# RHYTHM-FLEXIBLE VOICE CONVERSION WITHOUT PARALLEL DATA USING CYCLE-GAN OVER PHONEME POSTERIORGRAM SEQUENCES

*Cheng-chieh Yeh[1], Po-chun Hsu[1], Ju-chieh Chou[1], Hung-yi Lee[1], Lin-shan Lee[1]*

[1]College of Electrical Engineering and Computer Science, National Taiwan University
{r06942067, b03901071, r06922020, hungyilee}@ntu.edu.tw, lslee@gate.sinica.edu.tw

## ABSTRACT

Speaking rate refers to the average number of phonemes within some unit time, while the rhythmic patterns refer to duration distributions for realizations of different phonemes within different phonetic structures. Both are key components of prosody in speech, which is different for different speakers. Models like cycle-consistent adversarial network (Cycle-GAN) and variational auto-encoder (VAE) have been successfully applied to voice conversion tasks without parallel data. However, due to the neural network architectures and feature vectors chosen for these approaches, the length of the predicted utterance has to be fixed to that of the input utterance, which limits the flexibility in mimicking the speaking rates and rhythmic patterns for the target speaker. On the other hand, sequence-to-sequence learning model was used to remove the above length constraint, but parallel training data are needed. In this paper, we propose an approach utilizing sequence-to-sequence model trained with unsupervised Cycle-GAN to perform the transformation between the phoneme posteriorgram sequences for different speakers. In this way, the length constraint mentioned above is removed to offer rhythm-flexible voice conversion without requiring parallel data. Preliminary evaluation on two datasets showed very encouraging results.

***Index Terms***— voice conversion, sequence-to-sequence learning, unsupervised learning, cycle-gan

## 1. INTRODUCTION

Voice conversion (VC) is a task aiming to convert the speech signals from a certain acoustic domain to another while keeping the linguistic content the same. Examples of considered acoustic domains include not only the speaker identity, but many other factors orthogonal to the linguistic content such as speaking style, speaking rate [1], noise condition, emotion [2, 3], accent [4], etc., with potential applications ranging from speech enhancement [5, 6], computer-assisted pronunciation training for non-native language learner [4], speaking assistance [7], to speaker identity conversion [8, 9, 10, 11], to name a few.

When speaker identity conversion is considered, in addition to the fact that the same phoneme sounds different when produced by different speakers, it is well known that the prosody can also be very different for different speakers. The prosody of speech includes not only the pitch range, but at least the speaking rate and the rhythmic patterns. While speaking rate refers primarily to the average number of phonemes produced within some unit time, the rhythmic pattern refers to the duration distributions for realizations of different phonemes within different phonetic structures. It is obvious that the speaking rate and rhythmic patterns are very different for different speakers. When the goal is to mimic the voice characteristics of a specific speaker, it is important that the prosody including the speaking rate and the rhythmic patterns of the target speaker is reproduced. This is why flexible speaking rates and rhythmic patterns are highly desired for voice conversion (VC).

The many approaches proposed for VC may be in most cases classified into two types: text-independent and text-dependent. Text-independent VC directly predicts the target speech signals based on the source speech signals without considering the linguistic content or text. This is usually achieved with acoustic models such as Gaussian mixture models (GMMs) [8, 12] or deep neural networks (DNNs) [13]. Text-dependent VC, on the other hand, converts speech signals through the textual information. That is, a speech recognizer is used to estimate the textual information from the source speech and a speech synthesizer is used to predict the target speech from the textual information. The conversion units for text-dependent VC are usually rougher (e.g., phonemes, characters or words) than those used in text-independent VC (e.g., frames). Approaches recently proposed using phoneme posteriorgram vectors as the conversion unit [14, 15] may be considered as a compromise between the two, because the posteriorgram probabilities for all possible phonemes in the source speech signals are estimated, converted and used to generate the target speech signals frame by frame.

Typically, text-independent VC requires parallel data. In other words, the data of utterance pairs produced by the source and target speakers for the same sentences are needed to train the conversion model. But recently, methods based

on deep learning using only non-parallel data have been proposed [16, 17, 18]. However, in these approaches due to the limitations of the conversion models or acoustic features used, the utterance length before and after conversion has to be kept the same, so the goal of reproducing the speaking rate and rhythmic patterns of the target speaker is simply impossible to realize. Sequence-to-sequence learning performed on phoneme posteriorgram sequences may be a possible approach to achieve the above mentioned goal [19], but all such approaches reported so far required parallel data. In this paper, we propose a rhythm-flexible VC approach producing target speech signals of variable length but trained with non-parallel data only.

Below in subsection 2.1, we first introduce the related works on text-independent VC using deep learning trained with non-parallel data and the associated length constraint. We then show in subsection 2.2 a primarily text-dependent approach using sequence-to-sequence model to transform between source and target speakers over the phoneme posteriorgram sequences, which overcame the problem of length constraint but required parallel data. In section 3, we therefore present the approach proposed here in this paper using non-parallel data but overcoming the length constraint to offer rhythm-flexible VC. We list model architectures and implementation for this proposed approach in section 4, and show experimental results with evaluations in section 5. Finally, we make some discussions and concluding remarks in section 6.

## 2. RELATED WORK

### 2.1. Non-parallel VC using Deep Learning

Recently, deep generative models such as Variational Autoencoders (VAEs) [20] and Generative Adversarial Networks (GANs) [21] including Conditional GANs (CGANs) [22] were broadly studied because they can be applied to unsupervised learning problems. This is specially attractive for VC because that implies parallel data may not be needed. With VAEs, the encoder first extracts a latent feature representing the speaker-independent linguistic content, and then the decoder is trained to generate the voice of the target speaker conditioned on the latent feature and some extra information regarding the target speaker [16, 17, 23]. With CGANs, with the guidance of the discriminator, the conditional generator tries to generate acoustic features sounding like being produced by the target speaker conditioned on the acoustic features produced by the source speaker. Among the many extensions of CGANs, cycle-consistent adversarial network (Cycle-GAN) [24] and Star-GAN [25] have been very successfully used as domain translators between the source and target domains, and have been used for VC [18, 26, 27, 28].

Although the above approaches are able to perform voice conversion without parallel data, the length of the generated signals are locked to be the same as that of the input signals due to the neural network architectures or the acoustic features used. For example, some of them used combinations of recurrent neural networks (RNNs) and convolutional neural networks (CNNs) [16, 17, 23] rather then the sequence-to-sequence encoder-decoder architecture. These methods took a single frame or a segment of frames (e.g. 128 frames) as the input, and then generated a single frame or a segment of frames with the same length as the output. Some other approaches chose Mel-cepstral coefficients (MCEPs), logarithmic fundamental frequency (log F0), and aperiodicities (APs) as the features, but the conversion was performed on MCEPs only [18, 28]. The converted MCEPs have to be of the same length as the original ones in order to be aligned with the sequences of log F0 and APs when synthesizing back to the waveform. This fixed-length constraint makes it impossible for these very attractive deep learning approaches not requiring parallel data to be rhythm-flexible to better catch the prosody of the target speaker.

### 2.2. Sequence-to-sequence Conversion over Posteriorgram Sequences Trained with Parallel Data

An approach utilizing Recurrent Neural Networks (RNNs) encoder-decoder for sequence-to-sequence learning [29] transforming the phoneme posteriorgram sequences between different speakers that can overcome the length constraint mentioned above was proposed [19]. In this approach, in addition to a speech recognizer to produce the phoneme posteriorgram sequences and a speech synthesizer to reconstruct the signals, a module for transformation between the phoneme posteriorgram sequences for the source and target speakers was added in between to perform VC. This latter transformation module includes an RNN encoder and an RNN decoder operating frame by frame. The end-of-sequence token $<EOS>$ produced at the RNN decoder at any time removed the length constraint mentioned above and offered more flexible rhythm for the output speech. However, the supervised training for sequence-to-sequence learning requires parallel data. This leads to the new approach proposed in this paper, which offers rhythm-flexible VC with variable length but doesn't require parallel data, as is presented below.

## 3. PROPOSED APPROACH

The approach proposed here successfully overcomes the length constraint mentioned in subsection 2.1 and removes the need for parallel data mentioned in subsection 2.2 by adopting Cycle-GAN, which is an unsupervised style transfer model capable of transforming the phoneme posteriorgram sequences between speakers. The three components of the approach is respectively presented in subsections 3.1, 3.2, and 3.3 and Figure 1 (a)(b)(c), the Cycle-GAN in subsection 3.4 and Figure 2, while the overview of the whole VC process is in Figure 3.
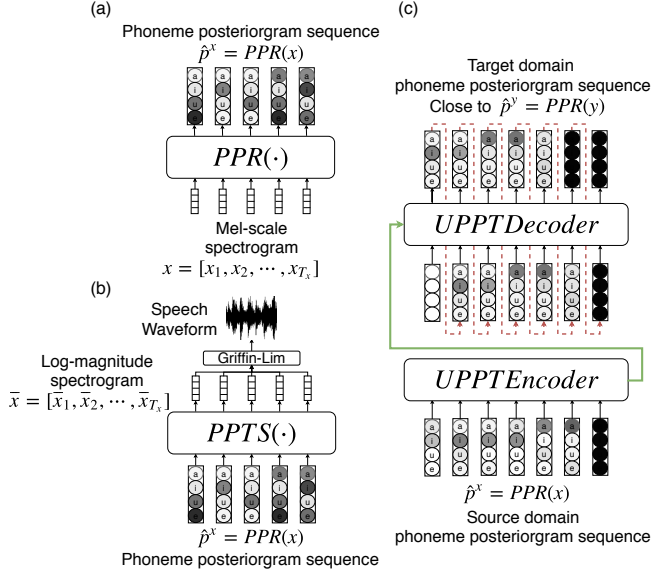
**Fig. 1**. The three components of the proposed approach. $PPR(\cdot)$ in (a) stands for Phoneme Posteriorgram Recognizer and $PPTS(\cdot)$ in (b) for Phoneme-Posteriorgram-to-Speech Synthesizer. $UPPT$ in (c) stands for Unsupervised Phoneme Posteriorgram Transformer, which includes an encoder and a decoder. Dotted red arrows around the $UPPT$ decoder indicates the output at the previous time index is used as the input at the next time index. The green arrow indicates the final state of the encoder is fed to the initial state of the decoder.

### 3.1. Phoneme Posteriorgram Recognizer

As in Figure 1 (a), let $x = [x_1, x_2, \cdots, x_{T_x}]$ and $y = [y_1, y_2, \cdots, y_{T_y}]$ be the acoustic feature vector sequences from the source and target speaker domains, $x_t$ and $y_t$ be the feature vector at time index $t$, and $T_x$ and $T_y$ be the lengths of $x$ and $y$. In Figure 1 (a), $x$ and $y$ are the Mel-scale spectrogram. Also, $l^x = [l_1^x, l_2^x, \cdots, l_{T_x}^x]$ and $l^y = [l_1^y, l_2^y, \cdots, l_{T_y}^y]$ are the ground truth label phoneme sequences corresponding to $x$ and $y$, respectively. Phoneme Posteriorgram Recognizer $PPR(\cdot)$ is a speaker-independent neural network that estimates the phoneme posterior probabilities frame by frame given an acoustic feature vector sequence. This recognizer $PPR(\cdot)$ is trained to minimize $L_{xent}(l^x, PPR(x))$ and $L_{xent}(l^y, PPR(y))$, which are the cross-entropy between the ground truth label sequences (an one-hot vector for each time $t$) and the estimated phoneme posteriorgram sequences for data in both source and target speaker domains.

### 3.2. Phoneme-Posteriorgram-to-Speech Synthesizer

As in Figure 1 (b), Phoneme-Posteriorgram-to-Speech Synthesizers $PPTS_x(\cdot)$ and $PPTS_y(\cdot)$ are the reverse process of $PPR(\cdot)$ for data in the source and target speaker domains respectively, or two neural networks that predict the speech feature vectors $\bar{x}$ and $\bar{y}$ frame by frame given the phoneme posteriorgram sequences $\hat{p}^x = PPR(x)$ and $\hat{p}^y = PPR(y)$. In Figure 1 (b), $\bar{x}$ and $\bar{y}$ are the log-magnitude version of $x$ and $y$. Griffin-Lim is the algorithm synthesizing the speech waveform from the predicted log-magnitude version $\bar{x}$ and $\bar{y}$ [30]. $PPTS_x(\cdot)$ and $PPTS_y(\cdot)$, are respectively trained to minimize the mean squared error between the ground truth speech feature vectors and the reconstructed version, $L_{mse}(\bar{x}, PPTS_x(\hat{p}^x))$ and $L_{mse}(\bar{y}, PPTS_y(\hat{p}^y))$.

### 3.3. Unsupervised Phoneme Posteriorgram Transformer

As shown in Figure 1 (c), the Unsupervised Phoneme Posteriorgram Transformer $UPPT$ is an attention-based sequence-to-sequence model including an $UPPT$ encdoer and an $UPPT$ decoder, which transforms a source domain posteriorgram sequence $\hat{p}^x = PPR(x)$ into another posteriorgram sequence very close to those for signals in the target domain, $\hat{p}^y = PPR(y)$. The green arrow indicates the final state of the encoder is fed to the initial state of the decoder, and the dotted red arrows around the $UPPT$ decoder indicate the output of the previous time index is used as the input at the next time index. This is a sequence-to-sequence model used to remove the length constraint and achieve the rhythm-flexible VC mentioned previously.

### 3.4. Cycle-GAN

Let $X$ and $Y$ be the two sets that contain all estimated phoneme posteriorgram sequences $\hat{p}^x$, $\hat{p}^y$ from the source and target speaker domains respectively. We adopt here the cycle-consistent generative adversarial network (Cycle-GAN) to learn the mapping between $X$ and $Y$ without paired data. As shown in Figure 2, the whole training procedure includes two sets of generative adversarial networks (GANs), each with a generator and a discriminator. After Cycle-GAN training, two generators, $G_{X \to Y}$ and $G_{Y \to X}$ are obtained. These two generators are two transformers ($UPPT$s in subsection 3.3) that transform $\hat{p}^x$ to $\hat{p}^{x \to y}$ ($\hat{p}^{x \to y} = G_{X \to Y}(\hat{p}^x)$) and $\hat{p}^y$ to $\hat{p}^{y \to x}$ ($\hat{p}^{y \to x} = G_{Y \to X}(\hat{p}^y)$) respectively, where $\hat{p}^{x \to y}$ is a phoneme posteriorgram sequence mapped from the source domain to target domain and $\hat{p}^{y \to x}$ vice versa. Two discriminators are also trained, $D_X$ and $D_Y$, to discriminate whether a phoneme posteriorgram sequence is a real one generated from a signal in a domain, or a fake one transformed from another domain.

#### 3.4.1. Training Goal of Generators (or UPPTs)

The generators $G_{X \to Y}$, $G_{Y \to X}$ take the phoneme posteriorgram sequences from a speaker domain as the input and produce another phoneme posteriorgram sequence close to those for another speaker domain. Both are built with attention-based sequence-to-sequence model to learn the mapping between $\hat{p}^x \in X$ and $\hat{p}^y \in Y$ such that the distribution of
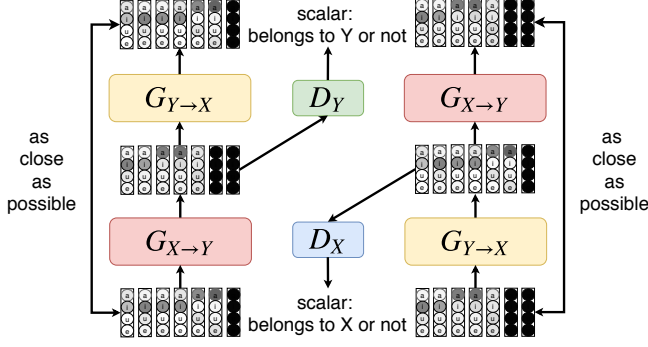
**Fig. 2**. Cycle-GAN. $G_{X\to Y}$ and $G_{Y\to X}$ refer to generators. $D_X$ and $D_Y$ refer to discriminators. Blocks with the same color share the same set of neural network parameters. Each generator is built with a pair of $UPPT$ encoder and $UPPT$ decoder.

$G_{X\to Y}(\hat{p}^x)$ is as indistinguishable from that of $\hat{p}^y$ as possible, and $G_{Y\to X}(\hat{p}^y)$ is as indistinguishable from $\hat{p}^x$ as possible. The training of the generators are guided by the discriminators described below to achieve the above scenario.

### 3.4.2. Training Goal of Discriminators

The discriminators $D_X$ and $D_Y$ take a phoneme posteriorgram sequence as the input, and produce a scalar indicating how "real" the input is from the sets $X$ or $Y$ for the domain considered, or actually fake or transformed from another domain. Such discriminators are to guide the generators. So the training objective of discriminators are to distinguish between the real sequences such as $\hat{p}^x$, $\hat{p}^y$ and fake sequences such as $G_{Y\to X}(\hat{p}^y)$, $G_{X\to Y}(\hat{p}^x)$ generated by the generators, and give higher scores to real ones and lower scores to fake ones.

### 3.4.3. Objective functions

Several objective functions are defined here as given below.
1. Adversarial Loss: Adversarial losses [21] are applied to both mapping functions, $G_{X\to Y}$ and $G_{Y\to X}$. For the mapping function $G_{X\to Y}$ and its discriminator $D_Y$, we express the objective as in (1).

$$
\begin{aligned}
L_{GAN}(G_{X\to Y}, D_Y) = {}& E_{y\sim Y}[\log D_Y(y)] \\
& + E_{x\sim X}[\log(1 - D_Y(G_{X\to Y}(x)))].
\end{aligned} \tag{1}
$$

$L_{GAN}(G_{Y\to X}, D_X)$ is defined in exactly the same way as (1), except the roles of $X$ and $Y$ are reversed.

2. Cycle Consistency Loss: Cycle consistency losses [24] are applied when training the two generators. The transform cycle should be able to bring $x$ back to the original phoneme posteriorgram sequence, i.e. $G_{Y\to X}(G_{X\to Y}(x)) \approx x$ and $G_{X\to Y}(G_{Y\to X}(y)) \approx y$. We express this objective as in (2).
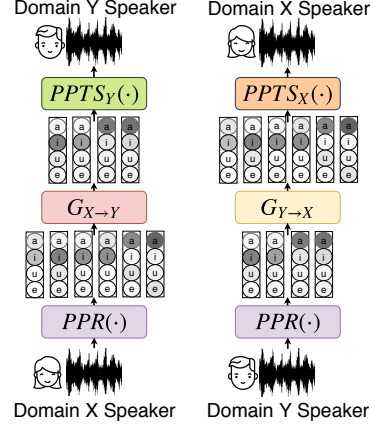


**Fig. 3**. The complete voice conversion process for the proposed approach. Blocks with the same color share the same set of neural network parameters.

(Note that $L_{xent}$ in (2) means cross-entropy)

$$
\begin{aligned}
L_{cycle}(G_{X\to Y}, G_{Y\to X}) = {}& E_{x\sim X}[L_{xent}(x, G_{Y\to X}(G_{X\to Y}(x)))] \\
& + E_{y\sim Y}[L_{xent}(y, G_{X\to Y}(G_{Y\to X}(y)))].
\end{aligned} \tag{2}
$$

3. Identity Mapping Loss: Identity mapping loss as proposed in the original work of Cycle-GAN [24] is also used here. When real samples of the target domain are provided as the input to the generator, the transformed result should be as close to the input as possible. It was found that adding this objective as an extra regularization term for the generators actually improved the transformed results. We express this objective as in (3).

$$
\begin{aligned}
L_{identity}(G_{X\to Y}, G_{Y\to X}) = {}& E_{x\sim X}[L_{xent}(x, G_{Y\to X}(x))] \\
& + E_{y\sim Y}[L_{xent}(y, G_{X\to Y}(y))].
\end{aligned} \tag{3}
$$

The full objective for Cycle-GAN training is the sum of (1)(2)(3):

$$
\begin{aligned}
L_{cycle\_gan}(G_{X\to Y}, G_{Y\to X}, D_X, D_Y) = {}& \\
L_{GAN}(G_{X\to Y}, D_Y) + L_{GAN}(G_{Y\to X}, D_X) & \\
+\lambda_1 L_{cycle}(G_{X\to Y}, G_{Y\to X}) + \lambda_2 L_{identity}(G_{X\to Y}, G_{Y\to X}), &
\end{aligned} \tag{4}
$$

where $\lambda_1, \lambda_2$ are balancing parameters. So overall we aim to solve:

$$
G^*_{X\to Y}, G^*_{Y\to X} = \arg \min_{G_{X\to Y}, G_{Y\to X}} \max_{D_X, D_Y} (L_{cycle\_gan}). \tag{5}
$$

### 3.5. Overall Voice Conversion

As shown in Figure 3, the overall voice conversion is achieved by first passing the source speaker's speech signal through $PPR(\cdot)$ in subsection 3.1 to obtain its phoneme posteriorgram sequence, then using $UPPT$ in subsection 3.3 to

transform it to the target domain, and finally using $PPTS(\cdot)$ in subsection 3.2 trained on the target domain to synthesize the target speech signal from the given phoneme posteriorgram sequence.

## 4. MODEL IMPLEMENTATION AND TRAINING

We adopted primarily the model architecture from the CBHG module [31] for all three parts of the proposed approach including the phoneme posteriorgram recognizer $PPR(\cdot)$, phoneme-posteriorgram-to-speech synthesizer $PPTS(\cdot)$ and the unsupervised phoneme posteriorgram transformer $UPPT$ (or the generators). The convolution-bank in CBHG module was found to be able to better capture the local information over time, reduce overfitting and generalize well to long and complex inputs, such as acoustic feature sequences [32]. As previously suggested [18], for the discriminators, we treated the input phoneme posteriorgram sequences as pictures with channel size one, and performed several 1D-convolution layers with strides larger then one for better capturing the local properties such as how many frames a speaker usually needs to produce a specific phoneme. The attention mechanism in $UPPT$ was shown to be able to effectively improve the decoder's prediction [33]. The overall model architecture and training details are available[1] but left out here for space limitation. We considered $PPR(\cdot)$ as a pseudo-labeler. We first trained $PPR(\cdot)$ with the objective mentioned in subsection 3.1, and then trained $PPTS(\cdot)$ with the objective mentioned in subsection 3.2. With the above done, we then collected the estimated results of $PPR(\cdot)$ to train $UPPT$.

For $PPR(\cdot)$, we used mel-scale spectrogram as the input acoustic features, and the phoneme set defined in Carnegie Mellon pronouncing dictionary [34] as the labels for the posteriorgram sequences. Thus the input to $UPPT$ were sequences of vectors with dimension 70 (39 phoneme types with stress combinations, each treated as mono-phoneme). For $PPTS$, we used log-magnitude spectrogram as the output acoustic features, over which Griffin-Lim algorithm [30] was applied to synthesize the waveform. All other detailed setting followed the previous work [32].

## 5. EXPERIMENTS AND RESULTS

### 5.1. Experimental setup

We used two datasets under a fully non-parallel setting. One is Librispeech [36], an audio book read by multi-speakers. The other one is VCTK [37], which is a multi-speaker dataset primarily reading newspapers and elicitation paragraphs intended to identify the speaker's accent. Both datasets were randomly split into training, validation and testing sets with percentages of 80%, 10% and 10%. The phone boundaries were not available in both datasets, so we used a force-aligner
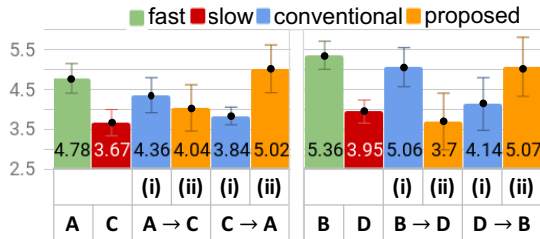
**Fig. 4**. Average speaking rates (number of syllables / sec) for utterances in testing set before and after conversion. The dots and bars indicate the averages and the standard deviations. Speakers A, B belonged to the fastest speaking group and C, D to the slowest speaking group, (i) achieved by the conventional method [35] while (ii) by the proposed approach. The numbers shown are the averages.

pretrained on Librispeech dataset [38] to get the phone boundaries and corresponding phone classes for training $PPR(\cdot)$.

#### 5.1.1. Librispeech Dataset

Using Praat Script Syllable Nuclei [39] to measure the speaking rate, we picked the fastest 20 speakers and the slowest 20 speakers to form a subset with a total length of 15.8 hours or 4609 utterances for evaluation of conversion across different speaking rates. When training the three components in Figure 1, we used all the 40 speakers to train the speaker-independent $PPR(\cdot)$, the grouped fastest and slowest 20 speakers as two domains to initialize the Cycle-GAN training for $UPPT$ (followed by individual training for each conversion pair), and individually trained speaker-dependent $PPTS(\cdot)$ for each speaker.

#### 5.1.2. VCTK Dataset

We chose 18 speakers, some native and some non-native, with a total of 7.3 hours or 7132 utterances, as a different scenario for rhythmic patterns. We used all the 18 speakers to train the speaker-independent $PPR(\cdot)$, but trained the $UPPT$ and $PPTS(\cdot)$ for each conversion pair individually.

### 5.2. Objective Evaluation

To show the proposed approach is able to learn the speaking rates of the target speakers, we chose two speakers A, B (with IDs 6925 and 460) from the fastest speaking group of Librispeech and two speakers C, D (with IDs 163 and 1363) from the slowest speaking group and performed the conversions A $\leftrightarrow$ C and B $\leftrightarrow$ D on the utterances in their testing sets using a conventional method [35] and the proposed approach. This is actually an ablation study, since the only difference between the two is whether the $UPPT$ proposed here was used or not. The results are plotted in Figure 4, where the averages
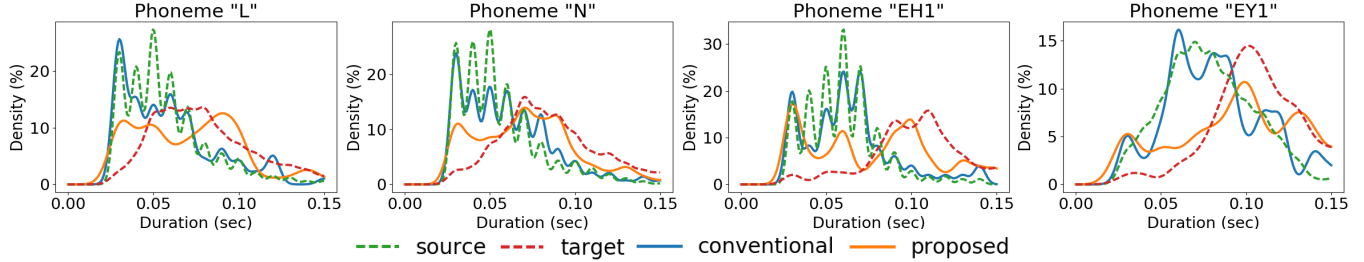
**Fig. 5**. Example rhythmic patterns (duration distributions) for phonemes "L", "N", "EH1", "EY1". The histograms were normalized by Gaussian kernel density estimation (bandwidth=0.125). Different colors represent the rhythmic patterns for different speakers (green and red) and converted voice (blue and brown).

and standard deviations of the speaking rates are shown for the two approaches. We can see from Figure 4 the proposed approach could mimic the speaking rates of the target speaker much better.

To show the proposed approach is capable of learning the rhythmic patterns (phoneme duration distributions) for the target speaker, we chose a pair of speakers E, F (with IDs p231 and p265) from VCTK and performed the conversion E → F on their testing utterances. We used the pretrained force-aligner to obtain the phoneme duration and normalized the histograms by Gaussian kernel density estimation. The example rhythmic patterns for two vowels and two consonants are plotted in Figure 5, in which the different colors are respectively for source and target speakers (green and red) and the converted voice by a conventional [35] (blue) and the proposed (brown) approaches. We can see from Figure 5 different speakers did show very different rhythmic patterns, and the proposed approach was able to mimic these patterns of the target speaker much better.

### 5.3. Subjective Evaluation

Subjective evaluation was performed on converted voice (including both intra-gender and inter-gender conversions) from VCTK datasets. In the binary preference test for speaker similarity, 20 subjects were given pairs of converted voice in random order and asked to choose one sounding more similar to a reference target utterance produced by the target speaker, comparing the proposed approach to a recently proposed non-parallel VC by Chou et al. [17] and the conventional method [35]. The results are in Figure 6 (a)(b). We can see the proposed approach obviously outperformed the two previous approaches in terms of speaker similarity.

The MOS for naturalness in Figure 6 (c) shows the proposed approach is better than the conventional method [35], although not as good as the recently proposed non-parallel VC [17], very probably because of the mean square error (MSE) objective function used in training $PPTS(\cdot)$ in subsection 3.2. It was found that models trained with MSE objective tend to output average predictions [40], which
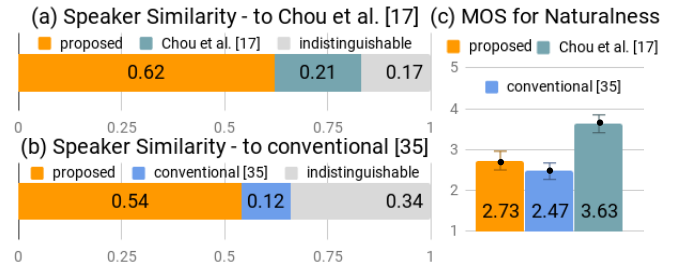


**Fig. 6**. Subjective evaluation results: binary preference test for speaker similarity compared to (a) the recently proposed non-parallel VC with length constraint (Chou et al. [17]), (b) the conventional method [35] (ablation study); and (c) 5-scale naturalness MOS scores similarly compared.

may lead to over-smoothed log-magnitude spectrograms and blurred sounds after the Griffin-Lim algorithm. Investigations for replacing Griffin-Lim vocoder with a neural vocoder [41] or applying post-filters to enhance the output log-magnitude spectrograms are under progress. Another possible direction may be applying sequence-to-sequence Cycle-GAN directly on log-magnitude spectrograms rather than on the phoneme posteriorgram sequences, but at the difficulties of the high feature dimension and complex model structures.

## 6. CONCLUSION

Objective and subjective evaluation on two different datasets showed that the proposed approach is able to mimic the voice characteristics of a target speaker, including the speaking rate and rhythmic patterns, without parallel data by utilizing sequence-to-sequence learning trained with Cycle-GAN to remove the length constraint. Although phoneme boundaries are needed for the training data, an easily obtained pretrained force-aligner can offer these boundaries.

# 7. REFERENCES

[1] Dimitrios Rentzos, S Vaseghi, E Turajlic, Qin Yan, and Ching-Hsiang Ho, "Transformation of speaker characteristics for voice conversion," in *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*. IEEE, 2003, pp. 706–711.

[2] Ryo Aihara, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki, "Gmm-based emotional voice conversion using spectrum and prosody features," *American Journal of Signal Processing*, vol. 2, no. 5, pp. 134–138, 2012.

[3] Hiromichi Kawanami, Yohei Iwami, Tomoki Toda, Hiroshi Saruwatari, and Kiyohiro Shikano, "Gmm-based voice conversion applied to emotional speech synthesis," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[4] Keisuke Oyamada, Hirokazu Kameoka, Takuhiro KANEKO, Hiroyasu ANDO, Kaoru HIRAMATSU, and Kunio KASHINO, "Non-native speech conversion with consistency-aware recursive network and generative adversarial network," in *Proceedings of APSIPA Annual Summit and Conference*, 2017, vol. 2017, pp. 12–15.

[5] Alexander B Kain, John-Paul Hosom, Xiaochuan Niu, Jan PH van Santen, Melanie Fried-Oken, and Janice Staehely, "Improving the intelligibility of dysarthric speech," *Speech communication*, vol. 49, no. 9, pp. 743–759, 2007.

[6] Frank Rudzicz, "Acoustic transformations to improve the intelligibility of dysarthric speech," in *Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies*. Association for Computational Linguistics, 2011, pp. 11–21.

[7] Keigo Nakamura, Tomoki Toda, Hiroshi Saruwatari, and Kiyohiro Shikano, "Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech," *Speech Communication*, vol. 54, no. 1, pp. 134–146, 2012.

[8] Yannis Stylianou, Olivier Cappé, and Eric Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[9] Alexander Kain and Michael W Macon, "Spectral voice conversion for text-to-speech synthesis," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*. IEEE, 1998, vol. 1, pp. 285–288.

[10] Daisuke Saito, Keisuke Yamamoto, Nobuaki Minematsu, and Keikichi Hirose, "One-to-many voice conversion based on tensor representation of speaker space," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[11] Tomi Kinnunen, Lauri Juvela, Paavo Alku, and Junichi Yamagishi, "Non-parallel voice conversion using i-vector plda: Towards unifying speaker verification and transformation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5535–5539.

[12] Tomoki Toda, Alan W Black, and Keiichi Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[13] Srinivas Desai, E Veera Raghavendra, B Yegnanarayana, Alan W Black, and Kishore Prahallad, "Voice conversion using artificial neural networks," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 3893–3896.

[14] Lifa Sun, Hao Wang, Shiyin Kang, Kun Li, and Helen M Meng, "Personalized, cross-lingual tts using phonetic posteriorgrams.," in *INTERSPEECH*, 2016, pp. 322–326.

[15] Feng-Long Xie, Frank K Soong, and Haifeng Li, "A kl divergence and dnn-based approach to voice conversion without parallel training sentences.," in *INTERSPEECH*, 2016, pp. 287–291.

[16] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," *arXiv preprint arXiv:1704.00849*, 2017.

[17] Ju-chieh Chou, Cheng-chieh Yeh, Hung-yi Lee, and Lin-shan Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," *arXiv preprint arXiv:1804.02812*, 2018.

[18] Takuhiro Kaneko and Hirokazu Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv preprint arXiv:1711.11293*, 2017.

[19] Hiroyuki Miyoshi, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," *arXiv preprint arXiv:1704.02360*, 2017.

[20] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[22] Mehdi Mirza and Simon Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[23] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2016 Asia-Pacific*. IEEE, 2016, pp. 1–6.

[24] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint arXiv:1703.10593*, 2017.

[25] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," *arXiv preprint arXiv:1711.09020*, 2017.

[26] Yang Gao, Rita Singh, and Bhiksha Raj, "Voice impersonation using generative adversarial networks," *arXiv preprint arXiv:1802.06840*, 2018.

[27] Fuming Fang, Junichi Yamagishi, Isao Echizen, and Jaime Lorenzo-Trueba, "High-quality nonparallel voice conversion based on cycle-consistent adversarial network," *arXiv preprint arXiv:1804.00425*, 2018.

[28] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion with star generative adversarial networks," *arXiv preprint arXiv:1806.02169*, 2018.

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[30] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[31] Jason Lee, Kyunghyun Cho, and Thomas Hofmann, "Fully character-level neural machine translation without explicit segmentation," *arXiv preprint arXiv:1610.03017*, 2016.

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

[33] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[34] Robert Weide, "The carnegie mellon pronouncing dictionary [cmudict. 0.6]," 2005.

[35] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *Multimedia and Expo (ICME), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[36] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[37] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al., "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2017.

[38] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger, "Montreal forced aligner: trainable text-speech alignment using kaldi," in *Proceedings of interspeech*, 2017.

[39] Nivja H De Jong and Ton Wempe, "Praat script to detect syllable nuclei and measure speech rate automatically," *Behavior research methods*, vol. 41, no. 2, pp. 385–390, 2009.

[40] Takuhiro Kaneko, Hirokazu Kameoka, Nobukatsu Hojo, Yusuke Ijima, Kaoru Hiramatsu, and Kunio Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4910–4914.

[41] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR abs/1609.03499*, 2016.