TIMBRETRON: A WAVENET(CYCLEGAN(CQT(AUDIO))) PIPELINE FOR MUSICAL TIMBRE TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we address the problem of musical timbre transfer, where the goal is to manipulate the timbre of a sound sample from one instrument to match another instrument while preserving other musical content, such as pitch, rhythm, and loudness. In principle, one could apply image-based style transfer techniques to a time-frequency representation of an audio signal, but this depends on having a representation that allows independent manipulation of timbre as well as high-quality waveform generation. We introduce TimbreTron, an audio processing pipeline which combines three powerful ideas from different domains: Constant Q Transform (CQT) spectrogram for audio representation, a variant of CycleGAN for timbre transfer and WaveNet-Synthesizer for high quality audio generation. We verified that CQT TimbreTron in principle and in practice is more suitable than its STFT counterpart, even though STFT is more commonly used for audio representation. Based on human perceptual evaluations, we confirmed that timbre was transferred recognizably while the musical content was preserved by TimbreTron.

1 INTRODUCTION

Timbre is a perceptual characteristic that distinguishes one musical instrument from another playing the same note with the same intensity and duration. Modeling timbre is very hard, and its complexity defies simple definition; it has been referred to as "the psychoacoustician's multidimensional waste-basket category for everything that cannot be labeled pitch or loudness"[1]. The timbre of a single note at a single pitch has a nonlinear dependence on the volume, time and even the particular way the instrument is played by the performer. While there is a substantial body of research in timbre modelling and synthesis (Chowning (1973); Risset and Wessel (1999); Smith (2010; 2011)), state-of-the-art musical sound libraries used by orchestral composers for analog instruments (e.g. the Vienna Symphonic Library (GmbH, 2018)) are still obtained by extremely careful audio sampling of real instrument recordings. Being able to model and manipulate timbre electronically carries importance for musicians who wish to experiment with different sounds, hear their music played in different (real or virtual) instruments and/or compose for multiple instruments.

In this paper, we consider the problem of high quality *timbre transfer* between audio clips obtained with different instruments. We take inspiration from recent successes in style transfer for images using neural networks (Gatys et al., 2015; Johnson et al., 2016; Ulyanov et al., 2016; Chu et al., 2017). An appealing strategy would be to directly apply image-based style transfer techniques to time-frequency representations of images, such as short-time Fourier transform (STFT) spectrograms. However, needing to convert the generated spectrogram into a waveform presents a fundamental obstacle, since accurate reconstruction requires phase information, which is difficult to predict (Engel et al., 2017), and existing techniques for inferring phase (e.g., Griffin and Lim (1984)) can produce characteristic artifacts, which is undesirable for high quality audio generation (Shen et al., 2017). See more details in Appendix D.

Recent years have seen rapid progress on generative models of audio that directly generate high-quality waveforms. WaveNet (van den Oord et al., 2016), SampleRNN (Mehri et al., 2016) and Tacotron2 (Shen et al., 2017) are examples of such generative networks, which were shown to successfully model human speech and perform end to end text to speech conversion. WaveNet's

[1] McAdams and Bregman (1979), pg 34

ability to condition on abstract audio representations is particularly relevant, since it enables one to perform manipulations in high-level auditory representations from which reconstruction would have previously been impractical. Tacotron 2 applied this strategy to obtain great success in the speech domain, where the audio is synthesized using WaveNet conditioned on a predicted Mel-Spectrogram representation. We adapt this general strategy to the music domain. We propose TimbreTron, a pipeline that performs CQT-based timbre transfer with high-quality waveform output on unpaired music data. For our representation, we choose the constant Q transform (CQT), a perceptually motivated representation of music (Brown, 1991). We show that this representation is particularly well-suited to musical timbre transfer and other manipulations due to its pitch equivariance and the way it simultaneously achieves high frequency resolution at low frequencies and high temporal resolution at high frequencies, a property that STFT lacks.

TimbreTron performs timbre transfer by three steps, shown in Figure 1. First, it computes the CQT spectrogram and treats its log-magnitude values as an image (discarding phase information). Second, it performs timbre transfer in the log-CQT domain using a variant of the CycleGAN (Zhu et al., 2017). Finally, it converts the generated log-CQT to a waveform using a conditional WaveNet synthesizer (which implicitly must infer the missing phase information).

Our TimbreTron can successfully perform musical timbre transfer on some instrument pairs empirically. The generated audio samples have realistic timbre that matches the target timbre while expressing the same overall musical content (e.g., rhythm, loudness, pitch). We qualitatively verified that the use of CQT representation is a crucial component in TimbreTron as it consistently yields better qualitative timbre transfer than its STFT counterpart.

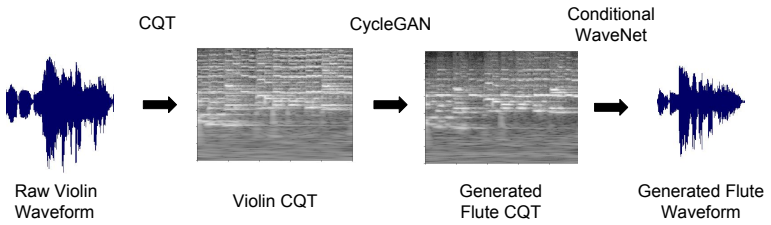

Figure 1: The TimbreTron pipeline that performs timbre transfer from Violin to Flute.

2 BACKGROUND

2.1 TIME-FREQUENCY ANALYSIS

Time-frequency analysis refers to techniques that aim to evaluate how the frequency domain representation signal changes over time.

Short Time Fourier Transform (STFT) is one of the most commonly applied techniques for this purpose. The discrete STFT operation can be compactly expressed as follows:

$$STFT\{x[n]\}(m, \omega_k) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega_k n}$$

The above formula computes the STFT of an input time-domain signal $x[n]$ at time step m and frequency ω_k. w refers to a zero-centered window function (such as Hann Window), which acts as a means of masking out the values that are away from m. Hence, the equation above can be interpreted as the discrete Fourier transform of the masked signal $x[n]w[n-m]$. An example spectrogram is shown in Figure 2.

Constant Q Transform (CQT) (Brown, 1991) is another time-frequency analysis technique in which the frequency values are geometrically spaced, with the following particular pattern (Blankertz): $\omega_k = 2^{\frac{k}{b}}\omega_0$. Here, $k \in \{1, 2, 3, ...k_{max}\}$ and b is a constant that determines the geometric separation between the different frequency bands. To make the filter for different frequencies adjacent to each other, the bandwidth of the kth filter is chosen as: $\Delta_k = \omega_{k+1} - \omega_k = \omega_k(2^{\frac{1}{b}-1})$ .This results in a

constant frequency to resolution ratio (as known as the "quality (Q) factor"):

$$Q = \frac{\omega_k}{\Delta_k} = (2^{\frac{1}{b}} - 1)^{-1}$$

Huzaifah (2017) systematically showed that CQT consistently outperform traditional representations such as Mel-frequency cepstral coefficients(MFCCs) in Environmental Sound Classification tasks using CNNs. **Rainbowgram** Engel et al. (2017) introduced the rainbowgram, a visualization of the CQT which uses color to encode time derivatives of phase; this highlights subtle timbral features which are invisible in a magnitude CQT. Examples of CQTs and rainbowgrams are shown in Figure 2.

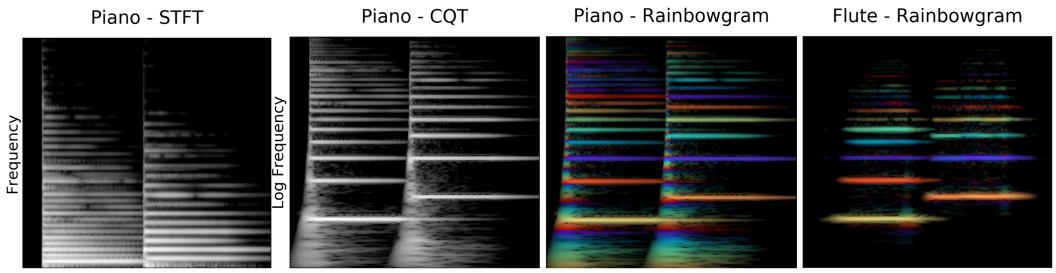

Figure 2: The STFT of a piano clip **(left)**, the CQT of the same piano clip **(second left)**, the rainbowgram of the same piano clip **(second right)** and the rainbowgram of a flute clip which has the same pitch as the first piano clip **(right)**. Note that the harmonics of different pitches are **approximate translations** of each other in the CQT representation.

2.2 WAVENET

WaveNet, proposed by van den Oord et al. (2016), is an auto-regressive generative model for generating raw audio waveform with high quality. The model consists of stacks of dilated causal convolution layers with residual and skip connections. With teacher forcing, WaveNet can be trained by processing many time steps convolutionally, removing the need for an expensive iteration over time steps as required by other methods (e.g., (Mehri et al., 2016)). WaveNet can be easily modified to perform conditional waveform generation; for example, it can be trained as a vocoder for synthesizing natural, high-quality human speech in TTS systems from low-level acoustic features (e.g., phoneme, fundamental frequency, and spectrogram) (Arik et al., 2017; Shen et al., 2017). One limitation of WaveNet is that the generation of waveforms can be expensive, which is not desirable for training procedures that require auto-regressive generation (e.g., GAN training, scheduled sampling).

2.3 GAN AND CYCLEGAN

Generative Adversarial Networks (GANs) are a class of implicit generative models introduced by Goodfellow et al. (2014). A GAN consists of a discriminator and a generator, which are trained adversarially via a two-player min-max game, where the discriminator attempts to distinguish real data from samples, and the generator attempts to fool the discriminator. The objective is:

$$G^*, D^* = \arg\min_G \max_D \mathbb{E}_{x\sim\mathcal{X}}[\log D(x)] + \mathbb{E}_{z\sim\mathcal{Z}}[\log(1 - D(G(z)))], \tag{1}$$

where D is the discriminator, G is the generator, z is the latent code vector sampled from Gaussian distribution $\mathcal{Z}$, and x is sampled from data distribution $\mathcal{X}$. GANs constituted a significant advance over previous generative models in terms of the quality of the generated samples.

CycleGAN (Zhu et al., 2017) is an architecture for unsupervised domain transfer: learning a mapping between two domains without any paired data. (Similar architectures were proposed independently by Yi et al. (2017); Liu et al. (2017); Kim et al. (2017).) The CycleGAN learns two generator mappings: $F : \mathcal{X} \to \mathcal{Y}$ and $G : \mathcal{Y} \to \mathcal{X}$; and two discriminators: $D_{\mathcal{X}} : \mathcal{X} \to [0, 1]$ and $D_{\mathcal{Y}} : \mathcal{Y} \to [0, 1]$. The loss function of CycleGAN consists of both adversarial losses (Eqn. 1), combined with a cycle consistency constraint which forces it to preserve the structure of the input:

$$\mathcal{L}_{\text{cyc}}(F, G, \mathcal{X}, \mathcal{Y}) = \mathbb{E}_{x\sim\mathcal{X}}[\|G(F(x)) - x\|_1] + \mathbb{E}_{y\sim\mathcal{Y}}[\|F(G(y)) - y\|_1] \tag{2}$$

3 MUSIC PROCESSING WITH CONSTANT-Q-TRANSFORM REPRESENTATION

This section focuses on the first and last steps of the TimbreTron pipeline: the steps related to the transforming raw waveforms to and from time frequency representations. We explain our reasoning for choosing the CQT representation and introduce our conditional WaveNet synthesizer which converts a (possibly generated) CQT to a high-quality audio waveform.

3.1 CQT FOR MUSIC REPRESENTATION

The CQT representation has desirable characteristics that make it especially suitable for processing musical audio signals. Unlike STFT, CQT has higher frequency resolution towards lower frequencies, which leads to better pitch specification for lower register instruments (such as cello or trombone), and higher time resolution towards higher frequencies, which makes it advantageous at handling rhythm at higher pitches. Also, Thanks to the geometric spacing of frequencies, a pitch shift corresponds (approximately) to a vertical translation of the "spectral signature" (unique pattern of harmonics) of musical instruments. This means that the convolution operation is (approximately) equivariant under pitch translation, which allows convolutional architectures to share structure between different pitches. (See Appendix B for more details.) One can exploit the geometric spacing of the frequencies to pick CQT parameters to exactly cover all the pitches present in the twelve tone, well-tempered scale, which makes it suitable for tasks related to music.

3.2 WAVEFORM RECONSTRUCTION FROM CQT REPRESENTATION USING CONDITIONAL WAVENET

Since empirical studies have shown it is difficult to directly predict phase in time-frequency representations (Engel et al., 2017), we discard the phase information and perform the image-based processing directly on a log-amplitude CQT representation. The details of our CQT operation can be found in Appendix C. Therefore, in order to recover a waveform consistent with the generated CQT, we need to infer the missing phase information, which is a difficult problem. (Velasco et al., 2011)

To convert log magnitude CQT spectrograms back to waveforms, we use a 40-layer conditional WaveNet with the dilation rate of $2^{k \pmod{10}}$ for the kth layer. The model is trained using pairs of a CQT and a waveform; this requires only a collection of unlabeled waveforms, since the CQT can be computed from the waveform.[2]

Beam Search Because the conditional WaveNet generates stochastically from its predictive distribution, it sometimes produces low-probability outputs, such as hallucinated notes. Also, because it has difficulty modeling the local loudness, the loudness often drifts significantly over the timescale of seconds. While these issues could potentially be addressed by improving the WaveNet architecture or training method, we instead take the perspective that the WaveNet's role is to produce a waveform which matches the target CQT. Since the above artifacts are macro-scale errors which happen only stochastically, the WaveNet has a significant probability of producing high-quality outputs over a short segment (e.g. hundreds of milliseconds). Therefore, we perform a beam search using the WaveNet's generations in order to better match the target CQT. See Appendix E.5 for more details about the beam search algorithm.

Reverse Generation In early experiments, we observed that attacks are sometimes hard to model during forward generation, resulting in multiple attacks or missing attacks. We believe this problem occurs because it is difficult to determine the onset of a note from a CQT spectrogram (in which information is blurred in frequency), and it is difficult to predict precise pitch at the note onset due to a broad frequency spectrum. We found that the problems of missing and doubled attacks could be mostly solved by having the WaveNet generate the waveform samples in reverse order, from end to beginning.

[2] We up-sample the CQT spectrograms to the rate of the audio using nearest neighbour interpolation before conditioning them to the WaveNet. The audio sample is quantized using 8-bit mu-law, and the output of the WaveNet is from softmax layer over 256 quantized values. At test time, we run the conditional WaveNet autoregressively with the initial condition of zero as the first sample value.

4 TIMBRE TRANSFER WITH CYCLEGAN ON CQT REPRESENTATION

In this section, we describe the middle step of our TimbreTron pipeline, which performs timbre transfer on log-amplitude CQT representations of the waveforms. As training data, we have collections of unrelated recordings of different musical instruments. Hence, our timbre transfer problem on log-amplitude CQT "images" is an instance of unsupervised "image-to-image" translation. To achieve this, we applied the CycleGAN architecture, but adapted it in several ways to make it more effective for time-frequency representations of audio.

Removing Checkerboard Artifacts The convnet-resnet-deconvnet based generators from the original CycleGAN led to significant checkerboard artifacts in the generated CQT, which corresponds to severe noise in the generated waveform. To alleviate this problem, we replaced the deconvolution operation with nearest neighbor interpolation followed with regular convolution, as recommended by Odena et al. (2016).

Full-Spectrogram Discriminator Due to the local nature of the original CycleGAN's transformations, Zhu et al. (2017) found it advantageous for the discriminator only to process a local patch of the image. However, when generating spectrograms, it's important that different partials of the same pitch be consistent with each other; a discriminator which is local in frequency cannot enforce this. Therefore, we gave the discriminator the full spectrogram as input.

Gradient Penalty Replacing the patch discriminator with the full-spectrogram one has led to unstable training dynamics because the discriminator was too powerful. To compensate for this, we added the Gradient Penalty(GP) (Gulrajani et al., 2017), to enforce a soft Lipschitz constraint:

$$\mathcal{L}_{\text{GP}}(G, D, \mathcal{Z}, \hat{\mathcal{X}}) = \alpha \cdot \mathbb{E}_{\hat{x}\sim\hat{\mathcal{X}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{3}$$

Here $\hat{\mathcal{X}}$ are samples taken along a line between the true data distribution $\mathcal{X}$ and the generator's data distribution $\mathcal{X}_g = \{F(z)|z \sim \mathcal{Z}\}$ via convex combination of a real data point and a generated data point. Fedus et al. (2018) showed empirically that GP can stable GAN training and in our work we verified that GP is necessary to stabilize the training dynamics of CycleGAN.

Identity loss In addition to the adversarial loss and the reconstruction loss that we applied to the generators, we also added identity loss, which was proposed by Zhu et al. (2017) to preserve color composition in the original CycleGAN. Empirically, we found out that the identity loss component helps generators to preserve music content, which yields better audio quality empirically.

$$\mathcal{L}_{\text{identity}}(F, G, \mathcal{X}, \mathcal{Y}) = \mathbb{E}_{x\sim\mathcal{X}}[\|F(x) - y\|_1] + \mathbb{E}_{y\sim\mathcal{Y}}[\|G(y) - x\|_1] \tag{4}$$

Our weighting of the identity loss followed a linear decay schedule (details in Appendix E.3). In this way, at the start of training, the generator is encouraged to learn a mapping that preserves pitch; as training progresses, the enforcement is reduced, allowing the generator to learn more expressive mappings.

5 RELATED WORK

There is a long history of using clever representations of images or audio signals in order to perform manipulations which are not straightforward on the raw signals. In a seminal work, Tenenbaum and Freeman (1999) used a multilinear representation to separate style and content of images. Ulyanov and Lebedev (2016) and Verma and Smith (2018) then applied the optimization technique proposed by Gatys et al. (2015) to the audio domain by applying the image-based architectures to spectrogram representations of the signals. Grinstein et al. (2017) took a similar approach, but used hand-crafted features to extract statistics from the spectrograms.

Zhu et al. (2017) introduced Cycle GAN approach to learn an "unsupervised image-to-image mapping" between two unpaired datasets using two generator networks and two discriminator networks with generative adversarial training. Given the success of the CycleGAN on image domain style transfer, Kaneko and Kameoka (2017) applied the same architecture to translate between human voices in the Mel-cepstral coefficient (MCEP) domain and Brunner et al. (2018) applied it to musical style transfer with MIDI representations.

What the aforementioned audio style transfer approaches have in common is that the reconstruction quality is limited by the existing non-parametric algorithms for audio reconstruction (e.g., the Griffin-Lim algorithm for STFT domain reconstruction (Griffin and Lim, 1984), or the WORLD vocoder for MCEP domain reconstruction of speech signals (Morise et al., 2016)), or existing MIDI synthesizer.

Another strategy is to operate directly on waveforms. van den Oord et al. (2016) demonstrated high-quality audio generation using WaveNet. Following on this, Engel et al. (2017) proposed a WaveNet-style autoencoder model operating on raw waveforms that was capable of creating new, realistic timbres by interpolating between already existing ones. Donahue et al. (2018) proposed a method to synthesize waveforms directly using GANs with improved quality over naive generative models such as SampleRNN (Mehri et al., 2016) and WaveNet. Mor et al. (2018) used an encoder-decoder approach for the Timbre Transfer problem, where they trained a shared encoder to learn the representation for raw waveform of various instruments, and then train instrument-specific decoder to reconstruct waveform from the learned representation.

6 EXPERIMENTS

We conducted two sets of experiments to 1) experiment with pitch-shifting and tempo-changing to further justify our choice of CQT representation; 2) test our full TimbreTron pipeline (along with ablation experiments to justify our architectural choices). See Appendix E for the details of our experiment settings. For this section, please listen to audio samples in the OnedDrive we provided as you read along: `1drv.ms/f/s!ApC93lRyk9iagZxyyx6EprcJGmyUTw`

6.1 DATASETS

We built our own MIDI dataset and real world music dataset for training the TimbreTron. Within each type of dataset, we divided the entire dataset into training set and test set. We also ensured that the training set and test set are entirely disjoint in terms of music content by splitting the dataset by music pieces. Training dataset consists of 4 second segments of audio, and the corresponding spectrogram represenations (CQT and STFT) for the audio segments. More details about our dataset is in Appendix E.1.

6.2 DISENTANGLING PITCH AND TEMPO USING CQT REPRESENTATION

Before presenting our timbre transfer results, we first consider the simpler problem of disentangling pitch and tempo in a musical recording. Recall that the two properties are entangled in the time domain representation, e.g. subsampling the waveform simultaneously increases the tempo and raises the pitch. Changing the two independently requires more sophisticated analysis of the signal. In the context of our TimbreTron pipeline, due to the CQT's pitch equivariance property, **pitch shifting** can be (approximately) performed simply by translating the CQT representation on the log-frequency axis. Any representation that does not have geometrically spaced sampled frequencies (such as STFT) does not lend itself easily to this type of simple transformation. We demonstrate that the aforementioned translation does indeed result in a perceivable pitch shift when fed to our conditional WaveNet. **Audio time stretching** can be simply performed using either the CQT or STFT representations, combined with the WaveNet synthesizer, by changing the number of waveform samples generated per CQT window. Regardless of the number of samples generated, the WaveNet synthesizer is able to produce the correct pitch based on the local frequency content. In conclusion, our method was able to vary the pitch and tempo independently while otherwise preserving the timbre and musical structure.

6.3 TIMBRE TRANSFER EXPERIMENTS

We started out with MIDI data because it is possible to produce paired test dataset for evaluation. After moving on to real world data, we noticed that real world data is harder to learn because compared to MIDI data it's more irregular and more noisy, thus makes it a more challenging task. In this section, we show our experimental findings on the full TimbreTron pipeline using real world data, verify the correctness of our reasoning about CQT, and show the generalization capability of TimbreTron.

CQT TimbreTron vs. STFT TimbreTron We found that STFT TimbreTron has two problems: 1) it sometimes fails to learn to translate low pitches, likely due to its poor frequency resolution at low frequencies, and 2) it sometimes fails to learn to preserve pitch, but learn a random pitch permutation. For example, we ran TimbreTron on a Bach piano sample played by a professional musician. The STFT TimbreTron transposed parts of the longer excerpt by different amounts, and for a few notes in particular, seemed to fail to transpose them by the same amount as it did the others. Those problems were completely solved using CQT TimbreTron. Furthermore, we confirmed that those two kinds of artifacts are introduced in the CycleGAN, because both STFT Griffin Lim and STFT Vocoder will produce exact the same pitch permutation, which rules out the possibility of vocoder being the problem. This empirically verified that, compared to STFT representation, CQT is advantageous for convolutional architectures because CQT is equivalent to pitch and can achieve high frequency resolution at low frequencies, as was mentioned in section 3.1.

Generalization Capability of TimbreTron To further explore the generalization capability of TimbreTron, we also tried one domain adaptation experiment where we take CycleGAN trained on MIDI data, test it on Real World test dataset, and synthesize audio with Wavenet trained on training real world data. And the result is just as good as previous experiments. This suggested that our model has the ability to generalize to unseen real world data, even though it's only trained on MIDI dataset.

6.4 Evaluation with Amazon Mechanical Turk

Comparing CQT vs. STFT To empirically test if our proposed TimbreTron with CQT representation is better than its STFT-Wavenet counterpart, or its STFT-GriffinLim counterpart, we conducted AMT human study. In the questionnaire, we asked Turkers to listen to three audio clips: the original audio from instrument A (the "instrument example"), the TimbreTron generated audio of instrument A, and its STFT conterparts, then asked them: "In your opinion, which one of A and B sounds more like the instrument provided in "instrument example"? , where A and B in the questions are the generated samples (presented in random order). Naturally, sounding closer to the "instrument sample" means the timbre quality is better. We conducted two groups of experiment. In the first group, the stft counterpart is the Wavenet and CycleGAN trained on STFT representation and the result is in first row of the Table 1: most people think the CQT TimbreTron is better. In the second group, we took the same CycleGAN trained on STFT, but instead simply generate the waveform using Griffin Lim algorithm. The results are in the second row: Even more people think CQT TimbreTron is better. In conclusion, compared to Griffin Lim as the baseline, training a Wavenet on STFT improved Timbre quality marginally. Furthermore, samples generated by TimbreTron trained on CQT was proven to have significantly better timbre quality.

Does TimbreTron transfer Timbre? To be effective, the system must transform a given audio input so that the output is (1) recognizable as the same (or appropriately similar) basic musical piece, and (2) recognizable as the target instrument. We address both of these criteria below

(1) Preserving the musical piece. A different instrument playing the same notes may not always sound subjectively like the same "piece". When this is done in musical contexts, the notes themselves are often changed in order to adapt pieces between instruments, and this is generally referred to as a new "arrangement" of an existing piece. Thus, even in the cases where we had a recording available in the target domain, the exact notes or timings were not always identical to those in the original recording from which we transferred. Overall, when we did have such a target domain recording of a real instrument, we found that for the pair of (Real Target Instrument, TimbreTron Generated Target Instrument), 88% of respondents considered the musical pieces to be nearly identical or very similar, while roughly 10.5% considered them related and 1.5% considered them different. Thus, it appears that generally the musical piece was indeed preserved. Details are provided in Appendix F.

(2) Transferring the timbre. Evaluating this is challenging because, if the transfer is not perfect (which it is not), then judging similarity of not-quite-identical instruments is fraught with perceptual challenges. With this in mind, we included a range of pairwise comparisons and gave a likert scale with various anchors (all details provided in Appendix F). Overall, we found that for the pair (Ground Truth Target audio, TimbreTron Generated audio), roughly 85% of respondents considered the instrument generating the audio to be very similar (e.g. still piano, but a different piano) or similar (e.g. another string instrument). We also asked participants to identify the instrument that they heard

Percentage / Answer / Architecture	CQT	same	STFT
STFT+WaveNet counterpart	46.67%	25.83%	27.5%
STFT+Griffinlim counterpart	56.67%	27.5%	15.83%

Table 1: Table for AMT results on timbre quality comparisons between our proposed TimbreTron, TimbreTron but with STFT Wavenet and TimbreTron with STFT Griffin Lim

in some of the audio excerpts, with an open-ended question. Generally we found that participants were indeed able to either identify the correct instrument, or confused with a very similar-sounding instrument. For example, one participant described a generated harpsichord as a banjo, which is in fact very close to harpsichord in terms of timbre. As a reference, participants had similar reasonable confusions about identifying ground truth instruments as well (e.g., one participant described a real harpsichord as being a sitar). See Appendix F for details. Based on perceptual evaluations above, we claim that TimbreTron is able to transfer timbre recognizably while preserving the musical content.

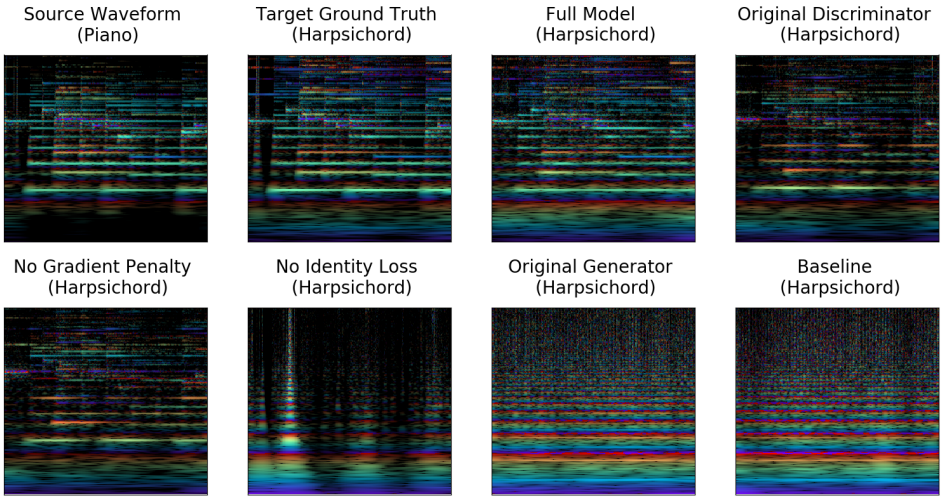

Figure 3: Rainbowgrams of the 4-second audio samples for the ablation study on MIDI test dataset. The source ground truth and the target ground truth come from a paired samples in the dataset. All other audio samples are the timbre transfer results from the source ground truth with different versions (full and ablated) of our TimbreTron. "Full Model" corresponds to the output of our final TimbreTron, which is perceptually closest to target ground truth and have the best audio quality. "Original discriminator" or "Original generator" corresponds to the TimbreTron pipeline with the discriminator or generator replaced by the original discriminator or generator in the original CycleGAN. "No gradient penalty", "No identity loss", and "No data augmentation" are referring to the full model without the corresponding modifications. "Baseline" is the original CycleGAN (Chu et al., 2017)

6.5 Ablation Study for TimbreTron

To better understand and justify each modification we made to the original CycleGAN, we conducted ablation study where we take away one modification at a time for MIDI CQT experiment. We used MIDI data for ablation because the dataset has paired samples, which provides a convenient ground truth for transfer quality evaluation. Figure 3 demonstrates the necessity of each modification for the success of our model.

7 Conclusion

We presented the TimbreTron, a pipeline for perfoming high-quality timbre transfer on musical waveforms using CQT-domain style transfer. The entire pipeline can be trained on unrelated real-world music segments, and intriguingly, the MIDI-trained CycleGAN demonstrated generalization

to real-world musical signals. We believe this work constitutes a proof-of-concept for CQT-domain manipulation of music signals with high-quality waveform outputs.

REFERENCES

Jont B Allen and Lawrence R Rabiner. A unified approach to short-time fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11):1558–1564, 1977.

Sercan O Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Jonathan Raiman, Shubho Sengupta, et al. Deep voice: Real-time neural text-to-speech. *arXiv preprint arXiv:1702.07825*, 2017.

Benjamin Blankertz. The constant q transform. URL `http://doc.ml.tu-berlin.de/bbci/material/publications/Bla_constQ.pdf`.

Judith C Brown. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

Gino Brunner, Yuyi Wang, Roger Wattenhofer, and Sumu Zhao. Symbolic music genre transfer with cyclegan. *arXiv preprint arXiv:1809.07575*, 2018.

J. M. Chowning. The synthesis of complex audio spectra by means of frequency modulation. *Journal of the Audio Engineering Society*, 21(7):526–534, 1973.

Casey Chu, Andrey Zhmoginov, and Mark Sandler. Cyclegan, a master of steganography. *CoRR*, abs/1712.02950, 2017. URL `http://arxiv.org/abs/1712.02950`.

Chris Donahue, Julian McAuley, and Miller Puckette. Synthesizing audio with generative adversarial networks. *CoRR*, abs/1802.04208, 2018. URL `http://arxiv.org/abs/1802.04208`.

Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi. Neural audio synthesis of musical notes with wavenet autoencoders. *CoRR*, abs/1704.01279, 2017. URL `http://arxiv.org/abs/1704.01279`.

William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. 2018.

Derry Fitzgerald, Matt Cranitch, and Marcin T Cychowski. Towards an inverse constant q transform. In *Conference papers*, page 12, 2006.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. URL `http://arxiv.org/abs/1508.06576`.

Vienna Symphonic Library GmbH, 2018. URL `https://www.vsl.co.at/en/Products`.

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

Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

Eric Grinstein, Ngoc Q. K. Duong, Alexey Ozerov, and Patrick Pérez. Audio style transfer. *CoRR*, abs/1710.11385, 2017. URL `http://arxiv.org/abs/1710.11385`.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

Jeffrey Hass. Chapter one: An acoustics primer, 2018. URL `http://www.indiana.edu/~emusic/etext/acoustics/chapter1_loudness.shtml`.

Nicki Holighaus, Monika Dörfler, Gino Angelo Velasco, and Thomas Grill. A framework for invertible, real-time constant-q transforms. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(4):775–785, 2013.

Muhammad Huzaifah. Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. *arXiv preprint arXiv:1706.07156*, 2017.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

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

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

librosa. librosa. https://librosa.github.io.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017.

Stephen McAdams and Albert Bregman. Hearing musical streams. *Computer Music Journal*, 3(4): 26–43+60, December 1979.

Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron C. Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. *CoRR*, abs/1612.07837, 2016. URL `http://arxiv.org/abs/1612.07837`.

Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman. A universal music translation network. *arXiv preprint arXiv:1805.07848*, 2018.

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*, 99(7):1877–1884, 2016.

Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. doi: 10.23915/distill.00003. URL `http://distill.pub/2016/deconv-checkerboard`.

Jean-Claude Risset and David Wessel. Exploration of timbre by analysis and synthesis. In Diana Deutch, editor, *The Psychology of Music*, pages 113–169. Elsevier, 2 edition, 1999.

Juan G Roederer. *The physics and psychophysics of music: an introduction*. Springer Science & Business Media, 2008.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *arXiv preprint arXiv:1712.05884*, 2017.

Julius O. III Smith. *Physical Audio Signal Processing: for Virtual Musical Instruments and Audio Effects*. W3K Publishing, 2010.

Julius O. III Smith. *Spectral Audio Signal Processing*. W3K Publishing, 2011.

Nicolas Sturmel and Laurent Daudet. Signal reconstruction from stft magnitude: A state of the art.

Joshua B Tenenbaum and William T Freeman. Separating style and content with bilinear models. 1999.

Dmitry Ulyanov and Vadim Lebedev. Audio texture synthesis and style transfer. 2016. URL `https://dmitryulyanov.github.io/audio-texture-synthesis-and-style-transfer/`.

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *CoRR*, abs/1603.03417, 2016. URL http://arxiv.org/abs/1603.03417.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016. URL http://arxiv.org/abs/1609.03499.

Gino Angelo Velasco, Nicki Holighaus, Monika Dörfler, and Thomas Grill. Constructing an invertible constant-q transform with non-stationary gabor frames. *Proceedings of DAFX11, Paris*, pages 93–99, 2011.

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

Zili Yi, Hao Zhang, Ping Tan Gong, et al. Dualgan: Unsupervised dual learning for image-to-image translation. *arXiv preprint arXiv:1704.02510*, 2017.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017. URL http://arxiv.org/abs/1703.10593.

A Components of a Musical Tone

In this section, we will briefly describe the main components of a musical tone: pitch, loudness and timbre (Roederer, 2008).

Pitch is described subjectively as the "height" of a musical tone, and is closely tied to the fundamental mode of oscillation of the instrument that is producing the tone. This oscillation mode is often called the *fundamental frequency*, and can often be observed as the lowest band in spectrogram visualizations (Figure 4).

Loudness is linked to the perception of sound pressure, and is often subjectively described as the "intensity" of the tone. It roughly correlates with the amplitude of the waveform of the perceived tone, and has a weak dependence to pitch (Hass, 2018).

Timbre is the perceptual quality of a musical tone that enables us to distinguish between different instruments and sound sources with the same pitch and loudness (Roederer, 2008). The physical characteristics that define the timbre of a tone are its *energy spectrum* (the magnitude of the corresponding spectrogram) and its *envelope*.

Since sounds generated by physical instruments mostly rely on oscillations of physical material, the energy spectra of instruments consist of bands, which correspond to (approximately) the integer multiples of the fundamental frequency. These multiples are called *harmonics*, or *overtones*, and can be observed in Figure 4. The timbre of an instrument is tightly related to the relative strengths of the harmonics. The spectral signature of an instrument not only depends on the pitch of the tone played, but also changes over time. To see this clearly, consider that a single piano note of duration 500 milliseconds is played in reverse - the resultant sound will not be recognizable as a piano, although it will have the same spectral energy. The *envelope* of a tone corresponds to how the instantaneous amplitude changes over time, and is mainly affected by the instrument's attack time (the transient "noise" created by the instrument when it is first played), decay/sustain (how the amplitude decreases over time, or can be sustained by the player of the instrument) and release (the very end of the tone, following the time the player "releases" the note). All these factors add to the complexity and richness of an instrument's sound, while also making it difficult to model it explicitly.

B Equivariance of Convolution over Log-Frequency Axis with respect to Translation

Since nearby pitches played by the same instrument are approximately translations of each other on the log-frequency axis, convolution operation is equivariant under pitch shift on any log-frequency

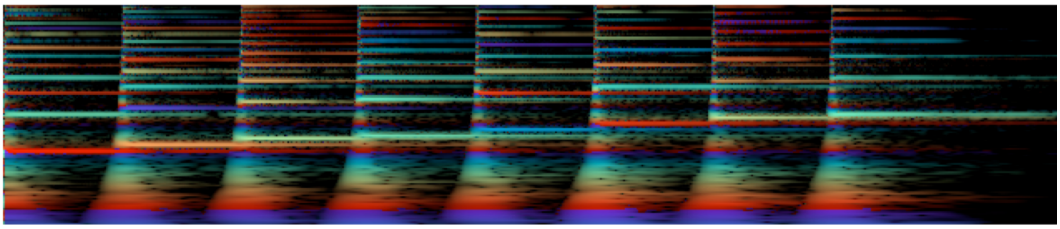

Figure 4: The rainbowgram of a C major scale played by piano.

representation. A demonstration of this can be seen in Figure 4. Since the harmonics of a musical instrument are approximately integer multiples of the fundamental frequency, scaling the fundamental frequency (hence the pitch) corresponds to a constant shift in all of the harmonics in log scale.

We also want to emphasize on some of the reasons why the equivariance is only approximate:

- **Imperfect multiples:** In real audio samples from instruments, the harmonics are only approximately integer multiples of the fundamental frequency, due to the material properties of the instruments producing the sound.
- **Dependence of spectral signature on pitch:** For each pitch, each instrument has a slightly different spectral signature, meaning that a simple translation in the frequency axis cannot completely account for the changes in the frequency spectrum.

C Spectrogram Processing Details

Waveform to CQT Spectrogram Using constant-Q transform as described in Section 2.1, CQT spectrogram can be easily computed from time-domain waveforms. In this work, we use a 16 ms frame hop (256 time steps under 16kHz), $\omega_0 = 32.70$ Hz (the frequency of C1 [3]), $b = 48$, $k_{max} = 336$ for the CQT transform. Standard implementations of CQT (e.g., librosa (librosa)) also allow scaling the Q values by a constant $\gamma > 0$ to have finer control over time resolution - choosing $\gamma \in (0, 1)$ results in increased time resolution. In our experiments, we choose $\gamma = 0.8$. After the transformation, we take the log magnitude of the CQT spectrogram as the spectrogram representation.

Waveform to STFT Spectrogram All the STFT spectrograms are generated using STFT with $k_{max} = 337$. The window function is picked to be Hann Window with a window length of 672. A 16 ms frame hop is also used (256 time steps under 16kHz). Similar to CQT spectrogram, we also take the log magnitude of the STFT spectrogram as the spectrogram representation after the STFT.

D Waveform Reconstruction from Spectrogram Representation Background

Synthesis of the aforementioned time-frequency analysis techniques in Section 2.1 can be performed in the presence of both magnitude and phase information (Allen and Rabiner, 1977) (Holighaus et al., 2013). In the absence of phase information, one of the common methods of synthetically generating phase from STFT magnitude is the Griffin-Lim algorithm (Griffin and Lim, 1984). This algorithm works by randomly guessing the phase values, and iteratively refining them by performing STFT and inverse STFT operations until convergence, while keeping the magnitude values constant throughout the process. Developed to minimize the mean squared error between the target spectrogram and predicted spectrogram, this algorithm is shown to reduce the objective function at each iteration, while having no optimality guarantees due to the non-convexity of the optimization problem(Griffin and Lim, 1984) (Sturmel and Daudet). Although recent developments in the field have enabled performing the inverse operation of CQT (Velasco et al., 2011) (Fitzgerald et al., 2006), these techniques still require both phase and magnitude information.

[3] C1 refers to the "C1" key, corresponding to the lowest "C" on the piano keyboard.

E DETAILED EXPERIMENTAL SETTINGS

E.1 DATASETS

MIDI Dataset Our MIDI dataset consists of two parts: MIDI-BACH [4] and MIDI-Chopin [5]. MIDI-BACH dataset is synthesized from a collection of bach MIDI files which have a total duration of around 10 hours [6]. Each dataset contains 6 instruments: acoustic grand, violin, electric guitar, flute, and harpsichord. We generated the audio with the same melody but different timbre, which makes it possible to obtain paired data during evaluation.

Real World Dataset Our Real World Dataset comprises of data collected from YouTube videos of people performing solo on different instruments including piano, harpsichord, violin and flute. Each instrument contains around 3 to 10 hours of recording.

E.2 DOMAIN SPECIFIC GLOBAL NORMALIZATION

As is shown in Figure 5 in Appendix E.7, the distribution of spectrogram pixel magnitude is roughly centered at -2, which is not good for learning because of the tanh activation function works better when the activation is in the range of $[-1, 1]$. Thus, we globally normalized the spectrogram data to be mostly in the range of $[-1, 1]$ for each instrument domain. We scaled and shifted the spectrograms based on the mean and standard deviation of each instrument domain to achieve Domain Specific Global Normalization in the input pipeline, and reverse this operation on the output of CycleGAN to minimize possible distribution shift before feeding the output for wavenet generation.

E.3 CYCLEGAN TRAINING DETAILS

In CycleGAN training, because we made several architectural changes, we re-tuned the hyperparameters. The weighting for our cycle consistency loss is 10 and the weighting of the identity loss is 5. In the original CycleGAN the weighting of identity loss is constant throughout training but in our experiment, it stays constant for the first 100000 steps, then it starts linearly decay to 0. We set the weighing for Gradient Penalty to be 10, as was suggested in Gulrajani et al. (2017). Our learning rate is exponentially warmed up to 1e 4 over 2500 steps, stays constant, then at step 100000 starts to linearly decay to zero. The total training step is 1.5 million steps, trained with Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0$ and $\beta_2 = 0.9$, with a batch size of 1.

E.4 CONDITIONAL WAVENET TRAINING

The conditional wavenet is trained with a learning rate of 0.0001 using Adam optimizer (Kingma and Ba, 2014), batch size of 4, sample length of 8196 ($\approx 0.5s$ for audio with 16000Hz sampling rate). To improve the generation quality we maintain an exponential moving average of the weights of the network with a decaying factor of 0.999. The averaged weights are then used to perform the autoregressive generation. To make the model more robust, we augmented the training dataset by randomly rescaling the original waveform based on its peak value based on a uniform distribution $uniform(0.1, 1.0)$.

E.5 BEAM SEARCH

During autoregressive generation, we perform a modified beam search where the global objective is to minimize the discrepancy between the target CQT spectrogram and the CQT spectrogram of the synthesized audio waveform. Our beam search alternates between two steps: 1) run the autoregressive WaveNet on each existing candidate waveforms for n steps ($n = 2048$) to extend the candidate waveforms, 2) prune the waveforms that have large squared error between the waveforms' CQT spectrogram and the target CQT spectrogram (beam search heuristic). We maintain a constant number of candidates (beam width = 8) by replicating the remaining candidate waveforms after each

[4] from website `www.jsbach.net/midi/`

[5] from website `www.piano-midi.de/chopin.htm`

[6] For all the synthesized audio, we use Timidity++ synthesizer

pruning process. To make sure the local beam search heuristic is approximately aligned with the global objective, we take n extra prediction steps forward and use the extra n samples along with the candidate waveforms to obtain a better prediction of the spectrogram for the candidate waveforms. The algorithm is provided in details as follows given the target spectrogram C_{target}:

1. $k \leftarrow 0$
2. Perform $2n$ autoregressive synthesis step on WaveNet on $\{x_1, \cdots, x_k\}$ with m parallel probes (m is the beam width) to produce m subsequent waveforms: $\{x^{(1)}_{k+1}, \cdots, x^{(1)}_{k+2n}\}, \{x^{(2)}_{k+1}, \cdots, x^{(2)}_{k+2n}\}, \cdots, \{x^{(m)}_{k+1}, \cdots, x^{(m)}_{k+2n}\}$.
3. Compute the CQT spectrogram C_i of $\{x^{(i)}_{k+1}, \cdots, x^{(i)}_{k+2n}\}$ for each $i \in \{1, 2, \cdots, m\}$, and find the waveform $\{x^{(i')}_{k+1}, \cdots, x^{(i')}_{k+2n}\}$ with the lowest square difference between C_i and the target CQT spectrogram C_{target}
4. Update the waveform $x_j = x^{i'}_j, \forall j \in \{k+1, k+2, \cdots, k+n\}$
5. $k \leftarrow k + n$

E.6 TIMBRETRON AT TEST TIME

One-shot generation In our earlier attempts, we tried generating 4 seconds segments and then merge them back. However, this resulted in volume inconsistencies between the 4 second generations. We suspect the CycleGAN learned a random volume permutation, because essentially there's no explicit gradient signal against it from the discriminator, after we enabled volume augmentation during train time. To resolve this issue, we removed the size constraint in our generator during test time so that it can generate based on input of arbitrary length. At test time, the dataset is no longer 4 second chunks, instead, we preserved the original length of the musical piece(except when the piece is too long we cut it down to 2 minutes due to GPU memory constraint). During test time generation, the entire piece is fed into the CycleGAN generator in one shot.

E.7 SPECTROGRAM RAW PIXEL INTENSITY HISTOGRAM

Figure 5 shows that the rough distribution of spectrograms are centered at -2. As is discussed in Section 3.1, we globally normalized our input data based oh the distribution of spectrograms for each domain of instruments.

F AMAZON MECHANICAL TURK (AMT) RESULTS: DOES TIMBRETRON TRANSFER TIMBRE?

We conducted experiments to investigate the question "Does TimbreTron transfer Timbre?". The experiment results are shown in this section.

The comparison-based experiments include two categories: instrument similarity and musical piece similarity, and each is done in two settings: with beam-search and without beam-search. Table 2 and 3 contain results for the instrument similarity comparison, where beam search is deployed in the former and not in the latter. Likewise, Table 4 and 5 contain results for the music piece similarity comparison.

The Turkers are also asked to provide their subjective judgment about the instrument used for the provided samples. A sample of the answers are shown in Table 6.

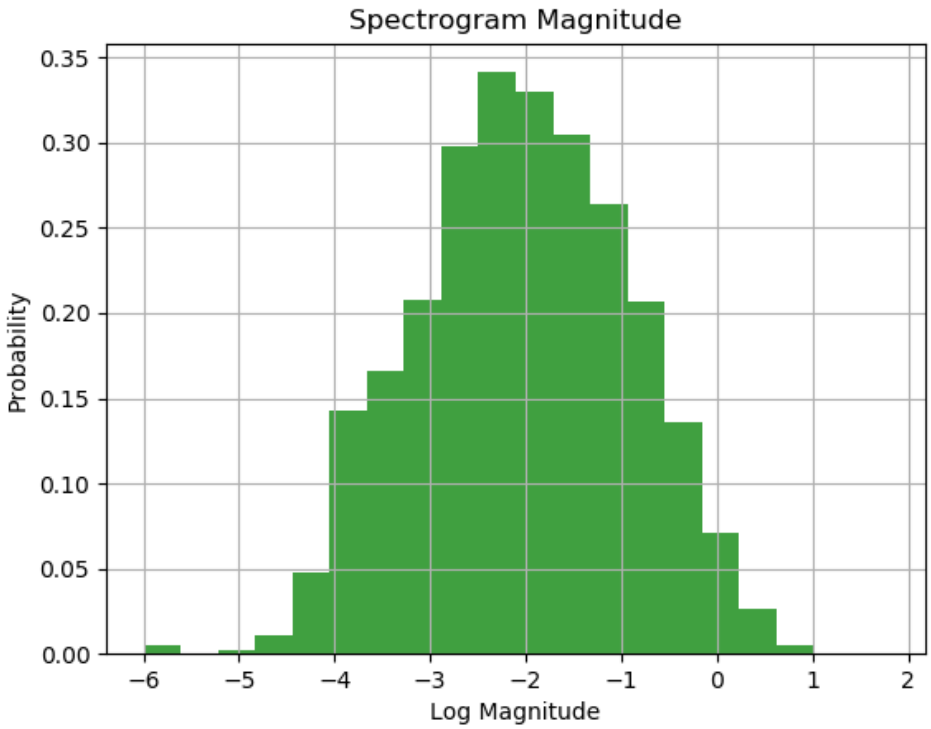

Figure 5: Spectrogram raw pixel intensity histogram

Percentage Answer / Architecture	Very Similar	Similar	Different	Do not know
Ground Truth Target Instrument & TimbreTron Generation	34.75%	39.83%	23.73%	1.69%
Ground Truth Original Instrument & TimbreTron Generation	20.0%	16.67%	61.67%	1.67%

Table 2: Table for AMT results on pair-wise instrument comparisons between our proposed TimbreTron with beam search, ground truth original instrument and ground truth target instrument

Percentage Answer / Architecture	Very Similar	Similar	Different	Do not know
Ground Truth Target Instrument & TimbreTron Generation	50.0%	35.48%	14.52%	0.0%
Ground Truth Original Instrument & TimbreTron Generation	27.42%	19.35%	53.23%	0.0%

Table 3: Table for AMT results on pair-wise instrument comparisons between our proposed TimbreTron without beam search, ground truth original instrument and ground truth target instrument

Percentage / Answer / Architecture	Nearly Identical	Very Similar	Related	Entirely Different	Do not know
Ground Truth Target Instrument & TimbreTron Generation	50.0%	32.2%	11.86%	5.93%	0.0%
Ground Truth Original Instrument & TimbreTron Generation	43.33%	28.33%	11.67%	16.67%	0.0%

Table 4: Table for AMT results on pair-wise musical piece comparisons between our proposed TimbreTron with beam search, ground truth original instrument and ground truth target instrument

Percentage / Answer / Architecture	Nearly Identical	Very Similar	Related	Entirely Different	Do not know
Ground Truth Target Instrument & TimbreTron Generation	42.74%	45.16%	10.48%	1.61%	0.0%
Ground Truth Original Instrument & TimbreTron Generation	33.87%	30.65%	19.35%	16.13%	0.0%

Table 5: Table for AMT results on pair-wise musical piece comparisons between our proposed TimbreTron without beam search, ground truth original instrument and ground truth target instrument

Harpsichord	Violin
Harpsichord	Violin
Piano	Cello
Banjo	Orchestra
Xylophone	Flute
Guitar	Organ
Sitar	Trumpet

Table 6: Example answers of the open-ended question that asks for what instrument a sample sounds like. The header indicates the correct answers (harpsichord and violin). Not all responses are listed here, the table content shows typical ones that make up for the majority of the answers.