INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Automatic Speech Disentanglement for Voice Conversion using Rank Module and Speech Augmentation

Zhonghua Liu[1], *Shijun Wang*[2], *Ning Chen*[1,*]

[1]East China University of Science and Technology, Shanghai, China
[2]University of St.Gallen, St.Gallen, Switzerland

y80210144@mail.ecust.edu.cn, shijun.wang@unisg.ch, nchen@ecust.edu.cn

Abstract

Voice Conversion (VC) converts the voice of a source speech to that of a target while maintaining the source's content. Speech can be mainly decomposed into four components: content, timbre, rhythm and pitch. Unfortunately, most related works only take into account content and timbre, which results in less natural speech. Some recent works are able to disentangle speech into several components, but they require laborious bottleneck tuning or various hand-crafted features, each assumed to contain disentangled speech information. In this paper, we propose a VC model that can automatically disentangle speech into four components using only two augmentation functions, without the requirement of multiple hand-crafted features or laborious bottleneck tuning. The proposed model is straightforward yet efficient, and the empirical results demonstrate that our model can achieve a better performance than the baseline, regarding disentanglement effectiveness and speech naturalness.

Index Terms: voice conversion, speech disentanglement, speech augmentation

1. Introduction

Voice Conversion (VC) converts one's voice to the voice of a target speaker while preserving the linguistic content of the source speech [1, 2]. VC brings benefits for many interesting and mature applications, such as the famous movie dubbing, AI anchor, intelligent voice conversion navigation, etc. In recent years, with the development of deep neural networks [3, 4, 5], voice conversion methods are also gradually improving and can reach excellent VC performance.

At present, to achieve successful VC, some approaches [6, 7, 8] use auxiliary models like Automatic Speech Recognition (ASR) or Text-To-Speech (TTS) models to achieve VC. In [9, 10, 11, 12], the authors employ Generative Adversarial Networks (GANs) to teach the decoder to generate speech that sounds like the target speakers. However, GAN-based models are usually hard to train. Disentanglement-based approaches such as [13, 14, 15, 16, 17] aim to split the speech into spoken content and speaker characteristic (i.e. timbre). These models have been widely studied because they can achieve excellent performance without using auxiliary models like ASR and it is easier to train compared with GAN-based VC models.

However, most disentanglement-based approaches neglect the fact that speech carries rhythm and pitch [18] as well as content and timbre. Content is the linguistic information conveyed by the speaker; rhythm indicates how quickly the speaker speaks; pitch is the perceived "highness" or "lowness" of a voice; timbre is perceived as the voice characteristic. All of these critical components carry important information and are entangled in the speech. If models simply consider timbre and ignore other components, the generated speech might be less natural and expressive. As a result, decoupling these four key components is crucial for successful VC.

SpeechSplit [19] and SpeechSplit2.0 [20] are two approaches that attempt to decouple the aforementioned speech components. Both models have a similar architecture, with content, rhythm, pitch encoders and one decoder. For SpeechSplit, the authors feed each encoder different types of hand-crafted features, that are manually designed to contain individual disentangled speech information. However, careful bottleneck tuning (usually utilizing small bottlenecks) is required for successful disentanglement, which sacrifices the quality of the generated speech. SpeechSplit2.0 is built upon SpeechSplit. By employing more signal processing techniques, hand-crafted features that are fed to different encoders can hold more distinct speech information. This strategy allows SpeechSplit2.0 to disentangle the speech without laborious bottleneck tuning.

However, the effectiveness of disentanglement in SpeechSplit2.0 highly depends on the hand-crafted features. It would be preferable if the model is able to automatically disentangle the speech, as it could save time on manually selecting the features. Moreover, an automatic disentanglement process can potentially produce disentangled representations that are more effective and accurate, since it can reduce bias and subjectivity that could be introduced by manual selection of features.

In this paper, we propose a VC approach, whose architecture is similar to SpeechSplit or SpeechSplit2.0. But instead of feeding numerous hand-crafted features, our model can automatically disentangle the speech with only two augmentation transformations (pitch modification and rhythm adjustment), while bottleneck tuning is not required. The idea is inspired by [21, 22], where the authors successfully used the Rank module to enable their models to automatically extract effective disentangled representations from the data. Thereby, it is reasonable to consider using the Rank module for disentangling speech as well. Precisely, we force the model to disentangle speech components by ranking the speech and its augmented version. For example, if we increase the pitch, then given the pitch representations from the pitch encoder, the model should rank the augmented version higher than the original version. At the same time, as the rhythm remains the same after the augmentation, the rhythm information from both versions should be ranked equally. Furthermore, we apply the idea from SimCLR [23] to effectively extract invariant representations from the data, i.e. content information, since content information is independent of the changes in pitch, timbre or rhythm. Particularly, we force the model to attract content representations from the original and the augmented versions, while repelling content represen-

*Corresponding author

10.21437/Interspeech.2023-1602

tations from all other speech data in the mini-batch.

We summarize our contributions as: (1) we propose a voice conversion model that is able to disentangle the speech into content, timbre, rhythm and pitch components; (2) the disentanglement process is automatic, and it does not require bottleneck fine-tuning or various hand-crafted features; (3) our empirical results show that, compared to the state-of-the-art SpeechSplit2.0, the proposed model achieves better disentanglement performance and generates more natural speech.

2. Methodology

We train our model in two steps. First, we train multiple encoders to extract content, rhythm and pitch information from the speech. For the second step, we train the decoder with the timbre information and other extracted speech components.

2.1. Speech Encoders

Like SpeechSplit2.0, to extract content, rhythm and pitch representations from the speech, we employ three encoders in our model. The training pipeline is shown in Fig. 1.

As we can see, we first augment the speech data $\mathbf{X}$ into $\mathbf{X}_{\text{Aug}}$. In our work, we use *SoX*[1] to apply two augmentation functions $\texttt{PitchAug}(\mathbf{X}, \tau^p)$ and $\texttt{RhythmAug}(\mathbf{X}, \tau^r)$ to modify the speech rhythm and pitch, where hyperparameter $\tau \in (0, 1)$ indicates the augmentation intensity. Given a specific augmentation function, $\tau < 0.5$ means negative augmentations (decrease pitch/rhythm), while $\tau > 0.5$ means positive augmentations (increase pitch/rhythm). Lastly, $\tau = 0.5$, which indicates no augmentation is applied. Please keep in mind that during the training, each sample is augmented only once by a randomly selected augmentation function.

We then send speech $\mathbf{X}$ and its augmentation version $\mathbf{X}_{\text{Aug}}$ into our encoders. On the left, $\mathbf{X}$ is fed into content encoder $\mathbf{E_c}$ and rhythm encoder $\mathbf{E_r}$, while its pitch contour $\mathbf{P}$ is fed into pitch encoder $\mathbf{E_p}$. With these three encoders, we can get representations $\mathbf{z}^c$, $\mathbf{z}^r$ and $\mathbf{z}^p$. Each one is a sequence and represents content, rhythm, and pitch information, respectively. Afterward, we apply two Rank modules (linear layers) to map $\mathbf{z}^r$ and $\mathbf{z}^p$ into two individual scores s^r and s^p. Both scores are scales, s^r indicates how fast the speech is, while s^p indicates how high the pitch of the speech is. Meanwhile, the content representation $\mathbf{z}^c$ is first averaged into a vector $\mathbf{h}^c$ for further operations, this is inspired by SimCLR [23], where the authors use projection heads to simplify the high-dimensional image representations and make them easier to compare with other image representations. Similarly, as we can see from the right path, with the augmented version $\mathbf{X}_{\text{Aug}}$, we can get representations $\mathbf{z}^c_{\text{Aug}}$, $\mathbf{z}^r_{\text{Aug}}$ and $\mathbf{z}^p_{\text{Aug}}$; two scores s^r_{Aug} and s^p_{Aug}; one vector $\mathbf{h}^c_{\text{Aug}}$.

In order to force the encoders $\mathbf{E_c}$, $\mathbf{E_r}$ and $\mathbf{E_p}$ to produce disentangled representations. We first feed the rhythm and pitch score differences into a Sigmoid function:

$$d^r = \frac{1}{1 + e^{-(s^r - s^r_{\text{Aug}})}}, \qquad d^p = \frac{1}{1 + e^{-(s^p - s^p_{\text{Aug}})}}, \tag{1}$$

then we apply the rank loss on d^p and d^r:

$$\mathcal{L}^r_{rank} = -\tau^r \log(d^r) - (1 - \tau^r)\log(1 - d^r), \qquad \mathcal{L}^p_{rank} = -\tau^p \log(d^p) - (1 - \tau^p)\log(1 - d^p), \tag{2}$$

[1] https://sox.sourceforge.net/

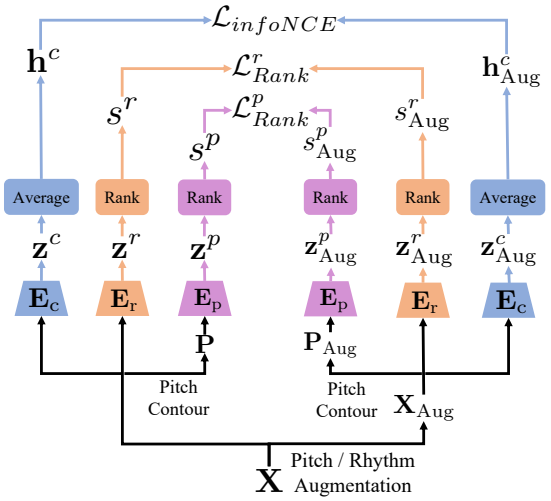

Figure 1: *Speech* $\mathbf{X}$ *and its augmented version* $\mathbf{X}_{Aug}$ *are sent to our encoders to produce pairs* (z^c, z^c_{Aug}), (z^r, z^r_{Aug}), (z^p, z^p_{Aug}) *for content, rhythm and pitch, respectively. After the Rank module, we rank the rhythm and pitch information between the original and the augmented speech with the losses* $\mathcal{L}^r_{rank}$ *and* $\mathcal{L}^p_{rank}$. *Meanwhile, we apply* $\mathcal{L}_{infoNCE}$ *to attract the content representations from the original and the augmented versions, while repelling other content representations in this mini-batch.*

as we mentioned before, τ^r and τ^p are two hyperparameters indicating the augmentation intensity. And our target is to force our encoders to produce disentangled representations by recognizing this augmentation intensity.

As an example, if we perform augmentation $\texttt{PitchAug}$ and increase the pitch ($\tau^p > 0.5$) of $\mathbf{X}$. Then, to decrease the pitch Rank loss $\mathcal{L}^p_{rank}$, the model needs to assign a higher score to s^p_{Aug} than s^p, which means the pitch representation $\mathbf{z}^p$ from the pitch encoder $\mathbf{E_p}$ is encouraged to carry effective pitch information. At the same time, since rhythm information is not modified by augmentation $\texttt{PitchAug}$, τ^r is set to 0.5. Therefore, the model has to assign the same score to both s^r and s^r_{Aug} to decrease the loss $\mathcal{L}^r_{rank}$. In other words, the rhythm encoder $\mathbf{E_r}$ is forced to be insensitive to the change of pitch, i.e., rhythm representation $\mathbf{z}^r$ and pitch representation $\mathbf{z}^p$ are disentangled from each other.

Furthermore, to make sure the content encoder $\mathbf{E_c}$ only produces the content-related representations, we apply the infoNCE loss from [23, 24] on $\mathbf{h}^c$ and $\mathbf{h}^c_{\text{Aug}}$, we make a slightly rephrased to Equation (1) from [23]:

$$\mathcal{L}_{infoNCE} = -\log \frac{\text{sim}(\mathbf{h}^c, \mathbf{h}^c_{\text{Aug}})}{\text{sim}(\mathbf{h}^c, \mathbf{h}^c_{\text{Aug}}) + \sum_{\mathbf{X}_{\text{Neg}}} \text{sim}(\mathbf{h}^c, \mathbf{h}^c_{\text{Neg}})}, \tag{3}$$

where $\text{sim}(\cdot, \cdot)$ is the exponential dot product of the two inputs, with a temperature parameter t. $\mathbf{X}_{\text{Neg}}$ denotes all speech samples in a mini-batch except $\mathbf{X}$, while h^c_{Neg} denotes the content vectors derived from $\mathbf{X}_{\text{Neg}}$.

With $\mathcal{L}_{infoNCE}$, we force the content representations from the same speech to be similar, regardless of which augmentation methods we use. Meanwhile, content representations derived from different speech samples are forced to be dissimilar. Such loss enables the content encoder $\mathbf{E_c}$ to be only focused on the invariant information of the speech, i.e. content information.

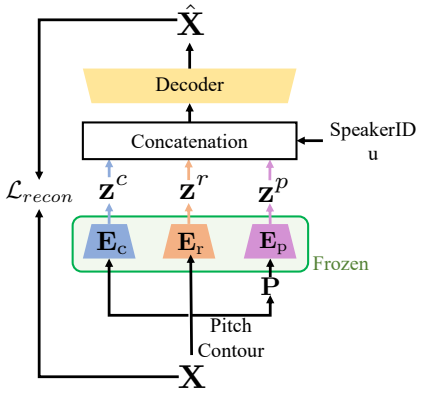

Figure 2: *The training of the voice conversion.*

Finally, we train the three encoders with the loss:

$$\mathcal{L}_{encoder} = \mathcal{L}^r_{rank} + \mathcal{L}^p_{rank} + \mathcal{L}_{infoNCE}. \quad (4)$$

2.2. Speech Decoder

For VC, we need to provide timbre information to represent speaker characteristics. We follow SpeechSplit2.0 and use speaker ID as timbre information. Then, we need to train a decoder to perform voice conversion. The training pipeline is shown in Fig. 2. From the input speech $\mathbf{X}$, the content, rhythm and pitch representations are extracted by pre-trained and frozen encoders. Meanwhile, speaker ID u is converted to speaker embedding. The rest is similar to SpeechSplit2.0, all these representations are concatenated and fed to the decoder. And the decoder generates $\hat{\mathbf{X}}$, and we apply the reconstruction loss (mean square error) on $\mathbf{X}$ and $\hat{\mathbf{X}}$:

$$\mathcal{L}_{recon} = ||\mathbf{X} - \hat{\mathbf{X}}||^2. \quad (5)$$

2.3. Implementation

Our model's architecture is the same as SpeechSplit2.0 [20]. But rather than training all of the modules at once, we train our model in two steps, because we found it can achieve better performance. In the first step (Sec. 2.1), we train our encoders for 30k iterations, and set the learning rate to 1e-6, while the temperature t in Eq. 3 is 0.1. For the second step (Sec. 2.2), The decoder is trained using a 1e-4 learning rate over 600k iterations. Adam optimizer is used for both steps. Demo audio samples can be found at `https://hhhuazi.github.io/`

3. Experiments

3.1. Experiment setup

Dataset and Data Preprocessing: We use VCTK corpus [25] to train and evaluate our models, which is a popular dataset for VC tasks. VCTK includes 109 English speakers, each speaker reads about 400 sentences. All audio recordings are downsampled to 16kHz. And we extract Mel-Spectrograms from waveform files by using a 50-millisecond window and a 50 percent overlap ratio, with 80 Mel Coefficients.
Baseline: We employ SpeechSplit2.0 [20] as our baseline. SpeechSplit2.0 is a state-of-the-art VC model that is able to disentangle speech like the proposed model.
Evaluational Setup: We use HiFi-GAN [26] to convert generated Mel-Spectrograms to waveforms. To perform objective and subjective tests, we randomly select 20 unseen source and target pairs to generate synthesized samples from both models. 20 subjects participate in the subjective evaluations.

3.2. Conversion Rate

To evaluate the disentanglement effect, we follow [20] and evaluate the conversion rate of the baseline and our model. In this subjective test, each subject is first asked to listen to the source and target reference speech in random order. Then, by listening to a synthesized sample, the subject selects which reference speech has more similar components (pitch, rhythm, or timbre) to the synthesized sample. Afterward, we can compute the conversion rate, defined as the percentage of answers selecting the target reference. The conversion rate can reflect the effectiveness of the decoupling. For example, if we only convert the timbre, then the timbre conversion rate is supposed to be high, while the pitch and rhythm conversion rates should be low. In our experiments, we test the conversion rate on all 7 combinations (Pitch-only, Rhythm-only, Timbre-only, Pitch+Rhythm, Pitch+Timbre, Rhythm+Timbre, Pitch+Rhythm+Timbre). The results are shown in Fig. 3.

As we can see, although both models have the ability to disentangle the speech, our model significantly outperforms the baseline for almost all combinations. Such results demonstrate that despite our encoders sharing the same architecture with the baseline, by training our encoders through the Rank objective (Eq. 2), our model is more effective at extracting disentangled pitch, rhythm, and timbre information. It is worth mentioning that although both models use speaker ID to represent timbre, our model's disentanglement performance of timbre exceeds that of the baseline (e.g. the conversion rate results of Timbre-only). This may be due to our model's pitch, rhythm, or content representations containing less source's timbre information, resulting in less negative influence on the timbre conversion when converting the source speech's timbre to the target's.

3.3. Naturalness MOS

We then conduct Mean Opinion Score (MOS) to evaluate the naturalness of generated speech. We ask the subjects to score the speech's naturalness on a scale of 1 (Bad) to 5 (Excellent). The results are shown in Fig. 4. All 7 combinations of different speech components have been tested. As we can see, our model largely outperforms the baseline and can produce more natural speech. High naturalness MOS scores indicate that the pitch, rhythm or timbre from the target speech is less likely to damage the naturalness of the generated speech.

3.4. Character Error Rate

We continue to carry out Character Error Rate (CER) experiments. A pre-trained ASR model [27] is applied to produce CER. CER quantifies the linguistic difference between original and converted speech to detect the ability of content preservation by models.

The results are shown in Tab. 1. As we can see, the error rate of our model is less than the baseline for almost all combinations, indicating that our model can preserve the content information better. Such results demonstrate the effectiveness of the infoNCE loss (Eq. 3) we applied to force the content encoder to extract disentangled content representations. It is worth mentioning that the CER results are relatively high when converting the pitch component, possibly due to length mismatch between the source and target speech. If the target speech is longer than the source, conflicts between the target's long pitch

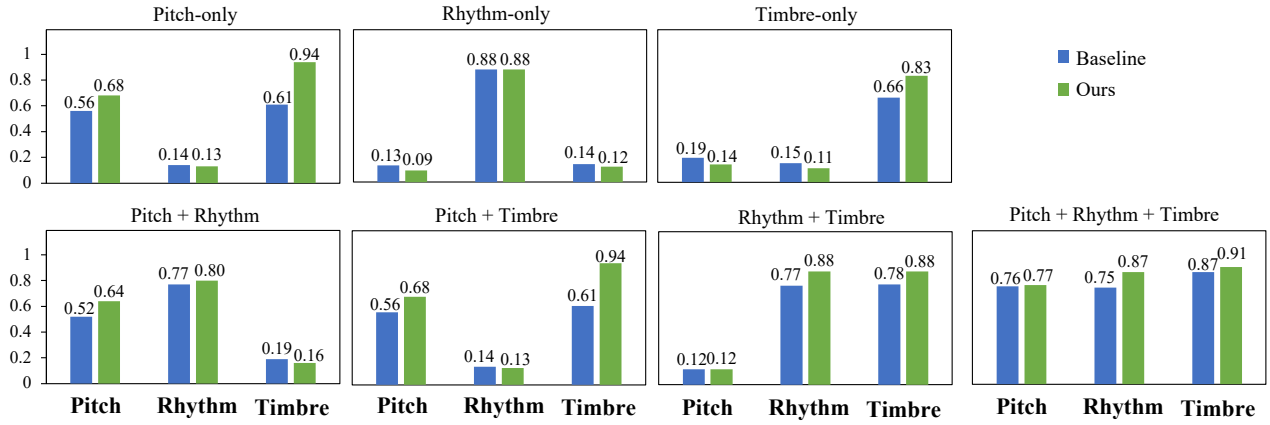

Figure 3: *Conversion rate*

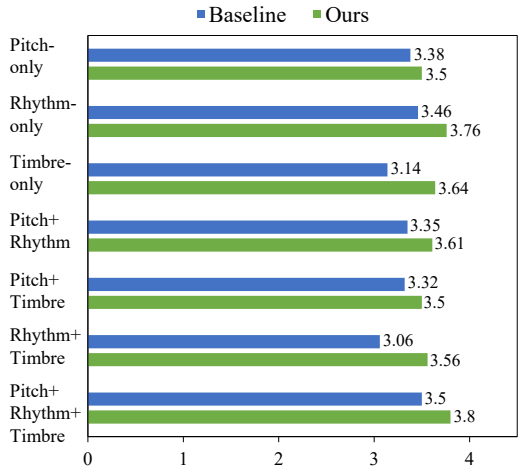

Figure 4: *Naturalness MOS evaluation*

and the original short rhythm or content information might occur, introducing noise into the generated speech.

3.5. Pearson Correlation Coefficient on Pitch

We further verify the effectiveness of the pitch representation by measuring the Pearson Correlation Coefficient (PCC) [28] of logF0. PCC measures the pitch correlation between two speech samples. The larger the value, the more similar the pitch patterns are between the two samples.

The results are shown in Tab. 2. It is important to note that only two samples of the same length can be used to compute PCC. Thus, we split the results into two parts. For Pitch-only and Pitch+Timbre combinations, the converted and source speech have the same length. We expect a lower pitch correlation between converted and source speech due to pitch conversion. The PCC results of our model are lower than the baseline for these two combinations, indicating more effective pitch conversion. Conversely, when converting the rhythm simultaneously, the length will be the same as the target, and we expect a higher PCC due to pitch transfer from the target speech. As we can see, Pitch+Rhythm and Pitch+Rhythm+Timbre combinations exhibit higher correlations than the baseline. In conclusion, our model can produce more disentangled and effective pitch representations.

Table 1: *CER (%)*

Combination	Baseline	Ours
Pith-only	64.5	**62.3**
Rhythm-only	31.8	**22.7**
Timbre-only	50.1	**46.0**
Pitch+Rhythm	**34.8**	36.2
Pitch+Timbre	61.8	**60.0**
Rhythm+Timbre	27.6	**20.5**
Pitch+Rhythm+Timbre	30.5	**23.6**

Table 2: *Impact of pitch representation*

Combination	Baseline	Ours
PCC between the source and the converted (the lower the better)		
Pitch-only	0.45	**0.41**
Pitch+Timbre	0.42	**0.40**
PCC between the target and the converted (the higher the better)		
Pitch+Rhythm	0.41	**0.49**
Pitch+Rhythm+Timbre	0.53	**0.69**

4. Conclusions

In this paper, we propose a VC model that is able to automatically disentangle the speech into four components without manually feeding various hand-crafted features or laborious bottleneck fine-tuning. The disentanglement is derived from the Rank Module, which relies only on two augmentation transformations. The subjective and objective evaluation shows that compared to the baseline, our model can achieve a better disentanglement performance and produce more natural speech.

5. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China [grant number 61771196].

6. References

[1] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. H. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," *ArXiv*, vol. abs/1804.04262, 2018.

[2] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 132–157, 2020.

[3] S. L. Taylor, A. Kato, I. Matthews, and B. P. Milner, "Audio-to-visual speech conversion using deep neural networks," in *Interspeech*, 2016.

[4] C.-C. Lo, S.-W. Fu, W.-C. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H. Wang, "Mosnet: Deep learning based objective assessment for voice conversion," *ArXiv*, vol. abs/1904.08352, 2019.

[5] 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," in *Interspeech*, 2019.

[6] W.-C. Huang, T. Hayashi, X. Li, S. Watanabe, and T. Toda, "On prosody modeling for asr+tts based voice conversion," *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 642–649, 2021.

[7] A. T. Liu, P. chun Hsu, and H. yi Lee, "Unsupervised end-to-end learning of discrete linguistic units for voice conversion," in *Interspeech*, 2019.

[8] W.-C. Huang, T. Hayashi, S. Watanabe, and T. Toda, "The sequence-to-sequence baseline for the voice conversion challenge 2020: Cascading asr and tts," *ArXiv*, vol. abs/2010.02434, 2020.

[9] F. Fang, J. Yamagishi, I. Echizen, and J. Lorenzo-Trueba, "High-quality nonparallel voice conversion based on cycle-consistent adversarial network," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5279–5283, 2018.

[10] T. Kaneko and H. Kameoka, "Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks," *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 2100–2104, 2018.

[11] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: non-parallel many-to-many voice conversion using star generative adversarial networks," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 266–273, 2018.

[12] Y. A. Li, A. A. Zare, and N. Mesgarani, "Starganv2-vc: A diverse, unsupervised, non-parallel framework for natural-sounding voice conversion," in *Interspeech*, 2021.

[13] D.-Y. Wu, Y.-H. Chen, and H. yi Lee, "Vqvc+: One-shot voice conversion by vector quantization and u-net architecture," *ArXiv*, vol. abs/2006.04154, 2020.

[14] R. Xiao, H. Zhang, and Y. Lin, "Dgc-vector: A new speaker embedding for zero-shot voice conversion," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6547–6551, 2022.

[15] Z. Wang, Q. Xie, T. Li, H. Du, L. Xie, P. Zhu, and M. Bi, "One-shot voice conversion for style transfer based on speaker adaptation," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6792–6796, 2021.

[16] D. Wang, L. Deng, Y. T. Yeung, X. Chen, X. Liu, and H. M. Meng, "Vqmivc: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion," *ArXiv*, vol. abs/2106.10132, 2021.

[17] M. Luong and V.-A. Tran, "Many-to-many voice conversion based feature disentanglement using variational autoencoder," *ArXiv*, vol. abs/2107.06642, 2021.

[18] W. Gan, B. Wen, Y. Yan, H. Chen, Z. Wang, H. Du, L. Xie, K. Guo, and H. Li, "Iqdubbing: Prosody modeling based on discrete self-supervised speech representation for expressive voice conversion," *ArXiv*, vol. abs/2201.00269, 2022.

[19] K. Qian, Y. Zhang, S. Chang, D. Cox, and M. A. Hasegawa-Johnson, "Unsupervised speech decomposition via triple information bottleneck," in *International Conference on Machine Learning*, 2020.

[20] C. H. Chan, K. Qian, Y. Zhang, and M. A. Hasegawa-Johnson, "Speechsplit2.0: Unsupervised speech disentanglement for voice conversion without tuning autoencoder bottlenecks," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6332–6336, 2022.

[21] Y. Souri, E. Noury, and E. Adeli, "Deep relative attributes," in *Asian Conference on Computer Vision*, 2015.

[22] S. Wang, D. Kostadinov, and D. Borth, "Zero-shot voice conversion via self-supervised prosody representation learning," *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 01–08, 2021.

[23] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," *ArXiv*, vol. abs/2002.05709, 2020.

[24] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *ArXiv*, vol. abs/1807.03748, 2018.

[25] C. Veaux, J. Yamagishi, and K. Macdonald, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2017.

[26] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *ArXiv*, vol. abs/2010.05646, 2020.

[27] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," *ArXiv*, vol. abs/2005.08100, 2020.

[28] G. Nahler, "Pearson correlation coefficient," *Definitions*, 2020.