Pitch Encoder-Based Zero-Shot Voice Conversion for Improving Speech Quality

Hwa-Young Park
AI Graduate School
Gwangju Institute of Science and Technology
Gwangju 61005, Korea
hwayoung_park@gm.gist.ac.kr

Chae-Woon Bang
AI Graduate School
Gwangju Institute of Science and Technology
Gwangju 61005, Korea
bang4045@gm.gist.ac.kr

Chanjun Chun
Dept. of Computer Engineering
Chosun University
Gwangju 61452, Korea
cjchun@chosun.ac.kr

Hong Kook Kim
[1] School of EECS,
Gwangju Institute of Science and Technology
[2] AunionAI, Co. Ltd.
Gwangju 61005, Korea
hongkook@gist.ac.kr

***Abstract*— This paper aims to improve speaker similarity of converted voice in a zero-shot voice conversion (VC) framework. To this end, an approach of utilizing pitch information into Perturbation AUTOVC, which is a zero-shot VC model, is proposed. Perturbation AUTOVC applies information perturbation associated with voice characteristics to input speech, delivering higher voice quality than AUTOVC. However, it has limited capability in representing speaker's characteristics because it disentangles only linguistic and timbre content from voice. To address this issue, the proposed approach first extracts pitch contour that is concatenated with the outputs of content and speaker encoders. Second, instead of directly concatenating the pitch contour, a pitch encoder is designed to extract pitch embeddings from the pitch contour, which are then concatenated. Lastly, the extracted pitch embedding is concatenated with the mel-spectrogram generated by passing through the decoder's fully connected layer. Performance evaluation shows that perturbation AUTOVC concatenating pitch contour yields lower character error rate and higher mean opinion score than that without using pitch information. Moreover, it is revealed that concatenating pitch information via the pitch encoder contributes to improving speaker similarity of converted voice.**

Keywords—voice conversion, zero-shot, Perturbation AUTOVC, information perturbation, pitch encoder

I. Introduction

Voice conversion refers to a technology that converts one voice to sound like another voice without changing the context [1]. Advances in deep learning have significantly enhanced the field of voice conversion, making it possible to generate more natural voices. Zero-shot voice conversion focuses on converting the speaker of the source speaker to mimic that of another who is unseen in the training dataset. This technique is widely used for various purposes, such as personal information protection and film dubbing [2], [3].

Many existing voice conversion methods set the input and output speakers to be the same during training. This is because training is approached by disentangling voice representations and restoring them using the disentangled pieces of information.

This work was supported in part by the GIST-MIT Research Collaboration grant and by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2024 (Development of artificial intelligence-based sign language interpretation and translation service technology for communication between a deaf person and the public, RS-2023-00303791, Contribution Rate: 50%).

However, this method has limitations due to training-inference mismatching. This can result in improper disentangling of the speech during actual conversion, preventing a seamless conversion process. The AUTOVC framework can achieve zero-shot voice conversion with non-parallel data and features a simple learning method that is based solely on autoencoder losses [4]. However, when controlling the bottleneck dimension to separate the speech into linguistic and speaker information, the performance of AUTOVC becomes highly sensitive to the bottleneck dimension. For example, when the bottleneck dimension is too narrow, linguistic information can be lost, causing difficulties in the restoration process. In contrast, a dimension that is too wide can result in improper separation of linguistic and speaker information, causing disruptions in training. For example, when the bottleneck dimension is wide, the speaker information remaining in the content embedding is useful for restoration when the speaker information (input and output speech) is the same. This problem is also present in other voice conversion methods that adopt similar approaches to AUTOVC. To address this issue, Perturbation AUTOVC can effectively address this issue through the simple method using information perturbation proposed by neural analysis and synthesis (NANSY) [5].

NANSY is a framework for speech analysis and synthesis that replaces the bottleneck approach with information perturbation, enhancing the model's controllability while achieving high-quality speech synthesis. One of its key features is the adoption of a self-supervised learning method, enabling effective training even in the absence of labeled data. Additionally, NANSY introduces a new pitch feature, Yingram, as part of its analysis, which sets it apart from conventional approaches.

In the training stage, Perturbation AUTOVC uses information perturbation to set the source and target speakers' information differently [6]. This reduces training–inference mismatch and subsequently enhances zero-shot conversion performance. Accordingly, Perturbation AUTOVC demonstrates higher performance than other voice conversion methods, such as activation guidance and adaptive instance normalization voice conversion (AGAIN-VC) and vector quantization and mutual information-based voice conversion (VQMIVC), even for non-training datasets [7], [8]. Both models focus on addressing the issue of speaker information leaking into content embeddings in voice conversion based on

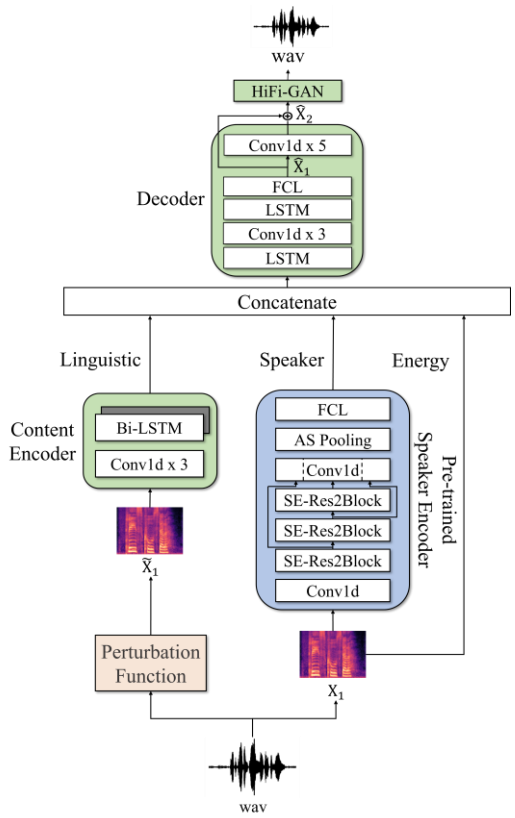

Fig. 1. Network architecture of the perturbation AUTOVC model [6] as the baseline model in this work.

disentanglement learning techniques, a goal they share with Perturbation AUTOVC. Nevertheless, Perturbation AUTOVC, which confines speaker's characteristics to linguistic and timbre, leaves opportunities for improvement in terms of handling additional characteristic such as pitch. Moreover, unlike the timbre content provided through the speaker encoder, the pitch information, which is not separately provided, can be included in the content embedding, thereby reducing the model's controllability.

In an effort to improve the quality and controllability of Perturbation AUTOVC, we added a pitch encoder to process pitch information. First, we use the pitch contour extracted through the yet another algorithm for pitch tracking (YAAPT) algorithm, concatenated channel-wise with the outputs of the content encoder and speaker encoder, as the input to the decoder [9]. Second, we model the pitch contour as a pitch embedding through the pitch encoder based on the vector quantized-variational autoencoder (VQ-VAE) model, then concatenate it with the output embeddings of other encoders in the same manner as the first method to use as input to the decoder. Additionally, the third method involves adding a VQ-VAE pith encoder, which is then concatenated with the output of the decoder's fully connected layer (FCL) and passes through an additional convolutional network. It is of particular importance that the training process for the pitch encoder was straightforward because no pre-training or additional losses were required since it was trained only on a reconstruction loss together with the VC model.

The major contributions of this study include the following:

- We used the pitch contour as input to enable the VC model to handle pitch information as well.
- We added a pitch encoder to Perturbation AUTOVC for modeling pitch information, enhancing speaker similarity and controllability.
- We trained the pitch encoder together with the VC model with no pre-training or additional losses, reducing the learning complexity of the entire model.

The remainder of this paper is organized as follows: Section II briefly outlines several research works closely related to this study. Then, Section III proposes an approach of incorporating pitch information into perturbation AUTOVC, and Section IV presents the experimental results. Finally, Section V concludes this paper.

II. Related Work

A. Perturbation AUTOVC

Perturbation AUTOVC is a method that combines the features of AUTOVC and NANSY to achieve high-performing zero-shot conversion [6]. Although AUTOVC disentangles linguistic and speaker information by controlling the bottleneck dimension, it is unable to identify the optimal bottleneck dimensions, which can vary depending on the datasets. In addition, it is trained using the same source and target speakers, potentially resulting in training–inference mismatch and causing linguistic and speaker information to be mixed in the content embedding.

Perturbation AUTOVC resolves these challenges using information perturbation proposed by NANSY. Information perturbation distorts speaker-dependent information (such as pitch and format inherent in voice), making the source and target speaker information different. This results in the model excluding unwanted speaker information and obtaining only the necessary linguistic information. This process increases the model's controllability by separating voice representations. Despite its simple method, Perturbation AUTOVC has demonstrated its ability to deliver high-performing zero-shot conversion for various datasets while maintaining the advantages of AUTOVC. Moreover, the output dimension can be set arbitrarily because it is not affected by the bottleneck dimension, meaning no consideration of the bottleneck dimension is required.

Fig. 1 illustrates the network architecture of the Perturbation AUTOVC. The Perturbation AUTOVC represents linguistic information and speaker-specific information using a content encoder and a speaker encoder, respectively. In other words, the mel-spectrogram of the input speech waveform is processed by a perturbation function to alter its pitch, formant, and frequency envelope and it is input to the content encoder to extract speaker-independent embeddings. In contrast, the speaker encoder extracts a speaker-specific embedding from the original, unperturbed mel-spectrogram. In parallel to the pitch encoder, the energy, which is calculated by averaging the original mel-spectrogram along the frequency axis, is extracted and then it is combined with both linguistic and speaker embeddings. This

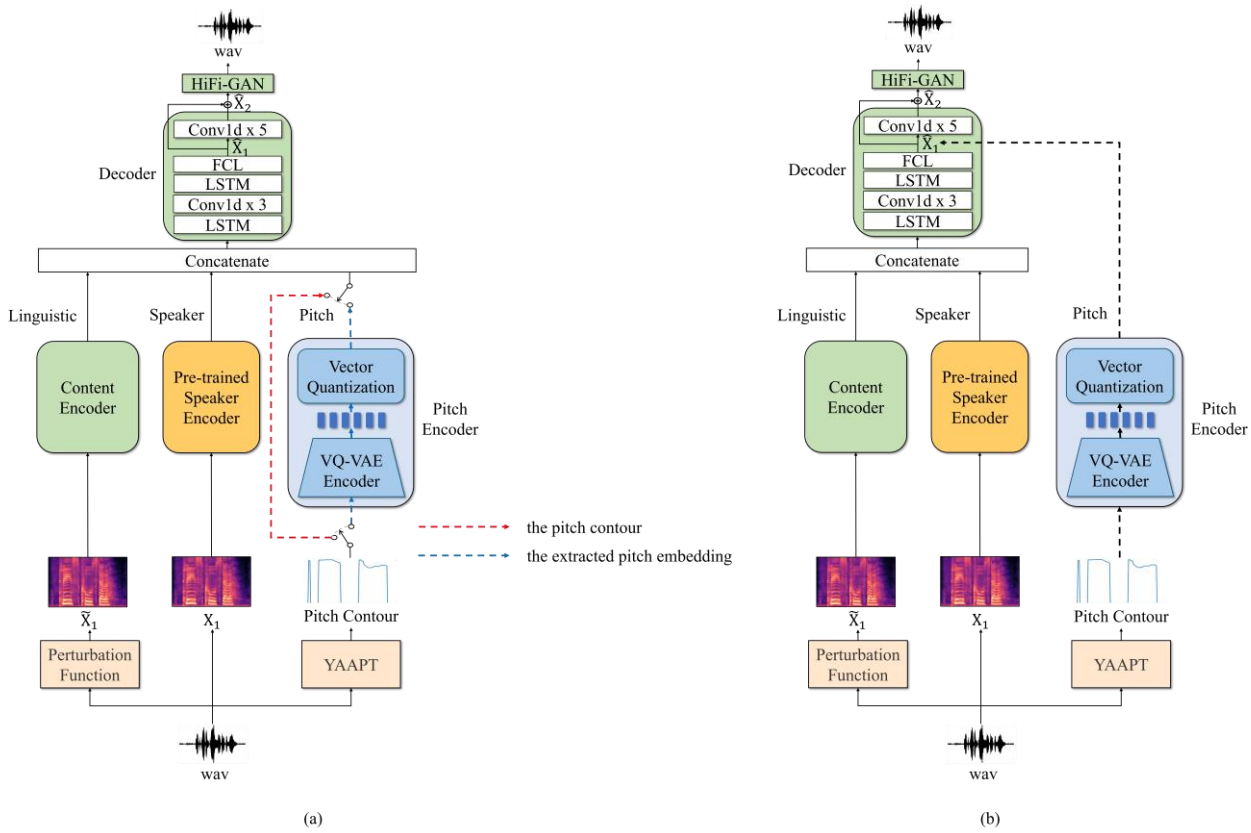

Fig. 2. Network architectures of the proposed methods: (a) concatenating pitch contour (denoted as a red dotted line) or embeddings from the pitch encoder (denoted as a blue dotted line) with linguistic and speaker embeddings (b) concatenating the pitch embedding with the output of the FCL in the decoder (denoted as a dotted line).

concatenated embedding is then used as input for the decoder of the Perturbation AUTOVC to reconstruct the original, undistorted speech waveform.

B. ControlVC

The ControlVC performs non-parallel zero-shot voice conversion with the aim of achieving time-varying control over pitch and speed [10]. ControlVC consists of three phases: pre-processing, analysis, and synthesis. In the pre-processing phase, the speed of the source speech is controlled using time-domain pitch synchronous overlap and add (TD-PSOLA), enabling either time-stretching or pitch shifting [11]. During the analysis phase, the pitch contour of the speed-controlled source speech is estimated and subsequently modified according to the pitch control curve. Then, pitch embedding is generated using a VQ-VAE pitch encoder. The VQ-VAE model is trained by minimizing the mean squared error (MSE) between the estimated pitch sequence (from the original speech where no speed and pitch controls were applied) and the original pitch sequence. Furthermore, using pre-trained hidden-unit bidirectional encoder representation from transformer (HuBERT), a speed-controlled language embedding is extracted, whereas the target speaker embedding is extracted through the pre-trained speaker encoder [12]. Finally, a waveform of converted speech is generated using the modified HiFi-GAN vocoder. Here, generative adversarial network for efficient and high-fidelity speech synthesis (HiFi-GAN) is a model trained from scratch to perform voice conversion while fixing the other models (HuBERT, speaker encoder, and pitch encoder).

III. Proposed Methods

Fig. 2 displays the entire structure of the proposed method. The content encoder takes the mel-spectrogram of the perturbed input speech, $\hat{X}$, and models an embedding that contains only linguistic information. The speaker embedding is extracted from the unperturbed speech using speaker recognition model. This process is identical to the baseline model, Perturbation AUTOVC, shown in Fig. 1. The proposed method adds a process for incorporating pitch.

The pitch contour extract from the unperturbed speech using the YAAPT algorithm is processed in two ways. As shown in Fig. 2(a), the first method is to concatenate pitch contour with the outputs of content and speaker encoders. The second method is to model the pitch contour as an embedding through the pitch encoder before concatenating it. Additionally, in the third method, the pitch information is modeled by VQ-VAE pitch encoder, which computes discrete pitch embeddings, and then concatenating it after the decoder and before the additional convolutional network. This can be seen in Fig. 2(b). Finally, the mel-spectrogram of the converted voice generated by the decoder is restored as speech waveforms using the HiFi-GAN vocoder [13]

A. Content Encoder

To extract linguistic information, we applied information perturbation to the audio input to distort speaker-dependent information, similar to the method used in Perturbation AUTOVC. In this study, we applied three perturbation functions that distort the pitch, formant, and frequency shaping using a Praat library. Detailed information related to information perturbation can be found in [14]. This perturbed input voice is fed into the content encoder.

Given the different speaker information between the input speech of the content encoder and the output speech generated through the decoder, only speaker-independent information is processed by minimizing the L1 loss between the two speeches, which is defined as

$$L_c = \mathbb{E}\left[\left\|C - \hat{C}\right\|_1^1\right] \tag{1}$$

where C and $\hat{C}$ represent the results obtained when the source and reconstructed speech, respectively, are input into the content encoder. Then, the model is trained to minimize the difference between these two results, preventing the loss of linguistic information from the original speech. The content encoder is constructed with three convolution layers followed by a bidirectional long short-term memory (Bi-LSTM) layer, which is identical to AUTOVC.

B. Pitch Encoder

To extract the pitch sequences of the source speech, we used the YAAPT algorithm. The extracted pitch sequences can be used as is or modeled through a pitch encoder. When using the pitch contour as is, it is concatenated channel-wise with content and speaker embedding. The extracted pitch sequences are fed into the VQ-VAE-based pitch encoder, generating 128-dimensional latent vector sequences to model the pitch as a pitch embedding. These vectors are then mapped to the closest codes through the bottleneck codebook. This process is similar to the model structure and method used in ControlVC. However, while ControlVC pre-trains VQ-VAE by minimizing the MSE loss between the unmodified speech pitch and the original pitch, the encoder and bottleneck part of VQ-VAE are trained together with the VC model in this study without using any additional loss.

The extracted pitch embedding is concatenated in two ways. First, the pitch embedding is concatenated with the content and speaker embeddings before being used as input to the decoder. Second, the pitch embedding is concatenated with the decoder's output $\hat{X}_1$ and then used as input to the convolution layer following the FCL. Comparison of the results for each method is shown in Section IV.

C. Speaker Encoder

We used the pre-trained speaker recognition model ECAPA-TDNN to extract the speaker information [15]. Using a time delay neural network, the ECAPA-TDNN model highlights the interactions of input characteristics through emphasized channel attention, propagation, and aggregation (ECAPA) and extracts consistent features while maintaining temporal information. We employed the VoxCeleb2 dataset to train the ECAPA-TDNN model [16]. Considering that a speaker recognition model trained with a large-scale dataset can capture information about unseen speakers, many voice conversion models use speaker recognition or speaker verification models pre-trained with such a large-scale dataset to extract the speaker information. The generated speaker embedding with 192 dimensions was used as vocal speaker information in this study.

D. Decoder

The decoder uses the information extracted from each network to generate the mel-spectrogram of the converted voice. A decoder network structure is identical to that of AUTOVC. Data concatenating the content, speaker (timbre), and pitch information pass through long short-term memory (LSTM) and convolution layers. The data generated after passing through the last fully connected layer is the mel-spectrogram, $\hat{X}_1$, which is the same dimension as the input dimension. $\hat{X}_1$ is then fed into an additional convolution layer to generate the mel-spectrogram, $\hat{X}_2$. Finally, $\hat{X}_1$ is added to $\hat{X}_2$. The decoder minimizes the MSE losses between X (the unperturbed original voice) and $\hat{X}_2$ (the final decoder output), defined as

$$L = \mathbb{E}\left[\left\|X - \hat{X}_1\right\|_2^2 + \left\|X - \hat{X}_2\right\|_2^2\right]. \tag{2}$$

The final output of the decoder, $\hat{X}_2$, is an 80-dimensional mel-spectrogram, which is then converted into a waveform using a pre-trained HiFi-GAN vocoder.

IV. Performance Evaluation

In this study, a series of experiments were conducted using a VCTK dataset containing the voices of 109 speakers [17]. The speakers were divided into seen and unseen speakers with a ratio of 10:1, and five speeches of each seen speaker were used as evaluation data. For training, the voice waveform was converted into an 80-dimensional mel-spectrogram, and the hyperparameters of fast Fourier transform (FFT) size, window size, and hop length used for conversion were set to 1024, 1024, and 256, respectively. During training, the frame length of all mel-spectrograms was padded to 128 or cut randomly. In this study, we measured and compared three objective indicators: character error rate (CER), cosine similarity, and neural mean opinion score (MOS).

A. Performance Metrics

CER is widely used for voice recognition by measuring the character errors between converted voice and original voice. The lower the measured CER, the better the linguistic information is preserved and not damaged during the voice conversion process. In this paper, we used the wav2vec2.0 model pre-trained for 960 hours to recognize speech and then measured CER using the Levenshtein distance [18], [19].

Cosine similarity is used to measure how similar the converted voice is to the target speaker's voice [20]. To measure cosine similarity, we used the speaker encoder from this paper to extract the speaker embeddings of the converted voice and the target speaker. The higher the measured cosine similarity, the better matches the target speaker and the converted voice match.

Finally, Neural-MOS uses a neural network to measure the naturalness of the generated voice. In this paper, we used a

TABLE I. PERFORMANCE COMPARISON OF VOICE CONVERSION BETWEEN THE BASELINE AND PROPOSED METHODS, APPLIED TO TEN SEEN SPEAKERS.

Method	CER (%) (↓)	Cosine Similarity (↑)	Neural-MOS (↑)
Baseline	12.5	0.664	3.167
Baseline+Pitch Contour	**8.2**	0.657	3.017
Baseline+Pitch Encoder	14.9	0.686	2.882
Baseline+Pitch Attention	12.1	**0.733**	**3.297**

Neural-MOS predictor instead of the time-consuming and costly MOS evaluation. The neural network used for evaluation is the SSL-MOS, which was used as the baseline in the Voice MOS challenge [21], [22].

B. Comparison of Voice Conversion

Table I compares the performance of four different models. The models are as follows:

1) Baseline: The original Perturbation AUTOVC,
2) Baseline+Pitch Contour: Perturbation AUTOVC with pitch contour concatenated,
3) Baseline+Pitch Encoder: Perturbation AUTOVC with pitch information modeled by adding a VQ-VAE pitch encoder and concatenating it with content and speaker embeddings as input to the decoder, and
4) Baseline+Pitch Attention**:** Perturbation AUTOVC with pitch information modeled by adding a VQ-VAE pitch encoder and concatenating it after passing through the decoder and before entering the additional convolutional network.

As shown in Table I, the experimental results demonstrates that the second model (Baseline+Pitch Contour) recorded the lowest CER of 8.2% among the three models. This is likely because pitch information includes features such as intonation and stress, which help in identifying word boundaries and distinguishing phonemes during speech recognition. On the other hand, in terms of cosine similarity, the last model (Baseline+Pitch Attention) showed the highest score of 0.733. The third model (Baseline+Pitch Encoder) also achieved higher cosine similarity than the baseline model, suggesting that the pitch encoder contributes to higher quality of converted voice. Specifically, the likely reason for the higher speaker similarity is that pitch information models speaker characteristics more efficiently as it passes through the network. As a result, the second model (Baseline+Pitch Contour) exhibits the lowest CER, but its cosine similarity is lower compared to the other models.

In terms of Neural-MOS, the last model (Baseline+Pitch Attention) scored the highest at 3.297, while the third model (Baseline+Pitch Encoder) scored the lowest at 2.882. The reason the MOS of the third model (Baseline+Pitch Encoder) is lower is that the content and pitch embeddings used as inputs to the decoder change together during the optimization process. In the baseline model, the decoder learns with content and fixed speaker embedding, making the training relatively stable. However, in the third model (Baseline+Pitch Encoder), both content and pitch embeddings are optimized simultaneously, making it harder for the decoder to learn, resulting in a lower MOS. In contrast, the last model (Baseline+Pitch Attention), which concatenates the pitch information after the decoder and before the convolution layer, shows a higher MOS as it facilitates easier training for the decoder.

TABLE II. PERFORMANCE COMPARISON OF ZERO-SHOT VOICE CONVERSION BETWEEN THE BASELINE AND PROPOSED METHODS, APPLIED TO TEN UNSEEN SPEAKERS.

Method	CER (%) (↓)	Cosine Similarity (↑)	Neural-MOS (↑)
Baseline	14.3	0.392	2.506
Baseline+Pitch Contour	**9.1**	0.397	2.734
Baseline+Pitch Encoder	11.6	0.445	2.592
Baseline+Pitch Attention	10.2	**0.500**	**2.924**

C. Comparison of Zero-shot Voice Conversion

To perform zero-shot voice conversion, we used the LibriTTS dataset, which was not included in the training data [23]. Table II shows the results of voice conversion using the LibriTTS dataset. Similar to the result in Table I, the second model (Baseline+Pitch Contour) recorded the lowest CER at 9.04 in the zero-shot conversion. The third model (Baseline+Pitch Encoder) achieved a cosine similarity of 0.445, while the last model (Baseline+Pitch Attention) achieved the highest cosine similarity at 0.5. In terms of Neural-MOS, the last model (Baseline+Pitch Attention) scored the highest at 2.924, while the baseline model scored the lowest at 2.506. In all objective evaluation metrics, the performance gap between the baseline and other proposed methods was much larger than in zero-shot voice conversion compared to the seen dataset. This indicates that the proposed method effectively models pitch information and suggests, as shown in the results of Table 1, that pitch information serves as a significant role in speaker similarity.

V. CONCLUSION

This paper demonstrated that incorporating pitch information into the Perturbation AUTOVC model, a non-parallel zero-shot voice conversion technique, provides significant performance benefits. Experiments were conducted using either pitch contours or pitch embeddings extracted through a pitch encoder, resulting in superior performance in CER, cosine similarity, and Neural-MOS across both seen and unseen speakers. Specifically, when using pitch contours directly, the model exhibited high performance in CER, although there was little difference in cosine similarity and Neural-MOS. In contrast, introducing a pitch encoder to model the pitch contour as pitch embedding led to improved performance in both cosine similarity and Neural-MOS. In future work, we plan to enhance the neural network architecture to further improve the model's performance. For instance, we will explore upgrading the content encoder to a Transformer or Conformer-based model structure to capture longer contextual information and improve the decoder's architecture to reconstruct higher quality mel-spectrograms. Furthermore, we plan to devise an effective pitch modeling method to resolve the trade-off observed in the experimental results between CER and speaker similarity.

REFERENCES

[1] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," Speech Commun., vol. 88, pp. 65–82, Apr. 2017.

[2] J. Yao, Q. Wang, Y. Lei, P. Guo, L. Xie, N. Wang, and J. Liu, "Distinguishable speaker anonymization based on formant and fundamental frequency scaling," arXiv preprint arXiv:2211.03038, 2022. [Online]. Available: https://arxiv.org/abs/2211.03038

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films, and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "AutoVC: Zero-shot voice style transfer with only autoencoder loss," in Proc. Int. Conf. Mach. Learn., 2019, pp. 5210–5219.

[5] H. S. Choi, J. Lee, W. Kim, J. Lee, H. Heo, and K. Lee, "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2021, pp. 16251–16265.

[6] H.-Y. Park, Y. H. Lee, and C. Chun, "Perturbation AUTOVC: Voice conversion from perturbation and autoencoder loss," IEEE Access, vol. 11, pp. 140174–140185, 2023.

[7] Y.-H. Chen, D.-Y. Wu, T.-H. Wu, and H.-Y. Lee, "Again-VC: A oneshot voice conversion using activation guidance and adaptive instance normalization," in Proc. ICASSP, 2021, pp. 5954–5958.

[8] D. Wang, L. Deng, Y. T. Yeung, X. Chen, X. Liu, and H. Meng, "VQMIVC: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion," in Proc. Interspeech, 2021, pp. 1344–1348.

[9] K. Kasi and S. A. Zahorian, "Yet another algorithm for pitch tracking," in Proc. ICASSP, 2002, pp. 361–364.

[10] M. Chen and Z. Duan, "ControlVC: Zero-shot voice conversion with time-varying controls on pitch and speed," in Proc. Interspeech, 2023, pp. 2098–2102.

[11] F. Charpentier and M. Stella, "Diphone synthesis using an overlap-add technique for speech waveforms concatenation," in Proc. ICASSP, 1986, pp. 2015–2018.

[12] W. N. Hsu, B. Bolte, Y. H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," IEEE/ACM Trans. on Audio, Speech, and Language Processing, vol. 29, pp. 3451–3460, 2021.

[13] J. Kong, J. Kim, and J. Bae. "Hifi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," Adv. Neural Inf. Process. Syst. (NIPS), vol. 33, pp. 17022–17033, 2020.

[14] P. Boersma and V. Van Heuven, "Speak and unspeak with PRAAT," Glot Int., vol. 5, nos. 9–10, pp. 341–347, 2001. [Online]. Available: https://www.fon.hum.uva.nl/praat/

[15] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in Proc. Interspeech, 2020, pp. 3830–3834.

[16] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in Proc. Interspeech, 2018, pp. 1086–1090.

[17] C. Veaux, J. Yamagishi, and K. MacDonald, "Superseded-CSTR VCTKcorpus: English multi-speaker corpus for CSTR voice cloning toolkit," Centre Speech Technol. Res. (CSTR), Univ. of Edinburgh, Edinburgh, U.K., Tech. Rep., 2016. [Online]. Available: https://codeocean.com/capsule/4989235/tree

[18] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2020, pp. 12449–12460.

[19] S. Zhang, Y. Hu, and G. Bian, "Research on string similarity algorithm based on Levenshtein distance," in Proc. IEEE 2nd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC), 2017, pp. 2247–2251.

[20] E. Cooper, W.-C. Huang, T. Toda, and J. Yamagishi, "Generalization ability of MOS prediction networks," in Proc. ICASSP, 2022, pp. 8442–8446.

[21] W. H. Gomaa and A. A. Fahmy, "A survey of text similarity approaches," International Journal of Computer Applications, vol. 68, no.13, pp. 13–18, 2013.

[22] W.-C. Huang, E. Cooper, Y. Tsao, H.M. Wang, T. Toda, and J. Yamagishi, "The VoiceMOS challenge 2022," arXiv:2203.11389, 2022. [Online]. Available: https://arxiv.org/abs/1312.6114

[23] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A corpus derived from LibriSpeech for text-to-speech," in Proc. Interspeech, 2019, pp. 1526-1530.