Spec2Spec: Towards the general framework of music processing using generative adversarial networks

Hyeong-Seok Choi*, Juheon Lee† and Kyogu Lee‡

Music and Audio Research Group, Seoul National University,
1 Gwanak-ro, Gwanak-gu, Seoul, Republic of Korea 08826

Abstract: The advent of deep learning has led to a great progress in solving many problems that had been considered challenging. Several recent studies have shown promising results in directly changing the styles between two different domains that share the same latent content, for example, from paintings to photographs and from simulated roads to real roads. One of the key ideas that lie in this series of domain translation approaches is the concept of generative adversarial networks (GANs). Motivated by this concept of changing a certain style of data into another style using GANs, we apply this technique to two challenging and yet very important applications in the music signal processing field: music source separation and automatic music transcription. Both tasks can be interpreted as a style transition between two different spectrogram domains that share the same content; i.e., from a mixture spectrogram to a specific source spectrogram in the case of source separation, and from an audio spectrogram to a piano roll representation in the case of music transcription. Through experiments using real-world audio, we demonstrate that one general deep learning framework, namely “spectrogram to spectrogram” or “Spec2Spec,” can successfully be applied to tackle these problems.

Keywords: Music signal processing, Spectrogram-to-spectrogram translation, Source separation, Polyphonic music transcription

PACS number: 43.58.Ta [doi:10.1250/ast.41.160]

1. INTRODUCTION

The advent of the deep learning has made a huge leap in many problems that have been considered challenging. Yet, some problems — converting highly complex data into other complex data, for instance — still remain a challenging task. However, the possibility of a deep-neural-network-based methodology has still not been fully exploited in many other problems and it was not until some studies such as [1] that some breakthroughs were made in a style transfer task between two different images. Also, some recent notable approaches including those in [2,3], successfully enabled the direct changing of the styles between two different domains that share the same latent contents — for example, from paintings to photographs and from simulated roads to real roads. One of the key ideas that lie in this series of domain translation studies is the concept of generative adversarial networks (GANs) [4]. The characteristic of GANs that the distance between the data distribution and the implicit generative distribution is reduced has also been successfully applied to, for example, speech enhancement [5] and singing voice separation [6] in the field of audio signal processing.

Motivated by this concept of changing a certain style of data into another style using GANs, we apply this technique to two challenging and yet very important applications in the music signal processing field, that is, music source separation and automatic music transcription. Music source separation is a task that can be considered as a style transition from a mixture spectrogram to a specific source spectrogram (e.g., singing voice or accompaniment), where the source is shared content between the two different, yet correlated, domains. Also, in the automatic music transcription task, a music score is shared content between a musical audio and a pitchwise representation such as a piano roll, and thus it can also be seen as a style transition task from an audio spectrogram to a piano roll representation. Therefore, because both tasks can be interpreted as a style transition between two different spectrogram domains sharing the same content, we believe that it should be possible to tackle these problems using one general deep learning framework, namely, “spectro-
gram to spectrogram' or 'Spec2Spec,' as illustrated in Fig. 1.

2. BACKGROUND

Music information retrieval (MIR) is a field of study that aims to extract information from music, and a number of diverse approaches have been proposed for the various topics studied in MIR. However, many subjects in MIR have the commonality that a music signal is transformed into a time-frequency representation, modified or processed, and inverse-transformed it to an audio signal. In such tasks, it would be very useful to have a model that can be universally applied to various tasks that aim to convert one spectrogram to another.

In this paper, we propose a general framework that can be applied to the two very challenging tasks, music source separation and automatic music transcription (AMT), which are regarded as two holy grails in the MIR field [7]. We show that the proposed deep learning model based on GANs is generally applicable to both tasks, which we analogously formulate as style transfer tasks in the audio domain. Therefore, our approach has a high significance as follows: first, it is the first case of tackling two challenging MIR tasks using one general framework as far as we know; secondly, we propose a new approach using generative models for both tasks; thirdly, the proposed model has a high potential to be extended to other tasks in music signal processing.

3. SPEC2SPEC

In this section, we first briefly summarize the background regarding GANs to explain the key ideas of our framework. Then, we explain our objective function to train our deep learning model structure. Next, we explain our deep learning model structure for the generator and discriminator. Lastly, using our model, we introduce two possible applications: music source separation and automatic polyphonic piano music transcription. An overview of our system is depicted in Fig. 2.

3.1. Related Work

GANs make up a generative model that aimed at creating a mapping function generator \( G \) that is able to map a noise sample \( z \) to the data space, where \( z \) is usually sampled from a probability distribution \( p(z) \), for instance, a Gaussian distribution or uniform distribution.

Theoretically, \( G \) is trained to reduce the Jensen-Shannon divergence (JSD) between \( P_r \) and \( P_g \), assuming that function \( D \) reaches its optimal state in every iteration [4]. However, it often shows an unstable training progress, and Arjovsky and Bottou [8] argued that there is an
underlying problem with the nature of JSD and thus, alternative metrics must be suggested. Numerous other metrics to deal with this problem have been suggested in many types of GANs, and one prominent approach that shows both promising theoretical and empirical results is Wasserstein GANs [9]. It suggests that a soft metric that can approximate the Wasserstein distance between the data distribution ($P_{\text{data}}$) and the generator distribution ($P_{\hat{g}}$). Using the Wasserstein GANs, the objective can be reformulated as Eq. (2). Note that, the condition. Now, using the regularization term, the previous condition.

$$L = \min_{D} \max_{\hat{g}} \mathbb{E}_{x \sim P_{\text{data}}} [D(x)] - \mathbb{E}_{\hat{x} \sim P_{\hat{g}}} [D(\hat{x})] + \lambda_{g} \cdot \mathbb{E}_{x \sim P_{\hat{g}}} [\|\nabla_{\hat{x}} D(\hat{x})\|^2 - 1]^2$$  

(2)

### 3.2. Objective

In many of the previous studies [2,5,6] GANs were used to change image styles from one domain to the other by the approach of conditional GANs [11]. To reformulate the aforementioned objective 2. (2) using the conditional GANs setting, we first must define which variable has to be given as a condition. Naturally, we can consider the input spectrogram $x_{\text{input}}$ as a condition and the generator as a neural sampler $G(x_{\text{input}}; \theta_{g})$ that can estimate a desired output. In this setting, the discriminator examines neither the ground truth data ($x_{\text{gt}}$) nor the output from the generator ($x_{\text{est}}$) alone but the concatenation of either ($x_{\text{input}}, x_{\text{gt}}$) or ($x_{\text{input}}, x_{\text{est}}$). For the gradient penalty term, we uniformly sampled $\hat{x} \sim P_{\hat{g}}$ from the straight line between the concatenation of ($x_{\text{input}}, x_{\text{gt}}$) or ($x_{\text{input}}, x_{\text{est}}$).

In addition to the generative adversarial loss, we also calculate the more conventional pixel wise $l_1$ or $l_2$ loss between $x_{\text{est}}$ and $x_{\text{gt}}$ in the time-frequency domain and add both the generative adversarial loss and the conventional loss to the objective function.

In summary, the final objective for the generator ($L_{G}$) and the discriminator ($L_{D}$) becomes as follows. $\lambda_{G}$ and $\lambda_{l}$ are denoted as hyperparameters between generative adversarial loss and conventional loss to adjust the ratio between the two loss terms.

$$L_{G} = -\lambda_{D} \cdot \mathbb{E}_{x_{\text{input}}, x_{\text{gt}} \sim P_{\text{data}}, x_{\text{est}} \sim P_{g}} [D(x_{\text{input}}, x_{\text{est}})] + \lambda_{l} \cdot \mathbb{E}_{x \sim P_{\text{data}}, x_{\text{est}} \sim P_{g}} [\|x_{\text{gt}} - x_{\text{est}}\|_1 + 2]$$  

(3)

$$L_{D} = \lambda_{D} \cdot (\mathbb{E}_{x_{\text{input}}, x_{\text{gt}} \sim P_{\text{data}}, x_{\text{est}} \sim P_{g}} [D(x_{\text{input}}, x_{\text{est}})]) - \mathbb{E}_{(x_{\text{input}}, x_{\text{gt}}) \sim P_{\text{data}}} [D(x_{\text{input}}, x_{\text{gt}}))] + \lambda_{g} \cdot \mathbb{E}_{\hat{x} \sim P_{\hat{g}}} [\|\nabla_{\hat{x}} D(\hat{x})\|^2 - 1]^2$$  

(4)

### 3.3. Proposed Model

In this section, we describe in detail the model architecture we used in experiments. Table 1 shows the structure of the networks.

#### 3.3.1. Generator

The generator network consists of an encoding stage and decoding stage whose layers are mainly composed of 2d convolutional layers. In the encoding stage, the strided convolutional layers are used to reduce the spatial dimensions of feature maps while increasing the number of feature maps. In the following decoding stage, strided convolutional layers are used to restore the size of the input.

However, there is a major difference between the conventional encoding and decoding convolutional network and the network we used in this work: that is, there exists a skip connection between the encoding and decoding stages. This skip connection is a result of by concatenating the feature maps from encoding stages into the channel axis of the corresponding decoding stages. Therefore, different levels of features can be extracted through the encoding stage and reused using the shortcut in the corresponding decoding stage, resulting in easier and more efficient information flow through the network. This kind of network structure has been used by many, including [2,5,12] and its effectiveness has been proven. We used batch normalization after each convolutional layer and used a leaky rectified linear unit (ReLU) as a nonlinearity, except in the last layer, where the ReLU was used for the music source separation task and sigmoid for the AMT task.

#### 3.3.2. Discriminator

When using the conditional Wasserstein GANs, the input of the discriminator becomes the concatenation of either ($x_{\text{input}}, x_{\text{est}}$) or ($x_{\text{input}}, x_{\text{gt}}$) across the channel axis. Our discriminator network consists of strided convolutional layers using a leaky ReLU as a nonlinear function, except in the last layer, which has no nonlinear function. Here, the output of the last layer is a matrix rather than a scalar value. Thus, each pixel of the final output matrix corresponds to each different receptive region having the same receptive field size in the input. Next, to obtain a scalar value for calculating the Wasserstein distance, we take the mean value of the matrix from the final output layer. Note that,
by setting a suitable network configuration, we can adjust
the size of the receptive field to be smaller than the size of
the input data. This procedure enables the discriminator to
make a decision over a different time-frequency region of
the input [2].

4. EXPERIMENTS

Below we present the experimental results using the
proposed networks for the two different tasks: i.e., music
source separation and automatic music transcription.

4.1. Music Source Separation

For the music source separation experiments, we used
the public dataset DSD100, which was used in the Signal
Separation Evaluation Campaign (SiSEC) 2016. This
dataset contains “vocal,” “bass,” “drums,” and “other”
tracks for each song. There are 50 songs in the training
set and 50 songs in the test set, giving 100 songs in total.
All songs were recorded in stereo and digitized with a
sampling frequency of 44,100 Hz. As a preprocessing step,
we first split all the songs into 2 s segments with an overlap
of 1 s, and each segment was converted to a spectrogram
using the short-time Fourier transform algorithm. We used
this spectrogram representation as an input and a target
for our deep learning pipeline. We trained two models, the
singing voice separation model and the accompaniment
separation model, by setting \( x_{\text{input}} = x_{\text{mixture}} \), \( x_{\text{gt}} = x_{\text{gt,vocal}} \)
or \( x_{\text{gt,accomp}} \) and \( x_{\text{est}} = x_{\text{est,vocal}} \) or \( x_{\text{est,accomp}} \) in Eqs. (3) and
(4), respectively.

Finally, each of the estimated magnitude spectrograms
was converted to a wave file multiplied by the phase
component of the input mixture spectrogram and using the
inverse short-time Fourier transform algorithm. The per-
formance was evaluated using the most standard evaluation
metrics for a source separation task: signal-to-distortion
Ratio (SDR), signal-to-interference Ratio (SIR), and
signal-to-artifact Ratio (SAR) [13]. We evaluated the
results by taking the mean of the 50 songs in the test set
using the evaluation code SiSEC 2016. The best results
we achieved are shown in the Table 2. We compared our
proposed model with the top three deep-learning-based
models submitted in SiSEC 2016.

In Table 2, the SDR metrics of our algorithm and the
top three deep-learning-based algorithms (STO1 [14],
NUG4 [15], UHL3 [16]) submitted to SiSEC 2016 are
shown. Each SDR value is evaluated by first taking the
mean SDR of each song and then taking the mean over the
50 songs in the test set. STO1 is based on the deep neural
network (DNN) using patched overlapped STFT frames
on the input and output. NUG4 and UHL3 are based on
the multichannel source separation model. NUG4 uses the
DNN to estimate the target source to acquire a multi-
channel filter using an iterative expectation-maximization
algorithm. UHL3 is an ensemble model that blends two
models that use the DNN and RNN to compute the
multichannel Wiener filter. It also uses various data
augmentation methods, e.g., random swapping of two
channels, random amplitude scaling, and random pairing

| Layers | Input Layer | Encoder | Decoder |
|--------|-------------|---------|---------|
| Components | \( \text{Conv} \) \((64 \times 7 \times 7)\) | \( \text{deConv} \) \((512 \times 3 \times 3)\) | \( \text{Conv} \) \((64 \times 7 \times 7)\) |
| | \( \text{Conv} \) \((128 \times 5 \times 5)\) | \( \text{deConv} \) \((512 \times 3 \times 3)\) | \( \text{Conv} \) \((128 \times 5 \times 5)\) |
| | \( \text{Conv} \) \((256 \times 5 \times 5)\) | \( \text{deConv} \) \((512 \times 3 \times 3)\) | \( \text{Conv} \) \((256 \times 5 \times 5)\) |
| | \( \text{Conv} \) \((512 \times 3 \times 3)\) | \( \text{deConv} \) \((512 \times 3 \times 3)\) | \( \text{Conv} \) \((512 \times 3 \times 3)\) |
| | \( \text{Conv} \) \((512 \times 3 \times 3)\) | \( \text{deConv} \) \((256 \times 5 \times 5)\) | \( \text{Conv} \) \((256 \times 5 \times 5)\) |
| | \( \text{Conv} \) \((512 \times 3 \times 3)\) | \( \text{deConv} \) \((128 \times 5 \times 5)\) | \( \text{Conv} \) \((128 \times 5 \times 5)\) |
| | \( \text{Conv} \) \((512 \times 3 \times 3)\) | \( \text{deConv} \) \((64 \times 7 \times 7)\) | \( \text{Conv} \) \((64 \times 7 \times 7)\) |

| Output | \((512, 128)\) | \((2, 512)\) |

Table 1: Architecture of generator and discriminator for source separation and AMT. \( \text{Conv} \) denotes the same convolution
layer and parentheses are \((\text{channel} \times \text{stride-width} \times \text{stride-height})\).
between instruments from different songs. We show that our model was superior to STO1 and NUG4 but still has a gap of 0.66 dB and 0.61 dB between UHL3 for vocal and accompaniment, respectively. However, considering that our model does not use the information of stereophonic sound and data augmentation, we assume that our model holds further promise and there is still a room for improvement through combination with other possible approaches.

4.2. Automatic Music Transcription

AMT is a task that aims to output a symbolic representation from music input data in a raw-audio format. In this study, we aimed to output the result of the MIDI format as symbolic data. We used the MAPS dataset, which is the most commonly used dataset in the AMT field [17]. Among the sound sources in the MAPS dataset, only the MUS type was used for the training and evaluation. We used ENSTDkAm and ENSTDkCl as a test set, and the remaining types were used as a training set. Each wav file was preprocessed into a Constant-Q gram (CQgram) [18] and given as an input to the model. The MIDI files were converted into a pianoroll format using the prettymidi library1 and given as the ground truth input to the model. That is, the pair of audio CQgram and MIDI pianoroll representations were used for learning, and the model was trained to output a pianoroll from the CQgram by setting $x_{\text{input}} = x_{\text{CQgram}}$, $x_{\text{gt}} = x_{\text{midi}}$ and $x_{\text{est}} = x_{\text{est.midi}}$ in Eqs. (3) and (4). The trained model was evaluated using the ENSTDkAm and ENSTDkCl datasets, and the frame-level $F$-score and the note-level $F$-score were measured. Our results and those of the other deep-learning-based models, including the current state-of-the-art model, are shown in Table 3.

As shown in Table 3, the proposed method achieves a frame-level performance comparable to those of the other state-of-the-art methods. However, the note-level performance is worse than that reported by Hawthorne et al. [19], who explicitly modeled the onset frames of the notes, resulting in a large performance gain.

5. CONCLUSIONS

Using GANs, we designed and evaluated a deep learning model that is generally applicable to music source separation and AMT. In both tasks, we found that the model trained using both conventional loss and generative adversarial loss showed better performance than the model trained using only generative adversarial loss or conventional loss alone. In particular, the performance of music source separation was similar to that of the state-of-the-art model. The performance of AMT was lower for the frame-level than the conventional methods, but comparable for the note-level.

REFERENCES

[1] L. A. Gatys, A. S. Ecker and M. Bethge, “A neural algorithm of artistic style,” arXiv preprint arXiv:1508.06576 (2015).
[2] P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” arXiv preprint arXiv:1611.07004 (2016).
[3] M.-Y. Liu, T. Breuel and J. Kautz, “Unsupervised image-to-image translation networks,” arXiv preprint arXiv:1703.00848 (2017).
[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative adversarial nets,” in Advances in Neural Information Process-

---

1 [http://craffel.github.io/pretty-midi/](http://craffel.github.io/pretty-midi/)
S. Pascual, A. Bonafonte and J. Serrà, “Segan: Speech enhancement generative adversarial network,” *arXiv preprint arXiv:1703.09452* (2017).

H.-S. Choi, J.-h. Lee and K. Lee, “Singing voice separation using generative adversarial networks,” *Proc. 31st Conf. Neural Information Processing Systems (NIPS 2017)*, 5 pages (2017).

M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications* (Springer, Cham, 2015).

M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” *arXiv preprint arXiv:1701.04862* (2017).

M. Arjovsky, S. Chintala and L. Bottou, “Wasserstein generative adversarial networks,” *Proc. Int. Conf. Machine Learning*, pp. 214–223 (2017).

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. Courville, “Improved training of wasserstein gans,” *arXiv preprint arXiv:1704.00028* (2017).

M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784* (2014).

O. Ronneberger, P. Fischer and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention*, Springer, pp. 234–241 (2015).

E. Vincent, R. Gribonval and C. Févotte, “Performance measurement in blind audio source separation,” *IEEE Trans. Audio Speech Lang. Process.*, 14, 1462–1469 (2006).

F.-R. Stöter, A. Liutkus, R. Badeau, B. Edler and P. Magron, “Common fate model for unison source separation,” *Proc. ICASSP 2016*, pp. 126–130 (2016).

A. A. Nugraha, A. Liutkus and E. Vincent, “Multichannel music separation with deep neural networks,” *Proc. EUSIPCO 2016*, pp. 1748–1752 (2016).

S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi and Y. Mitsufuji, “Improving music source separation based on deep neural networks through data augmentation and network blending,” *Proc. ICASSP 2017*, pp. 261–265 (2017).

V. Emiya, R. Badeau and B. David, “Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle,” *IEEE Trans. Audio Speech Lang. Process.*, 18, 1643–1654 (2010).

J. C. Brown, “Calculation of a constant q spectral transform,” *J. Acoust. Soc. Am.*, 89, 425–434 (1991).

C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore and D. Eck, “Onsets and frames: Dual-objective piano transcriptions,” *arXiv preprint arXiv: 1710.11153* (2017).

S. Sigtitia, E. Benetos, S. Cherla, T. Weyde, A. Garcez and S. Dixon, “An RNN-based music language models for improving automatic music transcription,” *Proc. 15th Int. Soc. Music Information Retrieval Conf. (ISMIR 2014)*, pp. 53–58 (2014).