ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096233

TRANSPLAYER: TIMBRE STYLE TRANSFER WITH FLEXIBLE TIMBRE CONTROL

Yuxuan Wu, Yifan He, Xinlu Liu, Yi Wang, Roger B. Dannenberg

Carnegie Mellon University

ABSTRACT

Music timbre style transfer aims at replacing the instrument timbre in a solo recording with another instrument, while preserving the musical content. Existing GAN-based methods can only achieve timbre style transfer between two given timbres. Inspired by the practice in voice conversion, we propose TransPlayer, which uses an autoencoder model with one-hot representations of instruments as the condition, and a Diffwave model trained especially for music synthesis. We evaluate our model in both the one-to-one transfer task and the many-to-many transfer task. The results prove that our method is able to provide one-to-one style transfer outputs comparable with the existing GAN-based method, and can transfer among multiple timbres with only one single model.

Index Terms— Timbre, style transfer, music synthesis

1. INTRODUCTION

Music style transfer is an important topic in applying AI technologies to music creation. In music style transfer, it's typically assumed that music consists of two complementary components, namely content and style [1]. By separating and recombining them, we can create new music that inherits musical attributes from different origins. Different definitions of content and style lead to different sub-topics, including performance style transfer, composition style transfer, etc.

Timbre style transfer is an intriguing sub-topic in music style transfer. Timbre is an essential element in a musical sound that can differentiate instruments or human voice sounds. Listeners can tell the difference between two instruments by their timbres, even when playing the same note with the same pitch, loudness and duration. Timbre is hard to model as it shows great differences across instruments in both time and frequency domains [2]. While there are well-designed physical models that simulate sound production using formulas and parameters [3, 4, 5], high-fidelity sample libraries are still preferred in virtual instrument plugins. In timbre style transfer, Generative Adversarial Network (GAN) methods are proved more successful than explicitly extracting timbre information from audio [6, 7]. However, they can only work between two given instruments (one-to-one transfer), without the possibility of more flexible timbre control. Models like NSynth and Differentiable Digital Signal Processing (DDSP) can also perform timbre transfer, but only limited to monophonic music [8, 9].

In this paper, we explore the problem of timbre style transfer, focusing on (1) transferring the timbre while preserving the musical content and the sound quality (2) flexible transfer among a number of instruments (many-to-many transfer). We refer to the pitches, loudness and durations as the content of music, and the instrument's timbre as the style. We take a simplified view of timbre, assuming that instruments and timbres are equivalent. We do not attempt to model the possibility of timbral variation among the sounds of individual instruments.

We propose TransPlayer, a conditional autoencoder-based model working on constant-Q transform (CQT) representations of music. The idea is to train an autoencoder with a style embedding layer, then convert the decoder output back to audio using a Diffwave model [10]. Experiment results show that the proposed model can successfully perform many-to-many timbre transfer, with the result quality comparable to state-of-the-art one-to-one timbre transfer models.

2. RELATED WORK

Many previous deep learning models explored various aspects of music style transfer, but only a few of them focused on timbre-related features in polyphonic music. Verma and Smith were the first to apply deep learning to timbre transfer [11]. A later work [12] proposed a WaveNet autoencoder structure to translate music waveforms across multiple style domains such as instruments. Some other works adopted GAN-based models, which do not require paired training data [6, 7]. They used consistency loss terms to restore the content during reconstruction. A Variational Autoencoder (VAE) method tried to transfer timbre among multiple domains with only one model [13], but it's only applicable to monophonic music. Some other studies focused on modeling the timbre embedding space with representation learning [8, 14]. DDSP proposed a novel idea for audio components' disentanglement and was able to do timbre transfer but limited to monophonic music [9]. Another work proposed the Guided Adversarial Autoencoder (GAAE), which is capable of generating audio using very few labeled data [15]. A semi-supervised approach was proposed for many-to-many timbre transfer, but its training process relied on paired data [16].

Although not tested with music audio, similar problems

have been well explored in voice [17]. AutoVC used a succinct autoencoder with a well-designed information bottleneck to disentangle speaker information and content information [18]. A revisional work proposed alteration invariant content loss and adversarial training for better robustness [19]. StarGAN-VC and its successors overcame the drawback of CycleGAN that it can only achieve one-to-one conversion [20, 21, 22, 23]. Some other works modeled voice conversion as a Sequence-to-Sequence (Seq2Seq) problem [24, 25]

Theoretically, we can convert Short-time Fourier Transform (STFT) representations back to audio if we have both the amplitude and phase information. However, this is not realistic in many situations. Alternative methods for audio generation include non-parametric algorithms such as the Griffin-Lim algorithm [26] and the WORLD vocoder [27]. Neural network-based methods for audio generation include the well-known WaveNet, [28], WaveGAN and Hifi-GAN [29, 30], and Diffwave which applied diffusion model to audio synthesis [10]. In this work, we selected Diffwave as our waveform generator because it's relatively easier to train than GAN-based models, and much faster than WaveNet [10].

3. METHOD

We describe a system for many-to-many timbre transfer, which works on both monophonic and polyphonic music. As illustrated in Fig. 1, we use constant-Q transform (CQT) to obtain spectral representations of music, and use an autoencoder to encode the content. Both the encoder and decoder are conditioned by a style latent code. The decoder generates CQT representations and they are converted back to the waveform with a trained Diffwave model.

3.1. Data Representation

Constant-Q transform can transform data series from the time domain to the frequency domain [31]. Unlike STFT, CQT is especially suitable for representing music audio, since the fundamental frequency of a note increases approximately exponentially with the increase in pitch. CQT can be described as a bank of filters logarithmically spaced in central frequency and bandwidth. The logarithmic spacing of filters makes CQT transposition equivariant. This means different pitches played by the same instrument have a similar harmonic pattern, which is closely related to the spectral features in a timbre. We use a hop length of 16ms, and obtain an 84-dimensional log-scale CQT representation for every frame, in which there are 12 dimensions for each of the 7 octaves.

3.2. Timbre Style Transfer Autoencoder

Consider two domains $\mathcal{X}$ and $\mathcal{Y}$, where x and y are two samples from $\mathcal{X}$ and $\mathcal{Y}$ respectively. In a style transfer problem, $\mathcal{X}$ and $\mathcal{Y}$ share the same content space $\mathcal{C}$, but differ in their

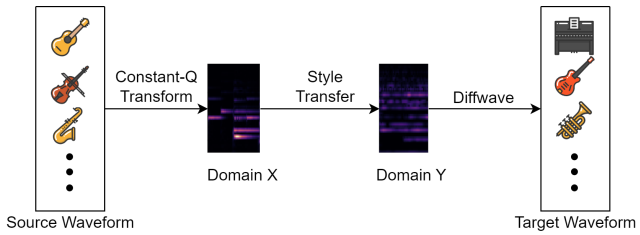

Fig. 1. Pipeline of TransPlayer

styles $\mathcal{S}$. Since $\mathcal{C}$ and $\mathcal{S}$ are complementary information, a sample of x can be generated by combining a content code $c \in \mathcal{C}$ and a style code $s \in \mathcal{S}$. We learn the mapping function from every domain $\mathcal{X}$ to the content space $\mathcal{C}$ conditioned on $\mathcal{S}$, and its inverse mapping function from $\mathcal{C}$ to $\mathcal{X}$. With the assumption that instruments and timbres are equivalent, every domain $\mathcal{X}$ has a fixed style code $s_{\mathcal{X}}$.

We design our model based on AutoVC [18], which is a successful many-to-many voice conversion model using only one autoencoder conditioned by pre-trained speaker embeddings [32]. Here we add a style embedding layer to adapt it to the timbre transfer problem.

The style embedding layer E_s learns the representation of style code s of any given timbre, formulated as $s_{\mathcal{X}} = E_s(x)$. This is a substitution for the X-vector extractor proposed in AutoVC [18]. It takes a one-hot representation of timbre as the input and outputs a latent style code as the style embedding. This style code s is used as the condition in both the content encoder E_c and the decoder D. The style code should be able to preserve sufficient information of the timbres.

The content encoder E_c learns the mapping function from every feature domain $\mathcal{X}$ to the content space $\mathcal{C}$. For every sample $x \in \mathcal{X}$, we have a content code $c_x = E_c(x, s_{\mathcal{X}})$. The goal of the content encoder is to produce different content codes for different music pieces of the same instrument, and produce the same content code for the same piece played by different instruments. The input of E_c is the CQT representation concatenated with style code s. It is fed into three convolutional blocks, each consisting of a 5×1 convolutional layer with 512 channels, a batch normalization layer and a ReLU

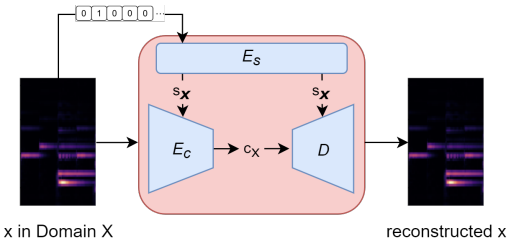

Fig. 2. Reconstruction in training. The content encoder Ec and the decoder D are conditioned on the same style code s.

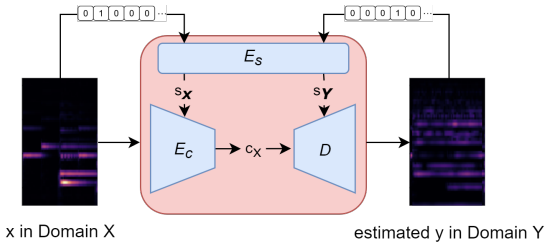

Fig. 3. Transfer between domains. The decoder D is conditioned on the target style code $s_{\mathcal{Y}}$. The output is fed into Ec again to compute the cross-domain content consistency loss.

layer. The output is fed into two bidirectional LSTM layers with a latent dimension of 32. Then we downsample the LSTM output along the time axis as a dimension reduction. What we finally attain here is the information bottleneck [18]. It should be wide enough to preserve all the musical content, but setting the bottleneck too wide will make the content code blended with style-related information. The bottleneck width is determined by the LSTM output dimension and the downsampling rate. In our case, the LSTM output dimension and the downsampling rate are empirically set to 32.

The decoder D is almost the inverse process of the content encoder E_c. The content code is first concatenated with the target style code, then upsampled by a factor of 32. We feed the concatenated embeddings into an LSTM layer with 512 channels. It's followed by three convolutional blocks identical to the ones in the encoder. Then we feed the output into two LSTM layers with an output dimension of 1024. Lastly, the feature is linearly projected to 84 dimensions.

In training, the model tries to first encode the musical content in x, and then transfer the content code c_x back to its original timbre, as shown in Fig. 2. We want this reconstruction process to be as accurate as possible. Here we have the reconstruction loss, formulated as follows:

$$\begin{aligned}\mathcal{L}_{recon} &= \mathbb{E}[|x - \hat{x}|_1] \\ &= \mathbb{E}[|x - D(E_c(x, s_{\mathcal{X}}), s_{\mathcal{X}})|_1]\end{aligned}$$

where $\hat{x}$ denotes the reconstruction result of x. L1 rather than L2 loss is adopted here for a sharper generation result. Also, the content code should remain unchanged when we feed the reconstruction back into E_c, which can be described using the reconstruction content consistency loss:

$$\begin{aligned}\mathcal{L}_{c\_consistency\_self} &= \mathbb{E}[|c_x - \hat{c_x}|_1] \\ &= \mathbb{E}[|c_x - E_c(\hat{x}, s_{\mathcal{X}})|_1]\end{aligned}$$

Finally, since the conditions on the E_c and D are actually different in testing, we add an extra step in training to simulate the testing scenario, as shown in Fig. 3. Denoting the expected transferred results as y, and the real results as $\hat{y} = D(E_c(x, s_{\mathcal{X}}), s_{\mathcal{Y}})$, we should have $c_x = c_{\hat{y}}$. This leads us to the cross-domain content consistency loss:

$$\begin{aligned}\mathcal{L}_{c\_consistency\_cross} &= \mathbb{E}[|c_x - c_{\hat{y}}|_1] \\ &= \mathbb{E}[|c_x - E_c(\hat{y}, s_{\mathcal{Y}})|_1]\end{aligned}$$

In every training iteration, we randomly select a target domain $\mathcal{Y}$ as the target style, so that there isn't an imbalance among different transfer pairs. The final training objective is the sum of the above loss functions.

3.3. Waveform Generation

One reason why CQT representations are not widely used in audio generation tasks is that CQT does not have a direct inverse transform like STFT does. However, this drawback can be overcome by using neural networks to generate the waveform. Among these models, Diffwave stands out for its versatility and fast generation speed in speech [10], but only a few works investigated its ability in generating music audio [33].

Diffwave is one instance of diffusion model in audio synthesis. It is composed of a stack of residual layers with a bidirectional dilated convolution architecture. For generation, the feature is upsampled to the same dimension as the expected waveform. Then we sample the transition distributions in the reverse process step by step to obtain the waveform. In our case, the generation is conditioned on CQT representations.

4. EXPERIMENTS AND EVALUATION

To ensure the quality, the audio files are synthesized using commercial-level virtual instruments and MIDI files from the POP909 dataset [34] and the MAESTRO dataset [35]. Instruments used include the piano, electric piano, flute, acoustic guitar, harp, organ, trumpet, and viola. The data of each instrument are about 8,000 seconds long in total. 90% of the data are used for training and the other 10% are for testing.

4.1. Experiment Settings

We train the conditional autoencoder for 1.5M iterations at a batch size of 2 [18]. The training objective is optimized with the Adam optimizer at a learning rate halved every 0.25M iterations starting from 1e-4. We trained the Diffwave model conditioned on CQT representations for 14000 epochs at a batch size of 32. The initial learning rate was also 1e-4 [1].

4.2. Evaluation of One-to-one Transfer

To evaluate our system, we first compare it to a baseline on one-to-one timbre style transfer. We consider a GAN-based model as our baseline [7]. It is able to generate satisfying results without paired data. Also, it's an open-source project so we can reproduce their results following the original settings.

[1]Code and examples can be viewed at `https://github.com/Irislucent/TransPlayer`.

We consider the task of timbre transfer between piano and guitar, which was a better-performed sub-task described in the original work. We trained the baseline for 500k iterations, and used the phase extracted from the source signal for waveform generation. We conducted anonymous listening tests on Amazon Mechanical Turk (AMT) to evaluate the system from human perspective. The 50 participants were presented first with three original music clips, and then their two transferred versions without knowing how they were generated. The Mean Opinion Scores (MOS) are given in three dimensions ranging from 1 to 5, including (1) Success in transfer (ST): how well the timbre of the transferred version matches the target perceptually. (2) Content preservation (CP): how well the musical content of the transferred version matches the original version. (3) Sound quality (SQ): how good the transferred audio quality is overall. The subjective scores in 1 showed that the proposed system can perform nearly as well as, if not significantly better than our baseline.

Table 1. MOS of One-to-one Timbre Transfer Comparing the Baseline and TransPlayer

Task	Piano to Guitar			Guitar to Piano		
Model	ST	CP	SQ	ST	CP	SQ
Baseline	3.66	**3.96**	3.71	3.92	**3.88**	3.58
TransPlayer	**3.68**	3.80	**3.84**	3.92	3.83	**3.64**

For an objective evaluation, we employed an instrument classifier by training an AlexNet-like network on 1-second audio segments sliced from our dataset to classify whether the transferred audio belongs to piano sound or guitar sound. The classifier outputs a scalar value passed through a Sigmoid function, which represents the classification likelihood per segment. We report this classification likelihood as a level of confidence. Segment-wise classification accuracy and the mean classification likelihood along segments are given as the results in Table. 2. The classification results showed that TransPlayer observably surpassed our one-to-one baseline in terms of the similarity to the target timbre. However, it falls slightly behind in content preservation quality. Generally, our model shows better sound quality, which shows that Diffwave is a promising direction for music synthesis.

Table 2. Classification Results of One-to-one Timbre Transfer Comparing the Baseline and TransPlayer

Task	Piano to Guitar		Guitar to Piano	
Model	Accuracy	Confidence	Accuracy	Confidence
Ground Truth	96.58%	95.50%	98.99%	97.74%
Baseline	89.91%	85.52%	91.35%	89.79%
TransPlayer	**91.72%**	**86.32%**	**93.14%**	**90.71%**

4.3. Evaluation of Many-to-many Transfer

We then evaluate our proposed system on the task of many-to-many timbre style transfer. We select random pieces for each instrument in the testing set, and transfer them to each of the remaining seven instruments, as well as the original instrument (reconstruction), resulting in $8 \times 8 = 64$ transfer pairs. Participants are presented first with the sounds of the original instruments, and then with the transfer results. The MOS scores are given in the same three dimensions (ST, CP, SQ) and then averaged as the final evaluation scores. We plot the listening test results of many-to-many transfer as a heatmap in Fig. 4. It's observed that certain transfer pairs have better performance than others, and the reconstruction quality is generally better than the transfer quality. Transferring from other instruments to the organ has the best performance, while transferring to harp sounds the worst among all. A possible explanation is that the organ is far from other points in the timbre space, which means it shares few similarities in timbre characteristics with other instruments. On the other hand, the harp timbre has more characteristics in common with other instruments. Another possible reason is that organ notes do not have decay, whereas harp notes have exponential decay, which is a long-term dependency that requires learning to ignore the source instrument's amplitude variation.

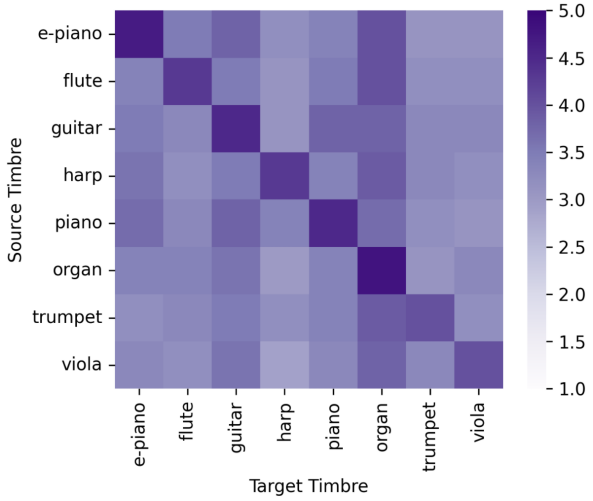

Fig. 4. MOS of Many-to-many Timbre Transfer Results.

5. CONCLUSION

In this paper, we propose TransPlayer, a timbre style transfer model based on the autoencoder structure. The model works on CQT representations and is conditioned on a style embedding layer that is simultaneously trained with the model. We convert CQT representations back to the waveform with a Diffwave model. The model can generate high-quality results comparable with the state-of-the-art one-to-one timbre transfer model on the given transfer pair. On top of that, its ability extends to a much wider range, enabling transfer among multiple domains for more flexible timbre control.

6. REFERENCES

[1] Gus G Xia and Shuqi Dai, "Music style transfer: A position paper," in *Proceedings of the 6th International Workshop on Musical Metacreation. Salamanca, Spain: MUME*, 2018, p. 6.

[2] David L Wessel, "Timbre space as a musical control structure," *Computer music journal*, pp. 45–52, 1979.

[3] Kevin Karplus and Alex Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music Journal*, vol. 7, no. 2, pp. 43–55, 1983.

[4] Michael E McIntyre, Robert T Schumacher, and James Woodhouse, "On the oscillations of musical instruments," *The Journal of the Acoustical Society of America*, vol. 74, no. 5, pp. 1325–1345, 1983.

[5] Julius O Smith, "Physical modeling using digital waveguides," *Computer music journal*, vol. 16, no. 4, pp. 74–91, 1992.

[6] Sicong Huang, Qiyang Li, Cem Anil, Xuchan Bao, Sageev Oore, and Roger B. Grosse, "Timbretron: A wavenet(cycleGAN(CQT(audio))) pipeline for musical timbre transfer," in *International Conference on Learning Representations*, 2019.

[7] Chien-Yu Lu, Min-Xin Xue, Chia-Che Chang, Che-Rung Lee, and Li Su, "Play as you like: Timbre-enhanced multi-modal music style transfer," in *Proceedings of the aaai conference on artificial intelligence*, 2019, vol. 33, pp. 1061–1068.

[8] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1068–1077.

[9] Jesse Engel, Lamtharn (Hanoi) Hantrakul, Chenjie Gu, and Adam Roberts, "Ddsp: Differentiable digital signal processing," in *International Conference on Learning Representations*, 2020.

[10] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020.

[11] Prateek Verma and Julius O Smith, "Neural style transfer for audio spectograms," *arXiv preprint arXiv:1801.01589*, 2018.

[12] Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman, "A universal music translation network," in *International Conference on Learning Representations*, 2018.

[13] Adrien Bitton, Philippe Esling, and Axel Chemla-Romeu-Santos, "Modulated variational auto-encoders for many-to-many musical timbre transfer," *arXiv preprint arXiv:1810.00222*, 2018.

[14] Jong Wook Kim, Rachel Bittner, Aparna Kumar, and Juan Pablo Bello, "Neural music synthesis for flexible timbre control," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 176–180.

[15] Kazi Nazmul Haque, Rajib Rana, and Björn W Schuller, "High-fidelity audio generation and representation learning with guided adversarial autoencoder," *IEEE Access*, vol. 8, pp. 223509–223528, 2020.

[16] Yu-Chen Chang, Wen-Cheng Chen, and Min-Chun Hu, "Semi-supervised many-to-many music timbre transfer," in *Proceedings of the 2021 International Conference on Multimedia Retrieval*, 2021, pp. 442–446.

[17] Eric Grinstein, Ngoc Q. K. Duong, Alexey Ozerov, and Patrick Pérez, "Audio style transfer," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 586–590.

[18] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5210–5219.

[19] Yunyun Wang, Jiaqi Su, Adam Finkelstein, and Zeyu Jin, "Controllable speech representation learning via voice conversion and aic loss," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6682–6686.

[20] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 266–273.

[21] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," in *INTERSPEECH*, 2019.

[22] Ruobai Wang, Yu Ding, Lincheng Li, and Changjie Fan, "One-shot voice conversion using star-gan," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7729–7733.

[23] Yinghao Aaron Li, Ali Zare, and Nima Mesgarani, "Starganv2-vc: A diverse, unsupervised, non-parallel framework for natural-sounding voice conversion," *arXiv preprint arXiv:2107.10394*, 2021.

[24] Jing-Xuan Zhang, Zhen-Hua Ling, Li-Juan Liu, Yuan Jiang, and Li-Rong Dai, "Sequence-to-sequence acoustic modeling for voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 3, pp. 631–644, 2019.

[25] Jing-Xuan Zhang, Zhen-Hua Ling, and Li-Rong Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 540–552, 2019.

[26] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.

[27] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[28] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[29] Chris Donahue, Julian McAuley, and Miller Puckette, "Adversarial audio synthesis," *arXiv preprint arXiv:1802.04208*, 2018.

[30] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17022–17033, 2020.

[31] Judith C Brown, "Calculation of a constant q spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[32] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[33] Nikhil Kandpal, Oriol Nieto, and Zeyu Jin, "Music enhancement via image translation and vocoding," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3124–3128.

[34] Ziyu Wang, Ke Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, Xianbin Gu, and Gus Xia, "Pop909: A pop-song dataset for music arrangement generation," *arXiv preprint arXiv:2008.07142*, 2020.

[35] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *International Conference on Learning Representations*, 2019.