mathbb{R}^{F \times T}$.

### C. Training objective

1) Reconstruction error: The first training objective is the reconstruction error of the generator. Let $q_0(c|x)$ be a probability density function (PDF) of content feature $c$ conditioned on log mel-spectrogram $x$, based on which the content encoder with parameter $\phi$ generates the content feature. Let $p_0(x|c, \lambda, s)$ be a PDF of $x$ conditioned on $(c, \lambda, s)$, based on which the decoder with parameter $\theta$ generates the output mel-spectrogram. We assume the prior distribution of $c$ to be a discrete uniform distribution independent of the $F_0$/timbre features (i.e., $p(c|\lambda, s) = p(c)$). By using Jensen’s inequality, the negative log marginal distribution can be upper bounded by

$$-\log p_0(x|\lambda, s) = -\log \int q_0(c|x)p_0(x|c, \lambda, s)p(c)\,dc$$

$$\leq -\int q_0(c|x)\log \left(\frac{p_0(x|c, \lambda, s)p(c)}{q_0(c|x)}\right)\,dc$$

$$= D - \mathbb{E}_{c \sim q_0(c|x)}[\log p_0(x|c, \lambda, s)],$$

(1)

where $D := D_{KL}[q_0(c|x)||p(c)]$ and $D_{KL}(q||p)$ indicates the KL divergence of $q$ from $p$. It can be shown that the difference between the right- and left-hand sides of (1) is equal to $D_{KL}(q_0(c|x)||p_0(x|c, \lambda, s))$, which is minimized when $q_0(c|x) = p_0(c|x, \lambda, s)$ (i.e., the content feature $c$ extracted by the encoder is independent of $(\lambda, s)$, given $x$). When $p(c)$ is a discrete uniform distribution, $D$ becomes constant [40].

By assuming $p_0(x|c, \lambda, s)$ to be a Gaussian distribution with mean $\mu$ and variance $\sigma$, where all the entries are mutually independent, the second term in (1) can be interpreted as the reconstruction error of $x$ [16]. Given training samples of $(x, \lambda, s)$, the actual training criterion $L_{\text{like}}$ will be the expectation of the right-hand side of (1) over $(x, \lambda, s)$.

2) Negative mutual information: As we later show experimentally in Section III, even if the conversion model is conditioned on specified $F_0$/timbre information $(\lambda_R, s_R)$, the converted speech does not always reflect them sufficiently with a model trained only based on the reconstruction loss $L_{\text{like}}$, since the decoder may disregard $(\lambda_R, s_R)$ and only use $c$ to reconstruct the log mel-spectrogram. Therefore, as the second training objective, given $c \sim q_0(c|x)$ and $s_r$, we consider the negative mutual information $I_P(\theta)$ between output log mel-spectrogram $x \sim p(x|c, \lambda_R, s_R)$ and log $F_0$ pattern $\lambda_R$, which is given by

$$I_P(\theta) := -\mathbb{E}_{x \sim p_0(x|c, \lambda_R, s_R)}[\lambda_R \sim p(\lambda_R|x) \left\{ \log\left(\frac{p(\lambda_R, x)}{p(\lambda_R)p(x)}\right) \right\}$$

$$= -\mathbb{E}_{x \sim p_0(x|c, \lambda_R, s_R)}[\lambda_R \sim p(\lambda_R|x)\log p(\lambda_R|x)] + H_P,$$

(2)

where $H_P := \mathbb{E}_{\lambda_R \sim p(\lambda_R)}[\log p(\lambda_R)]$ is a constant that does not depend on $\theta$. By using the fact that a KL divergence is non-negative and the law of total expectation (i.e., Lemma 5.1 in [41]), the expectation of the first term in (2) over $\lambda_R$ can be upper bounded by

$$-\mathbb{E}_{\lambda_R \sim p(\lambda_R), x \sim p_0(x|c, \lambda_R, s_R)}[\lambda_R \sim p(\lambda_R|x)\log p(\lambda_R|x)]$$

$$\leq -\mathbb{E}_{\lambda_R \sim p(\lambda_R), x \sim p_0(x|c, \lambda_R, s_R), \lambda_R \sim p(\lambda_R), s \sim p(s)}[\log q(\lambda_R|x)]$$

$$= -\mathbb{E}_{\lambda_R \sim p(\lambda_R), x \sim p_0(x|c, \lambda_R, s_R), s \sim p(s)}[\log q(\lambda_R|x)] =: R_P,$$

(3)

where $q(\lambda_R|x)$ is an arbitrary conditional PDF. The equality holds if $q(\lambda_R|x) = p(\lambda_R|x)$. Therefore, assuming $q(\lambda_R|x)$ in some parametric form, the left-hand side of (3) can be decreased by alternately decreasing $R_P$ with respect to $(\phi, \theta)$ and $q$. Specifically, for all $i \in \{1, \ldots, T\}$, we represent $q(\lambda_R|x)$ as the Laplace distribution with location parameter $(\lambda_{R_i})$ and scale parameter of one, where $\lambda_{R_i} = P_{\text{ext}}(\mu_R)$ is an output of an auxiliary neural network (i.e., $F_0$ extractor) $P_{\text{ext}}: \mathbb{R}^{F \times T} \rightarrow \mathbb{R}^F$ and $(\mu_R, \sigma_R) = DEC(C, \lambda_R, s_R)$, and consider training its parameters $\psi_P$ along with $\phi$ and $\theta$. Hence, $P_{\text{ext}}$ is trained to estimate the log $F_0$ pattern from an input log mel-spectrogram.

Given $c$ and $\lambda_R$, we also consider using the negative mutual information $I_T(\theta)$ between $x$ and $s_R$ as a training criterion. As above, the expectation of $I_T(\theta)$ except for a constant term over $s_R$ can be upper bounded by $R_T := -\mathbb{E}_{s_R \sim p(s_R), x \sim p_0(x|c, \lambda_R, s_R)}[\log r(s_R|x)]$, where $r(s_R|x)$ is an arbitrary categorical distribution and the exact bound is achieved when $r(s_R|x) = p(s_R|x)$. Thus, it can be decreased by alternately decreasing $R_T$ with respect to $r$ and $(\phi, \theta)$. Here, we assume $r(s_R|x)$ to be a categorical distribution with each probability given as an output of an auxiliary neural network (i.e., speaker classifier) $CLS: \mathbb{R}^{F \times T} \rightarrow \mathbb{R}^{T_5}$,\footnote{$T_5$ is an arbitrary natural number. As shown in Fig. 1, CLS contains convolutional layers and thus $T_5$ depends on $T$ in the experiment.} and consider training its parameters $\psi_T$ along with $\phi$, $\theta$, and $\psi_P$. Hence, CLS is trained to detect the speaker from an input log mel-spectrogram.

The criteria $R_T$ and $R_T$ are defined under specific $(c, s_R)$ and $(c, \lambda_R)$, respectively. Since $c$, $s_R$, and $\lambda_R$ are determined stochastically according to the training examples, the actual criteria used for training will be the expectations of $R_P$ and $R_T$ with respect to $c$, $s_R$, and $\lambda_R$. Let $L_P := \mathbb{E}_{x \sim p_0(x|c \sim q_0(c|x), s \sim p(s))}[R_P]$ and $L_T := \mathbb{E}_{x \sim p_0(x|c \sim q_0(c|x), \lambda_R \sim p(\lambda_R))}[R_T]$, where $x' \sim p(x')$ indicates the log mel-spectrogram of a randomly drawn training sample. The process $\lambda_R \sim p(\lambda_R)$ can be implemented as random resampling (RR), which converts an input log $F_0$ pattern $\lambda$ into random one $\lambda_R$. In the experiment, we defined
the RR of \( F_0 \) by \( (\lambda_R)_i = \lambda_i + \beta \) with \( \beta \sim \text{Uniform}([0, 3, 3]) \), if \( \lambda_i \neq 0 \), and \( (\lambda_R)_i = \lambda_i \), otherwise. We also defined the RR of speaker index by generating \( s_R \) from a discrete uniform distribution on \([1, \ldots, S]\). The idea of RR has also been adopted in SpeechFlow [19], however, there has been no consideration for the discrepancy between the specified \( F_0 \)/timbre information and those of the conversion result.

The target log \( F_0 \) pattern \( \lambda \in \mathbb{R}^T \) is defined as follows. For each training sample \( x' \), the decoder generates \( x \) using \( c = \text{ENC}_C(x') \) along with \( \lambda_R \) and \( s_R \) randomly drawn from predefined distributions \( p(\lambda_R) \) and \( p(s_R) \), respectively. The auxiliary network then extracts \( \lambda_0 \) and \( s_0 \) from \( x \) and checks whether they are consistent with \( \lambda_R \) and \( s_R \). In addition, to better express unvoiced and voiced speech and checks whether they are consistent with \( \lambda_R \), we also defined \( \lambda_0 = \text{DEC}(C, \lambda_0, s) \) as a training criterion, where \( (\mu_0, \sigma_0) = \text{DEC}(C, \lambda_0, s) \). We also define \( \lambda_1 \) by \( \lambda = \text{Pext}(x) \), \( \lambda_2 \) for \( \lambda \) and \( \lambda_{\text{rec}} = \text{Pext}(\hat{\mu}) \), \( \hat{\lambda}_T \) for \( s \) and \( s_{\text{rec}} = \text{CLS}(x) \), and \( \hat{\lambda}_T \) for \( s \) and \( \hat{\lambda}_{\text{rec}} = \text{CLS}(\hat{\mu}) \).

We train the entire network to minimize \( L := \eta L_{\text{like}} + \eta_1(\lambda_P + L_{P_0} + \lambda_{P_1} + L_{P_2}) + \eta_2(\hat{\lambda}_T + \frac{1}{2}L_{T_1} + \frac{1}{2}L_{T_2}) \) based on back-propagation. In the experiment, we set \( \eta = 1/F'T \), \( \eta_1 = 1/4T \), and \( \eta_2 = 1/2T_3 \), and replaced expectation over \( x' \) in computing each loss with a sample mean of a given mini-batch, which consists of 16 randomly selected training samples. A training sample was defined as \( \{(x_s, \lambda_s, s)|s = 1, \ldots, S\} \), where \( x_s \) and \( \lambda_s \) were a log mel-spectrogram and a log \( F_0 \) pattern of the \( s \)th speaker, respectively. It must be noted that the criterion \( L \) can be used not only for training the generator considered in this study, but also for training other existing generator networks that allow \( F_0 \)/timbre conditioning (e.g., SpeechFlow [19]).

D. Conversion Process

By using the trained DisC model, we can apply the following types of conversion to a given test sample \( (x, \lambda, s) \): (1) change log \( F_0 \) pattern by a given value \( \beta \in \mathbb{R} \), (2) change log \( F_0 \) pattern from speaker \( s \) to \( \hat{s}_p \), and (3) change timbre from speaker \( s \) to \( \hat{s}_T \). By defining \( \beta = 0 \), \( \hat{s}_p = s \), or \( \hat{s}_T = s \), we can also perform each type of conversion individually.

The target log \( F_0 \) pattern \( \lambda \in \mathbb{R}^T \) is defined as follows. Let \( \mu_P(s) \) and \( \sigma_P(s) \), respectively, be the sample mean and sample standard deviation of the log \( F_0 \) pattern (except for the zero entries) of the \( s \)th speaker in the training data set. For all \( i \in \{1, \ldots, T\} \), we define \( \lambda_i = [\sigma_P(s_i)/\sigma_P(s)][\lambda_i - \mu_P(s)] + \mu_P(s) + \beta \), if \( \lambda_i \neq 0 \), and \( \lambda_i = 0 \), otherwise.

We obtain \( (\mu, \sigma) = \text{DEC}(\text{ENC}_C(x), \lambda, \hat{s}_T) \) and reconstruct a waveform from \( \mu \) with a neural vocoder. In the experiment, we used Parallel WaveGAN [33].

III. EXPERIMENT

A. Experimental settings

By using the CMU Arctic speech databases [42], we trained the DisC model with the 1,000 training samples per speaker for 5,000 epochs, applied \( F_0 \) and/or timbre conversion to the 132 test samples per speaker, and compared the results with those obtained with the following baseline methods. The number of speakers was four in both the training and test data sets.

DisC-NoAux is DisC with no auxiliary network, and it is trained by minimizing \( L_{\text{like}} \) in Section II-C1. ACVAE-C [17] is
the original ACVAE that takes MCCs as input and reconstructs a waveform from converted MCCs and \( F_0 \) pattern by using a WORLD vocoder [27]. ACVAE-S [29] is the extended ACVAE that takes mel-spectrogram as input and reconstructs a waveform from a converted mel-spectrogram by using a neural vocoder. For reference, we also used WORLD vocoder [27] itself for \( F_0 \)-only conversion by using the MCCs of an input signal and \( \lambda \) given in Section II-D. We also tried \( F_0 \) and timbre conversion with BNE-Seq2SeqMoL (BNE) [35] as an example of \( F_0 \)-controllable methods. It must be noted that we cannot compare its result with those of the other methods as it is, since we used the pretrained synthesis model provided by the authors of [35], which was trained with a different data set than the CMU Arctic one.

To compare these methods, we tried task P to change log \( F_0 \) pattern by \( \beta \in \{ \log(1.5), -\log(1.5) \} \), task T to change timbre to that of a target speaker, and task PT to change both \( F_0 \) pattern and timbre to those of a target speaker. Since only simultaneous conversion of \( F_0 \) pattern and timbre is possible with ACVAE-S, we use the same results in tasks T and PT for this method. Before evaluation, we trimmed the leading and trailing silent sections of all the target and converted samples. In tasks T and PT, we tried all 12 patterns of combinations of source and target speakers.

**B. Objective evaluation**

To evaluate \( F_0 \) and timbre conversion performance, we adopted the \( F_0 \) error \( \Delta F_0 \) (i.e., the root mean squared error between two log \( F_0 \) patterns) and the mel-cepstral distortion (MCD), respectively. Since target and converted speech signals are not necessarily aligned in the time direction, we first computed \( \Delta F_0 \) between the target log \( F_0 \) pattern and that of the converted speech. We evaluated the MCD between the samples before and after conversion for task P and those between the target and converted samples for tasks T and PT.

Tables I, II, and III show the results of tasks P, T, and PT, respectively. Table I shows that the timbre of the source speech was maintained more successfully with DisC than with DisC-NoAux in \( F_0 \)-only conversion. In the same task, we also see that \( F_0 \) conversion performance of DisC was better than that of DisC-NoAux. Tables II and III show that timbre conversion was more successfully done by DisC than by ACVAE-C, whereas \( F_0 \) conversion was more successfully done by DisC than by ACVAE-S (especially in timbre-only conversion task). ACVAE-C and -S achieved the best performance in terms of \( \Delta F_0 \) and the MCD, respectively, however, they could not keep these criteria small at the same time.

**C. Subjective evaluation**

To conduct subjective evaluation, we randomly chose three test samples for each of the quality and timbre evaluations and for each setting (i.e., a combination of \( \beta \), a source speaker, and a conversion method in task T and a combination of a source and target speakers and a conversion method in tasks T and PT). For timbre evaluation, we concatenated target and converted speech samples with a 1 s silent interval. We defined the target speech sample as a randomly selected one of the four options, “1: Bad,” “2: Poor,” “3: Fair,” “4: Good,” and “5: Excellent.” Timbre evaluation was done based on the ABX test for speaker similarity, where he/she chooses one out of the four options, “1: Different (sure),” “2: Different (not sure),” “3: Same (not sure),” and “4: Same (sure),” for each concatenated sample.

We conducted a mean opinion score (MOS) test for sound quality, where each listener scores the sound quality of each sample by choosing one out of the five options, “1: Bad,” “2: Poor,” “3: Fair,” “4: Good,” and “5: Excellent.” Timbre evaluation was done based on the ABX test for speaker similarity, where he/she chooses one out of the four options, “1: Different (sure),” “2: Different (not sure),” “3: Same (not sure),” and “4: Same (sure),” for each concatenated sample.

Tables IV and V, respectively, show the results of the quality and timbre evaluations. In task P, \( F_0 \) conversion by WORLD vocoder achieved the best sound quality, however, DisC could maintain the timbre of the source speaker the best. When converting only timbre (i.e., task T), neither DisC nor DisC-NoAux could achieve as high sound quality and timbre

---

**TABLE I**

| Task   | DisC | DisC-NoAux | WORLD | ACVAE-C | ACVAE-S |
|--------|------|------------|-------|--------|--------|
| MCD    | 0.14 ± 0.37 | 0.33 ± 0.47 | 0.23 ± 0.11 | 0.14 ± 0.12 |
| \( \Delta F_0 \) | 0.22 ± 0.10 | 0.24 ± 0.11 | 0.12 ± 0.11 | 0.56 ± 0.18 |

**TABLE II**

| Task   | DisC | DisC-NoAux | ACVAE-C | ACVAE-S |
|--------|------|------------|--------|--------|
| MCD    | 7.29 ± 0.74 | 7.33 ± 0.75 | 7.62 ± 0.73 | 6.65 ± 0.68 |
| \( \Delta F_0 \) | 0.24 ± 0.14 | 0.30 ± 0.20 | 0.12 ± 0.11 | 0.50 ± 0.18 |

**TABLE III**

| Task   | DisC | DisC-NoAux | ACVAE-C | ACVAE-S |
|--------|------|------------|--------|--------|
| MCD    | 0.84 ± 0.04 | 6.00 ± 0.04 | 7.62 ± 0.74 | 6.65 ± 0.68 |
| \( \Delta F_0 \) | 0.13 ± 0.11 | 0.14 ± 0.11 | 0.12 ± 0.10 | 0.19 ± 0.13 |

**TABLE IV**

| Task   | DisC | DisC-NoAux | WORLD | ACVAE-C | ACVAE-S |
|--------|------|------------|-------|--------|--------|
| P      | 2.99 ± 0.83 | 1.83 ± 0.83 | 1.64 ± 0.58 | 1.59 ± 0.65 |
| F      | 2.06 ± 1.18 | 2.00 ± 1.10 | 1.59 ± 0.85 | 3.41 ± 0.69 |
| T      | 4.34 ± 0.80 | 4.37 ± 0.79 | 2.26 ± 0.88 | 4.41 ± 0.69 |

**TABLE V**

| Task   | DisC | DisC-NoAux | WORLD | ACVAE-C | ACVAE-S |
|--------|------|------------|-------|--------|--------|
| P      | 1.26 ± 0.57 | 1.15 ± 0.57 | 1.10 ± 0.57 | 1.09 ± 0.57 |
| F      | 2.06 ± 1.18 | 1.10 ± 1.10 | 1.09 ± 1.09 | 1.08 ± 1.08 |
| T      | 4.34 ± 0.80 | 4.37 ± 0.79 | 2.26 ± 0.88 | 4.41 ± 0.69 |
maintenance as ACVAE-S. However, by converting $F_0$ pattern and timbre simultaneously (i.e., task PT), they could achieve comparable performance to ACVAE-S. A possible reason for this would be that speaker identity strongly depends on the average fundamental frequency, as also pointed out in [43].

IV. CONCLUSION

We developed a new $F_0$-controllable VC method DisC that takes into account discrepancy between the target and generated $F_0$ patterns in the training process. In this paper, we assumed that the same set of speakers appears in training and test data sets. It would be important to extend DisC so that it can be applied to unseen speakers in the test phase. Another future direction would be to take into account other prosody information (e.g., rhythm and stress) as controllable features.

REFERENCES

[1] S. H. Mohammadi and A. Kain, “An overview of voice conversion systems,” Speech Communication, vol. 88, pp. 65–82, 2017.
[2] A. Kain and M. W. Macon, “Text-to-speech voice adaptation from sparse training data,” in ICSLP, 1998, p. paper 0902.
[3] Z. Inanoglu and S. Young, “Data-driven emotion conversion in spoken English,” Speech Communication, vol. 51, no. 3, pp. 268–283, 2009.
[4] C. Robinson, N. Obin, and A. Roebel, “Sequence-to-sequence modelling of $F_0$ for speech emotion conversion,” in ICASSP, 2019, pp. 6830–6834.
[5] A. Kain and M. W. Macon, “Spectral voice conversion for text-to-speech synthesis,” in ICASSP, 1998, pp. 285–288.
[6] Y. Styliaou, O. Cappe, and E. Moulines, “Continuous probabilistic transform for voice conversion,” TASLP, vol. 6, no. 2, pp. 131–142, 1998.
[7] T. Toda, A. W. Black, and K. Tokuda, “Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory,” TASLP, vol. 15, no. 8, pp. 2222–2235, 2007.
[8] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, “Voice conversion using artificial neural networks,” in ICASSP, 2009, pp. 3893–3896.
[9] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, “Spectral mapping using artificial neural networks for voice conversion,” TASLP, vol. 18, no. 5, pp. 954–964, 2010.
[10] M. V. Ramos, “Voice conversion with deep learning,” Master’s thesis, Instituto Superior Técnico, 2016.
[11] J.-X. Zhang, Z.-H. Ling, L.-J. Liu, Y. Jiang, and L.-R. Dai, “Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory,” TASLP, vol. 27, no. 3, pp. 631–644, 2019.
[12] S. Liu, Y. Cao, S. Kang, N. Hu, X. Liu, D. Su, D. Yu, and H. Meng, “Transferring source style in non-parallel voice conversion,” in Interspeech, 2020, pp. 4721–4725.
[13] W.-C. Huang, Y.-C. Wu, and T. Hayashi, “Any-to-one sequence-to-sequence voice conversion using self-supervised discrete speech representations,” in ICASSP, 2021, pp. 5944–5948.
[14] Y. Y. Lin, C.-M. Chien, J.-H. Lin, H. y. Lee, and L. s. Lee, “Fragmentvc: Any-to-any voice conversion by end-to-end extracting and fusing fine-grained voice fragments with attention,” in ICASSP, 2021, pp. 5939–5943.
[15] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, “Phonetic posteriograms for many-to-one voice conversion without parallel data training,” in ICME, 2016, pp. 1–6.
[16] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in ICLR, 2014.
[17] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, “ACVAE-VC: Non-parallel voice conversion with auxiliary classifier variational autoencoder,” TASLP, vol. 27, no. 9, pp. 1432–1443, 2019.
[18] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, “AutoVC: Zero-shot voice style transfer with only autoencoder loss,” in ICML, vol. 97, 2019, pp. 5210–5219.
[19] K. Qian, Y. Zhang, S. Chang, M. Hasegawa-Johnson, and D. Cox, “Unsupervised speech decomposition via triple information bottleneck,” in ICML, vol. 119, 2020, pp. 7836–7846.