# Vocoder-free End-to-End Voice Conversion with Transformer Network

June-Woo Kim
*Department of Sensor and Display Engineering*
*Kyungpook National University*
Daegu, Republic of Korea
kaen2891@gmail.com

Ho-Young Jung
*Department of Artificial Intelligence*
*Kyungpook National University*
Daegu, Republic of Korea
hoyjung@knu.ac.kr

Minho Lee
*School of Electronics Engineering*
*Kyungpook National University*
Daegu, Republic of Korea
mholee@gmail.com

***Abstract*—Mel-frequency filter bank (MFB) based approaches have the advantage of higher learning speeds compared to using the raw spectrum due to a smaller number of features. However, speech generators with the MFB approach require an additional computationally expensive vocoder for the training process. The pre- and post-processing needed by the MFB and the vocoder is not essential to convert human voices, because it is possible to use only the raw spectrum to generate different style of voices with clear pronunciation. In this paper, we introduce a vocoder-free end-to-end voice conversion method using a transformer network to alleviate the computational burden from additional pre- and post-processing. Our transformer-based architecture, which does not have any CNN or RNN layers, has shown the benefit of learning fast while solving the limitation of sequential computation of the conventional RNN. For this reason, our model is a fast and effective approach to convert realistic voices using raw spectra in a parallel manner to generate different style of voices with clear pronunciation. Furthermore, we can get an adapted MFB for speech recognition by multiplying the converted magnitude with the phase information, and therefore our conversion model is also suitable for speaker adaptation. We perform our voice conversion experiments on TIDIGITS-dataset using the naturalness, similarity, and clarity with Mean Opinion Score as metrics.**[1]

***Index Terms*—voice conversion, vocoder-free, transformer, spectrum, phase**

## I. Introduction

Voice conversion has gained considerable attention in various industrial areas. Recently, encoder-decoder models built with recurrent neural networks (RNNs), such as the long short-term memory (LSTM) [1], bidirectional long-short term memory (BiLSTM) [2], and gated recurrent unit (GRU) [3] have been widely utilized for sequence modelling. There are several neural network models based on the RNN encoder-decoder structure, also known as sequence-to-sequence (Seq2Seq) [4], that have achieved good results for voice conversion tasks.

RNNs, however, process words one by one. This sequential property can be an obstacle for parallel computation on GPUs and results in slower training. Furthermore, if the temporal relationships are long, the model tends to forget distant data points or mixes them with the subsequent data. The transformer network [5] partially solved these problems of RNNs by using an attention mechanism to derive global dependency between input and output, which reached state-of-the-art performance in many fields. The transformer, which does not have any convolutional (CNN) [6] or recurrent layers, has shown the advantage of learning fast and eliminates the problem of sequential computation imposed by the conventional RNN.

Given a speech waveform as the input for voice conversion, the short-time Fourier transform (STFT) converts it into a raw spectrum in time-frequency domain form. This spectrum computed with the STFT can provide more useful information than the plain waveform. The conventional approaches used in text-to-speech (TTS), voice conversion, and speech recognition, obtain a Mel-frequency filter bank (MFB, also called Mel-spectrogram) from the raw spectrum after the STFT. This raw spectrum is then compressed according to the Mel curve [7] reflecting the characteristics of the Cochlea in the human ear. The phase information is removed when the spectrum is compressed via the Mel curve.

The MFB, which consists of only 40 to 80 feature dimensions per time step, has the advantage of higher learning speed compared to raw spectrum. However, it cannot be converted directly to waveform speech because of the lost phase information. Thus, speech generators with MFB approach require additional computationally expensive vocoder for the training process. In other words, MFB fed into the Seq2Seq must be synthesized to natural speech through phase estimation with the help of a vocoder which synthesizes the linear scale spectrum. Only then, it is possible to get the final output of the model into waveform speech.

Thus, speech generators with MFB approach require additional vocoder that demands a computationally heavy training process. Although, the voice quality may be better when using a vocoder such as Griffin-Lim [8] or WaveNet [9], it is necessary to consider the problems with complexity due to the extra computation.

The goal of this paper is to achieve smaller computational cost and clear converted voices without a vocoder. In this paper, we introduce a vocoder-free end-to-end voice conversion method using transformer network to alleviate the computational burden from additional pre- and post-processing. Our model is a fast and effective approach to convert realistic voices using raw spectra in a parallel manner to generate different style of voices with clear pronunciation. We focus

[1]Codes are available at https://github.com/kaen2891/kaen2891.github.io

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

on converting the raw spectrum obtained by the STFT without the help of a vocoder, which would require iterative synthesis. In addition, it is possible to use phase information to restore the waveform speech through inverse STFT.

Our conversion model can also be used in speaker adaptation for speech recognition. Our approach can convert the source voice to a target voice without using MFB or vocoder. We can get an adapted MFB for speech recognition by multiplying the converted magnitude with the corresponding phase. Furthermore, it is also possible to convert the voices of minors, elderly, speakers with dialects or those with speech impediments to those of the typical. Through this speaker adaptation, our model can achieve better speech recognition performance. We perform our voice conversion experiments on TIDIGITS-dataset using the naturalness, similarity, and clarity with Mean Opinion Score (MOS) as metrics.

## II. Related Work

In this section, we introduce the prior research on vocoder, voice conversion, and the transformer network used in this paper.

### A. Vocoder

Vocoder is used to synthesize linear scale spectrum into speech signals by synthesizing natural speech through phase estimation. In Griffin-Lim algorithm [8], the STFT of the speech signal output in the previous step is calculated and the amplitude is replaced by the modified-STFT magnitude given as input. This algorithm recovers speech signals with the STFT magnitude that is the most similar to a given modified-STFT through an iterative process of restoring the original signal by minimizing the squared error of the amplitudes between the new STFT and the modified STFT given as input.

WaveNet [9] is an autoregressive model that uses sequential features between speech samples and has succeeded in synthesizing high quality speech by predicting the next sample using previous samples. However, the rate of the generation is slow because each sample is generated one by one from the previous samples. Parallel WaveNet [10], which uses inverse autoregressive flow (IAF) to synthesize voices, is designed to solve the WaveNet's slow sample generation. Since IAF does not know the distribution of the target voice data set during training, the learning is performed by extracting the distribution information of the target data set using a well-trained WaveNet (teacher network) and comparing it with the result of IAF. It has the advantage of faster speech synthesis than WaveNet, but the drawback of lower synthesized speech quality. Unlike the parallel WaveNet [10], WaveGlow [11] does not require a pre-trained teacher network and has the advantage of fast voice synthesis. However, since WaveGlow uses a distribution based loss function, the quality of synthesized speech is poor. Furthermore, when combined with TTS, it poses the problem that the quality of synthesized speech depends on the quality of the MFB predicted from the text.

### B. Voice Conversion

In Parrotron [12], the voices of speaker with a disability are converted into generic voices. The encoder consists of CNNs and three BiLSTMs, while the decoder consists of two LSTMs. The model uses attention between the encoder-decoder. In order to solve the problem of signal-to-signal conversion, the auxiliary speech recognition decoder is connected to the encoder output for multitask learning [13] and is used only while training.

Usually, in order to translate between voices in different languages and synthesize the translated output as speech, the data had to go through speech recognition, translation, and TTS tasks. In this paper which called Translatotron [14], however, they convert the speech of different languages with an end-to-end attention based Seq2Seq network. The model can directly translate the speech of one language into another without going through other steps. The encoder is composed of 8 BiLSTMs, and its output is used to predict the phoneme temporal information of the input and the target through auxiliary tasks. Likewise, in Parrotron, these auxiliary decoders were used only while training. In addition to this, the decoder can be optionally adjusted according to the speaker. Thus, the voice can be converted to the desired speaker's voice by using pre-trained speaker encoder. They used the WaveRNN vocoder [15] rather than Griffin-Lim because it dramatically improves the voice quality.

### C. Transformer network

RNN is widely used method for sequence modeling tasks such as neural translation and language modeling. RNNs, however, process words one by one. This sequential process can be an obstacle for parallelization and leads to slow learning. Furthermore, if the temporal relationships are long, the model tends to forget distant data points or mixes them with the subsequent data. The transformer network [5] relies entirely on attention mechanisms to derive global dependencies between inputs and outputs. As Fig. 1 shows, the transformer model architecture without CNN and RNN have shown the advantage of fast learning. The shortcomings of traditional RNN due to poor performance in long temporal dependencies, have been solved with self-attention. BERT [16], which evolved from transformer, is used in many natural language processing (NLP) tasks including translation, summary and prediction of sentence relevance. BERT is used in other fields too. VideoBERT [17] learned a two-way joint distribution of visual and linguistic token sequences derived from bidirectional vectors for speech recognition from video data. This has led to the research in a variety of tasks, including action classification and video captions. In [18], combination transformer network with TTS model called Tacotron2 [19] is used to present the results of speech synthesis. In [18], as well as in [20], voice conversion is performed based on the transformer network. Especially, the later one perform voice transformation with pre-trained model parameters using vocoder-based synthesis.

Just like [18], [20], we use transformer network for voice conversion due to its generalization performance through self-

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

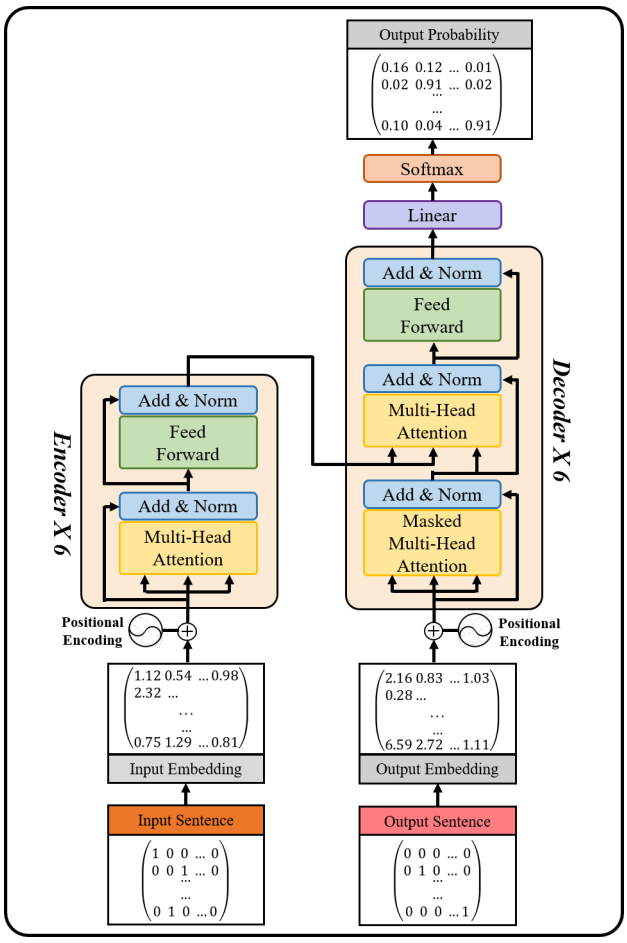


Fig. 1. Vanila transformer network.

attention as well as fast and effective parallel learning techniques. In these methods, a vocoder is also used to improve the quality of speech synthesis. The improved quality however comes with the cost of additional computation required for the synthesis. Therefore we perform our experiments by focusing on the conversion of raw spectrum stage without adopting the voice synthesis method through the vocoder. More details are given in Section 3.

## III. Method

This section introduces the usage of raw spectrum rather than MFB for end-to-end voice conversion without the help of a vocoder.

### A. Raw spectrum

Fig. 2 shows a flowchart of the conversion of a waveform speech into spectrum, MFB, and back to waveform speech. Given a continuous audio signal $x[n]$, this can be expressed as:

$$x[n] = Acos(\omega nT + \phi) = Acos(2\pi fnT + \phi) \quad (1)$$

where $A$ is amplitude, $\omega$ is angular frequency in radians/seconds, $f$ is $\omega/2\pi$, $\phi$ is initial phase in radian, $n$ is time index, and $T$ is $\frac{1}{f_s}$, respectively. The signal is then processed

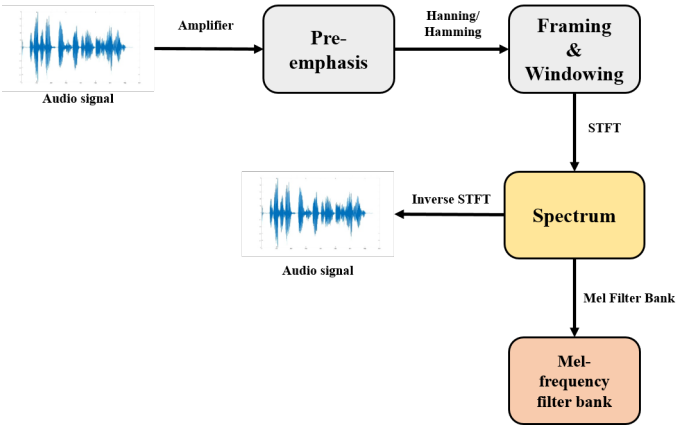


Fig. 2. The steps required to obtain the spectrum and Mel-frequency filter banks from an audio signal.

by applying a pre-emphasis filter on the $x$ to amplify the high frequencies. The pre-emphasis filter is useful in several ways. For example, the high-frequencies are generally lower in amplitude than low-frequencies and therefore using a pre-emphasis filter helps to avoid numerical problems during STFT and improves the signal-to-noise ratio.

After applying the pre-emphasis filter, the signal is split into short time frames. Since the frequency contour of the signal is lost over time, the Fourier transform is performed assuming that the frequency of the signal is stationary for a very short period, not over the entire signal. The typical frame size for speech processing is from 20ms to 40ms, with a 50% overlap. For example, a common choice is 25ms for frame size and 10ms (15ms overlap) for stride overlap size.

The next step is to cut the signal into frames and apply Hamming or the Hanning window function to each frame. The spectrum can be calculated by performing an N-point FFT (NFFT) on each frame. Here, N is generally set as 256 ($16ms$) or 512 ($32ms$). Finally, the spectrum that is obtained through STFT can be expressed with magnitude and phase by the following equation:

$$D = S * P \quad (2)$$

where $D$ is complex-valued spectrum, $S$ is magnitude and $P$ is phase, respectively.

In summary, raw spectrum can be recovered from speech waveform directly as shown in Fig. 2. Thus, we use spectrum to perform voice transformation in an effective way with out any post-processing.

### B. Proposed model structure

*1) Model flow:* The vocoders mentioned in Section 2 are complex and computationally expensive, they require a lot of repetitive computation to restore the audio waveforms. To solve this problem, we focus on the conversion at the spectrum level. Fig. 3 shows, the conventional method of using MFB in the upper part of the figure, and the proposed transformer network in the lower part of the figure.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

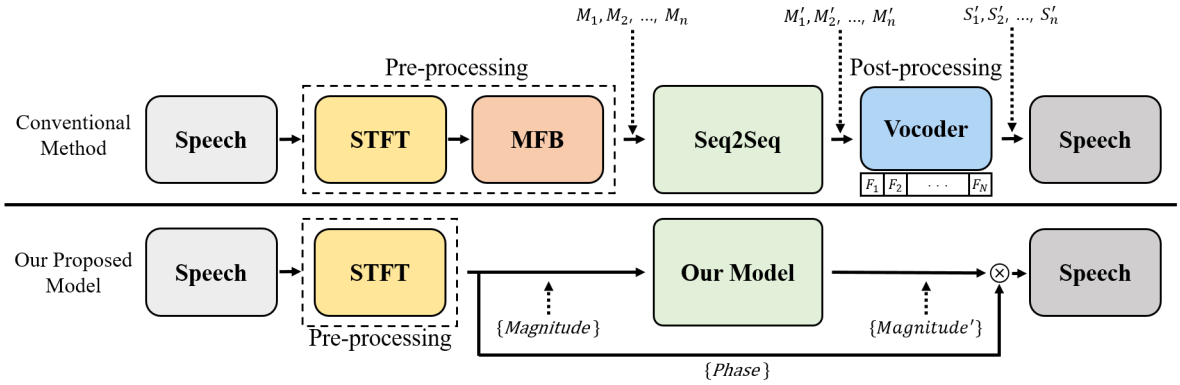


Fig. 3. Difference between the conventional method and our proposed model on voice conversion. The conventional method shown in the upper part requires pre- and post-processing with MFB, while our proposed model only requires the raw spectrum to create the waveform.

One of the conventional methods, Tactoron [21] uses the output of the MFB $M_1, M_2, ..., M_n$ as the input to Seq2Seq-based model and obtains the output through the vocoder. The encoder input in the Seq2Seq considers all the temporal information and in that way is no different from our model. However the decoder predicts $n$ frames of MFB at once, thereby reducing the number of decoder steps to $n/\gamma$, where $\gamma$ is the reduction factor. Post-processing of linear scale spectrum $F$ is performed using CBHG (1D convolution bank, highway network, bidirectional gated recurrent unit) module which results in $F_1, F_2, ..., F_n$. The vocoder is essential to convert $F$ into a waveform expressed as $S'_1, S'_2, ..., S'_n$. The method uses the conventional autoregressive vocoder which predicts current step based on the previous input. Once $S'_1$ is obtained, $S'_1$ is used to predict $S'_2$ and finally $S'_n$. However, this iterative process leads to a high computational cost.

On the other hand, in the proposed model shown in Fig. 3, the magnitude $S$ and phase $P$ are obtained using Eq. (2) from the raw spectrum after passing through STFT. The $S$ is then set as the input to the model encoder and converted in a parallel manner using the decoder. After element-wise multiplication between final output of the model $\hat{x}$ and input phase $P$, it is possible to get a converted target speech by inverse STFT. We can recover the predicted voice instantly using the converted magnitude and phase of the source without help of the vocoder. Our proposed model is a fast and effective approach to convert realistic voices using raw spectrum in a parallel manner and does not dependent on post-processing.

*2) Tokens and zero-padding:* Entering the model input using corpus is done via word embedding. The spectrum, unlike the corpus, consists of continuous values. The spectrum contains $N$ dimensions by time $T$. These values are not sparse representations. The corpus sets the maximum length and proceeds with a start of sentence (SOS) -token in the front and an end of sentence (EOS) -token at the end.

The SOS-token combined sequence is used as the decoder input, because Seq2Seq-based model needs to be trained with real values by teacher forcing. However, in the inference phase, the input of decoder uses only SOS-token. Through this, the autoregressive transformer performs prediction using beam search or greedy search. We then put the EOS-token into our decoder input and perform voice conversion. In addition, since beam search is based on beam depth and the softmax function, we use greedy search.

We apply zero-padding for the whole spectrum. The reason for using zero-padding is that the transformer network considers the whole sequence and learns in parallel. Even if the voice scripts are the same, the length of each speaker's characteristics is different.

In order to avoid attention between a zero value and the real vector, we multiply the vector with $-1e-9$ when there is a zero value on the dimension in each time step. The zero-padding is described in the next section.

*3) Transformer-based model architecture:* Fig. 4 shows our transformer-based model architecture. Firstly, we obtain a spectrum that depends on the $NFFT$ coefficients and then separate $S$ and $P$ by Eq.(2). After that, $S$ is used as the encoder input. In this case, we do not use word embedding [22] because the $S$ is a time-frequency domain that consists of sampling the frequency along the time axis. The final input is $S$ plus the position vector passed through Positional Encoding (PE). Then, multi-head attention is performed through the $N$-encoders. The multi-head attention results pass through two-layer feed-forward network that contains rectified linear units (ReLu) [23]. The process up to now is to make new context information by combining the entire temporal information for each time step. We then use a residual connection [24] that adds input data to the values obtained until now. This means that the context information that is not included in the input temporal information is processed by the input and added. The encoder looks at the given temporal information and encodes each time step into a better representation.

The decoder only uses the magnitude from the target $y$, which has passed through STFT in training phase. However, the decoder is different from the encoder since it uses masked multi-head attention when performing self-attention. The reason for using masked multi-head attention is to prevent self-attention. This is done by covering features after the current

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

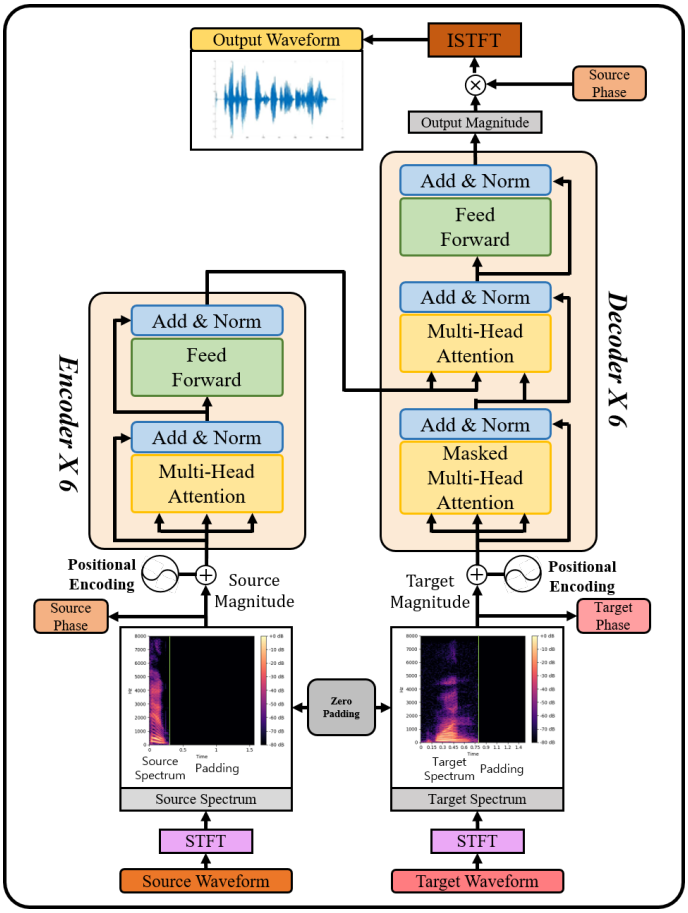


Fig. 4. Our transformer-based model architecture. The input of the encoder is the magnitude of the raw spectrum and the output of decoder is the converted magnitude. Predicted $\hat{x}$ is multiplied element-wise with the phase of the source spectrum. In our method, word embedding, output linear, and softmax function are not needed.

time step during self-attention. This shows that the transformer network is an autoregressive model. After that, attention is concatenated between the encoder outputs and decoder outputs. This process determines how much the decoder uses temporal information from the input spectrum of $x$ to express $y_i$. The results of encoder-decoder attention are added to the masked multi-head attention results of the decoder and passed to a feed-forward network. So far, the outputs $\hat{x}$ have the same dimension $d_{model}$ as inputs $x$ and targets $y$, only the temporal lengths of the magnitude are different. The predicted $\hat{x}$ only has the magnitude converted from source $x$ to target, which is then multiplied by $P$ to make a spectrum containing complex numbers. Finally, it can be restored to waveform speech using the inverse STFT.

The transformer has fewer parameters than other models, and because it uses feed forward network, parallelism is easily achieved and fast operation is possible. In addition, modeling can be more accurately because the information between long temporal relationships is directly linked.

## IV. Experimental Setup

In this section, we introduce the dataset, pre-processing, and hyperparameters.

### A. Database and feature extraction

We use the TIDIGITS [25] dataset which consists of 326 speakers (111 men, 114 women, 50 boys, 51 girls) who pronounce numbers. Among them we experiment with independent numeric units (e.g., "one", "two", ..., "oh", "zero"). Our experiments require a pair of source and target data from each corresponding speaker. Therefore, we train on a paired dataset of 55 men, 57 women, 25 boys and 26 girls. The testing- and training data was split according to the division used in TIDIGITS. The sampling rate of the corpus is $20kHZ$ and dataset was collected with an Electro-Voice RE-16 Dynamic Cardiod microphone in a quiet space.

We downsampled $20kHZ$ to $16kHZ$ in order to reduce the computation. We preprocessed the dataset with $NFFT$ as 512 ($32ms$) and $hop\_length$ as 256 ($16ms$) to get the raw spectrum. The dimension of the obtained spectrum is $(257, T)$. However, since the transformer $d_{model}$ is $2^n$, we intentionally remove the last imaginary part of the spectrum.

### B. Data pre-processing

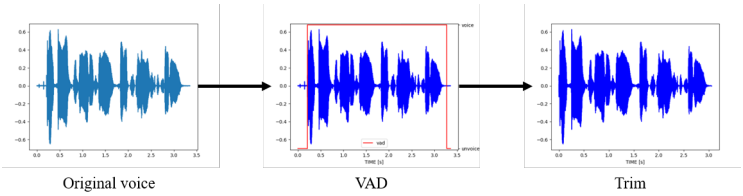


Fig. 5. Original waveform (left), VAD (middle), trimmed waveform (right)

*1) Voice Activity Detection:* Voice Activity Detection (VAD) is a technology applied to voice processing that detects the presence or absence of human voice. As shown in Fig. 5, VAD is an algorithm mainly used in speech recognition that determines the threshold criteria for distinguishing background noise from real speech. We use VAD[2] to reduce the maximum sequence length of the dataset by removing the silent sections at the front and the back of the data based on a threshold to speed up computation. Through the pre-processing, this technique not only makes our model accelerate learning, but also prevents the complexity from growing too fast as the temporal information gets longer.

*2) SOS-token, EOS-token, Padding:* In NLP, the first token of a sentence is SOS-token, and the last token is EOS-token. Usually, EOS-token is used to let the model know when the input sentence is over. In addition, SOS-token is utilized in the inference phase as the decoder input. Thus, we created SOS- and EOS-token corresponding to the $(256, 1)$ dimension which are uniformly distributed at random, with values between 0 and 1. We concatenated the SOS-token in front of the decoder

[2]https://github.com/F-Tag/python-vad

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

inputs in the whole training dataset. In the test phase, we put only the SOS-token into the decoder and our model infers the prediction using greedy search.

The last step of the pre-processing is inserting the padding. First, we find the maximum sequence lengths in the training dataset. Then, to match the magnitude temporal information, we zero-pad with the whole training dataset to the maximum sequence length. During training, $-1e-9$ values are used to prevent multi-head attention from occurring in the zero-padded locations. Moreover, to match the same sequence length for inputs of the model, we add zero-padding after concatenating the EOS-token with target dataset at the end.

### C. Hyperparameter

We used the Adam optimizer [26] with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e-9$ set as the parameters. Since the number of training dataset is small, we could not use the original learning rate in [5]. On the other hand, our initial learning rate is $1e-4$ and we have the number of $decay\_step$ as 4000 and $decay\_rate$ as 0.96.

We implemented our model with Tensorflow 2.0 and trained with one Titan RTX GPU. However, since we have small paired dataset and no post-processing, it is enough to use one 1080TI GPU in our experiments. During the inference, the GPU only uses 500 – 550 MiB of memory.

TABLE I
MODEL HYPERPARAMETERS

| Hyperparameters | Value |
|---|---|
| $N_{encoder}$ | 6 |
| $N_{decoder}$ | 6 |
| $N_{heads}$ | 8 |
| $d_{model}$ | 256 |
| $d_{ff}$ | 1024 |
| $D_{rate}$ | 0.1 |

Table I shows the hyperparameters. Six encoders and decoders, as well as eight multi-head attentions were used in our model. The model size $d_{model}$ is 256 and the dimension size used for the feed forward network $d_{ff}$ is 1024. As the dropout [27] rate we selected 0.1 and used it for training only. We adopted two losses.

$$L_1 = \sum_{i=1}^{n} |y_{true} - y_{predicted}| \tag{3}$$

$$L_{MSE} = \frac{1}{2} \sum_{i=1}^{n} (y_{true} - y_{predicted})^2 \tag{4}$$

$$L_{final} = L_1 * 0.5 + L_{MSE} * 0.5 \tag{5}$$

Eq. (4) has the advantage of minimizing the difference between variance and bias quickly, while Eq. (3) tends to ignore the outliers, which is problem with Eq. (4). Therefore, we combined these equations based on the hypothesis that they could complement each other in this case.

## V. RESULTS

In this section, we perform our voice conversion experiments on TIDIGITS dataset using the metrics naturalness, similarity, and clarity with mean opinion score.

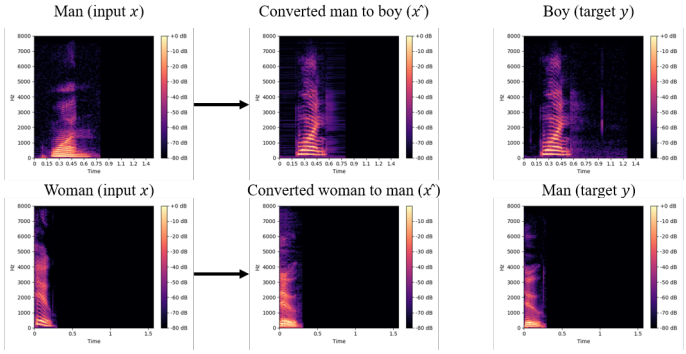


Fig. 6. Visualization of our model's conversion results. The first row of the figure shows the inference results of conversion from a man's voice to a boy's voice saying "1". The second row shows the inference results of conversion from a woman's voice to a man's voice saying "5". In all figures, $8kHz$ is maximum frequency corresponding to the y-axis.

Fig. 6 shows the speech conversion results of our proposed model. The figures in the first row are the results of voice conversion from a man to a boy. The figure on the left is the input spectrum of a man's voice, the center is the converted output, while the source spectrum from a boy's voice is on the right. As shown on the top row in Fig.6, after converting $x$ to $\hat{x}$, the spectrum spreads out to higher frequencies to resemble the target $y$ closely. Likewise, a similar effect can be observed on the second row, where the spectrum is compressed instead.

Fig. 7 shows more accurate analysis of our conversion results. The first row shows the amplitude spectra of the source, prediction, and the target respectively when transforming from a man's voice to a boy's voice. The maximum over the y-axis in $man_x$ is nearly 1.3 and $boy_y$ is around 0.9, while our converted result $\hat{x}$ is similarly close to 0.9. Frequencies of $man_x$ in the lower frequency bins are higher than the frequencies of $boy_y$. Through this analysis, it is clear that low frequencies from $man_x$ are densely distributed and higher in magnitude than $boy_y$.

Likewise, each figure on the second row shows the amplitude spectra of the source, prediction, and the target respectively when transforming from a woman's voice to a man's voice. The maximum value over the y-axis in $woman_x$ peaks at 7.0 and the maximum of $man_y$ is around 2.4, while our conversion result $\hat{x}$ similarly has the maximum to 2.4. Furthermore, the maximum value of $\hat{x}$ occurs around the same frequencies as in the target $man_y$. Before conversion, the highest magnitude in lower frequency bins is 7.0, but gets scaled to the range of $man_y$. Our model correctly attenuates the magnitude of the frequency bins 50 to 100 in order to closer resemble the man's voice. Therefore the results, as shown in Fig. 7 indicate that our proposed model successfully performed the conversion.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

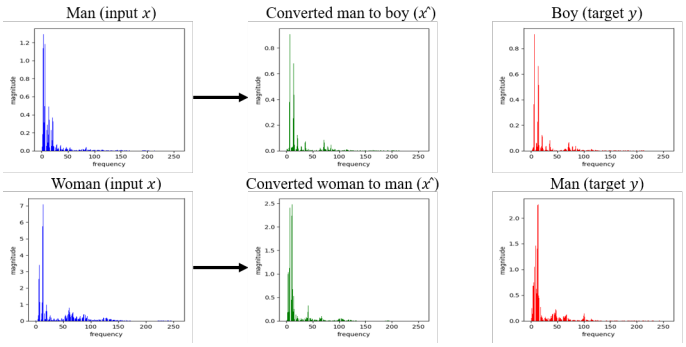


Fig. 7. Visualization of the difference in the spectral content our model's conversion results. The samples shown here are the same as in Fig.6. The x-axis consists of 256 bins accordingly to the NFFT with N=512.

To get quantitative performance, we randomly gathered 38 adults in the age range from 20 to 30 years old. We measured our proposed model's performance using the metrics naturalness, similarity, and clarity with mean opinion score. Samples of voices were randomly selected, and the same batches of samples were given to each participant. For each source and each target we generated four random samples, in total, 144 samples were evaluated.[3] Source speakers and target speakers are different.

TABLE II
MEAN OPINION SCORE EVALUATION FOR NATURALNESS OF CONVERTED SPEECH WITH 95% CONFIDENCE INTERVAL. HIGHER SCORE CORRESPONDS TO A MORE NATURAL VOICE (1-5).

| Source \ Target | Man | Woman | Boy | Girl |
|---|---|---|---|---|
| Man | - | 3.28±0.29 | 4.20±0.53 | 3.72±0.33 |
| Woman | 2.82±0.29 | - | 3.18±0.30 | 3.45±0.29 |
| Boy | 2.97±0.31 | 3.24±0.27 | - | 3.80±0.28 |
| Girl | 3.01±0.29 | 3.56±0.25 | 3.53±0.32 | - |

Table II is an evaluation of how natural the converted voice sounds to a human. The highest score ($4.20 \pm 0.53$) was obtained from the conversion tasks from man to boy, while the lowest score ($2.82 \pm 0.29$) from conversion tasks from woman to man. Table III is an evaluation of how similar the converted voice is to the target voice. We got the highest similarity ($4.26 \pm 0.22$) from conversion tasks from man to boy and the lowest similarity ($3.09 \pm 0.31$) from conversion tasks from woman to man.

TABLE III
SIMILARITY EVALUATION FOR THE CONVERTED SPEECH WITH 95% CONFIDENCE INTERVAL. HIGHER SCORE IS MORE SIMILAR TO TARGET VOICE (1-5).

| Source \ Target | Man | Woman | Boy | Girl |
|---|---|---|---|---|
| Man | - | 3.91±0.24 | 4.36±0.19 | 4.26±0.22 |
| Woman | 3.09±0.31 | - | 3.69±0.31 | 3.93±0.24 |
| Boy | 3.30±0.30 | 3.50±0.27 | - | 4.28±0.18 |
| Girl | 3.39±0.30 | 4.04±0.20 | 4.13±0.24 | - |

[3]Audio samples are available at https://kaen2891.github.io/

TABLE IV
CLARITY EVALUATION FOR THE CONVERTED SPEECH WITH 95% CONFIDENCE INTERVAL. HIGHER SCORE IS CLEARER WITH RESPECT TO THE SCRIPT (1-5).

| Source \ Target | Man | Woman | Boy | Girl |
|---|---|---|---|---|
| Man | - | 3.78±0.27 | 4.31±0.19 | 4.22±0.21 |
| Woman | 3.57±0.30 | - | 3.83±0.26 | 3.80±0.22 |
| Boy | 3.47±0.26 | 3.80±0.26 | - | 4.24±0.22 |
| Girl | 3.84±0.30 | 4.00±0.23 | 4.24±0.22 | - |

Table IV is an evaluation of how clear the pronunciation of the converted voice is given the script. We got the highest clarity ($4.31 \pm 0.19$) from conversion tasks from man to boy and the lowest clarity ($3.47 \pm 0.26$) from conversion tasks from boy to man. The score when converting to a child's voice was generally high. In the overall speaker average mean opinion score, we obtained $3.40 \pm 0.31$ in naturalness, $3.82 \pm 0.25$ in similarity, and $3.93 \pm 0.25$ in clarity. Our results showed that the proposed method can perform the transformation with good clarity while maintaining appropriate naturalness and similarity.

## VI. CONCLUSION

### A. Summary

We proposed a voice transform with self-attention mechanism in a raw spectrum level, while conventional methods use a vocoder in MFB level. MFB-based approaches have the advantage of higher learning speeds compared to using the raw spectrum due to a smaller number of features. However, speech generators with MFB approach require an additional computationally expensive vocoder for the training process. With the vocoder, it is possible to get better quality of the voice in the synthesis. On the contrary, the problems with complexity due to the extra computation are inevitable. The additional pre- and post-processing such as MFB and vocoder are not essential to convert human voices. In this paper, we proposed a vocoder-free end-to-end voice conversion method using transformer network to alleviate the computational burden from additional pre- and post-processing. Our proposed model is a fast and effective approach to convert realistic voices using raw spectra in a parallel manner to generate different style of voices with clear pronunciation. We focused on converting the raw spectrum obtained by the STFT without the help of the vocoder, which would have required iterative synthesis. We gathered 38 participants and conducted MOS evaluation on the naturalness, similarity and clarity of the converted speech. In the overall speaker average mean opinion score, we obtained $3.40 \pm 0.31$ in naturalness, $3.82 \pm 0.25$ in similarity, and $3.93 \pm 0.25$ in clarity. Our results showed that the proposed method could perform the transformation with good clarity while maintaining appropriate naturalness and similarity.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

### B. Future Work

In the evaluation phase, there was an unnatural converted part of $\hat{x}$. It seems to be caused by misalignments since the lengths of $\hat{x}$ and $phase_x$ deviate significantly. This is a feature of the transformer-based model which converts to the maximum length. In other words, the lengths in the whole dataset are the same because of zero-padding. However, if the actual vector length of $phase_x$ is less than the $\hat{x}$, it causes a serious problem that leads to misalignment. In the above case, the quality of the recovered waveform can be poor. Thus, the pitch is broken, and it sounds less natural. Therefore, our model needs to modify the phase information as well to solve the misalignment problem. This discovery is unexpected, and it suggests that there is a problem related to the input spectrum length.

We indentified the importance of $phase$ in the study. The problem can be solved if $phase_x$ and the converted $\hat{x}$ are aligned with each other. To do this, we must use complex neural network [28] to align the magnitude and phase included in the raw spectrum. If the phase is aligned based on the converted magnitude, the quality of voice will be improved. It will be possible to convert voices of minors with poor speech recognition performance to those of common adults. We can achieve better speech recognition performance through speaker adaptation which replaces the features of minor's voice with the features of common adult's voice. We are going to research phase adaptation and alignment with magnitude as our next task.

## ACKNOWLEDGMENT

This work was partially supported by the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (R7124-16-0004, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding).

## REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[3] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[6] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[7] B. Logan *et al.*, "Mel frequency cepstral coefficients for music modeling." in *Ismir*, vol. 270, 2000, pp. 1–11.

[8] D. Griffin and J. 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.

[9] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[10] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[11] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[12] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanvesky, and Y. Jia, "Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation," *arXiv preprint arXiv:1904.04169*, 2019.

[13] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[14] Y. Jia, R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, and Y. Wu, "Direct speech-to-speech translation with a sequence-to-sequence model," *arXiv preprint arXiv:1904.06037*, 2019.

[15] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," *arXiv preprint arXiv:1802.08435*, 2018.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7464–7473.

[18] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6706–6713.

[19] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[20] W.-C. Huang, T. Hayashi, Y.-C. Wu, H. Kameoka, and T. Toda, "Voice transformer network: Sequence-to-sequence voice conversion using transformer with text-to-speech pretraining," *arXiv preprint arXiv:1912.06813*, 2019.

[21] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[22] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in neural information processing systems*, 2014, pp. 2177–2185.

[23] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] R. G. Leonard and G. Doddington, "Tidigits ldc93s10." Linguistic Data Consortium, 1993.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," *arXiv preprint arXiv:1705.09792*, 2017.