

# Demonstration of PerformanceNet: A Convolutional Neural Network Model for Score-to-Audio Music Generation

Yu-Hua Chen, Bryan Wang and Yi-Hsuan Yang  
 Research Center for IT Innovation, Academia Sinica, Taiwan  
 {cloud60138, bryanw, yang}@citi.sinica.edu.tw

## Abstract

We present in this paper PerformanceNet, a neural network model we proposed recently to achieve score-to-audio music generation. The model learns to convert a music piece from the symbolic domain to the audio domain, assigning performance-level attributes such as changes in velocity automatically to the music and then synthesizing the audio. The model is therefore not just a neural audio synthesizer, but an AI performer that learns to interpret a musical score in its own way. The code and sample outputs of the model can be found online at <https://github.com/bwang514/PerformanceNet>.

## 1 Introduction

Music is generally considered as organized sounds created by human and is transmitted as audio waveforms. People have designed musical symbols to notate various aspects of music. Accordingly, we can transcribe a sound recording in a handwritten or printed form, facilitating the communication of the “content” of the music. However, given the same musical score sheet, different musicians can interpret the music in different ways and use their personal “styles” while performing the music. Such performance-level attributes of the music are usually easier to find directly in the audio waveform, not in the symbolic music notation.

Recent years have witnessed a growing interest in building machine models for music generation. However, most existing work focuses on only one of the two main domains of music—symbolic or audio—rather than the two domains at the same time. People working on *symbolic-domain music generation*, a.k.a. algorithmic composition, typically focus on generating original musical content such as melody and chords and tend to use off-the-shelf audio synthesizers to play the music they generate (e.g., [Yang *et al.*, 2017; Dong *et al.*, 2018a; Brunner *et al.*, 2018; Simon *et al.*, 2018]). And, people working on *audio-domain music generation* usually focus on the synthesis part only and aim to generate original sounds of whatever musical content (e.g., [Engel *et al.*, 2017; Engel *et al.*, 2019; Marafioti *et al.*, 2019]). There are some things in between that cannot be modeled without considering data from the aforementioned two domains together, such as the performance-level attributes and playing styles.

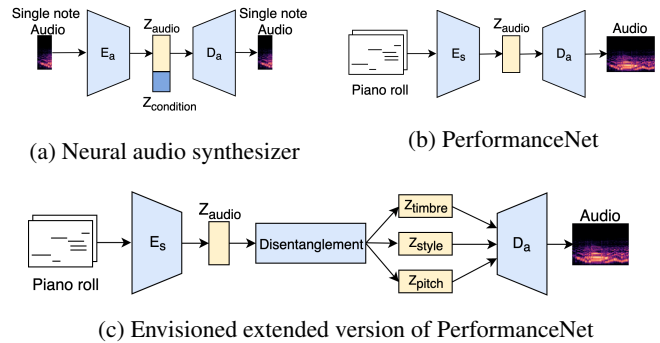


Figure 1: Diagram of encoder/decoder networks for different tasks; E, D, z denote the encoder, decoder and latent code, respectively.

In a prior work, we address this gap by proposing a neural network model, dubbed the “PerformanceNet,” that takes symbolic representations of a music piece as input and generates as output a sound recording playing that piece expressively [Wang and Yang, 2019]. The goal of PerformanceNet is to predict the performance-level attributes, such as changes in velocity (i.e., dynamics/loudness) and modulations in pitch (e.g., *vibrato*) that a human performer may apply while performing the music. As shown in Figure 1(b), the model also learns to synthesize audio in an end-to-end manner. To our knowledge, PerformanceNet, and the work presented independently and concurrently to our work in [Kim *et al.*, 2019], represent the first models that learn explicitly the score-to-audio mapping of music, for arbitrary instruments.

In this demonstration, we discuss the difference between the *note-level synthesis* task addressed by existing neural audio synthesizers (e.g., [Engel *et al.*, 2019]) and the *phrase-level synthesis* task addressed by PerformanceNet. We also present a graphical user interface with which people can load and edit a musical score and then ask our PerformanceNet to perform it expressively using different instruments.

## 2 Note-level & Phrase-level Audio Generation

Most existing neural audio synthesizers employ a neural network model to learn to generate high-quality musical sounds. The model is usually trained with audio recordings of isolated musical notes (e.g., C4 and C5) from different instruments. Therefore, we refer to them as performing note-level

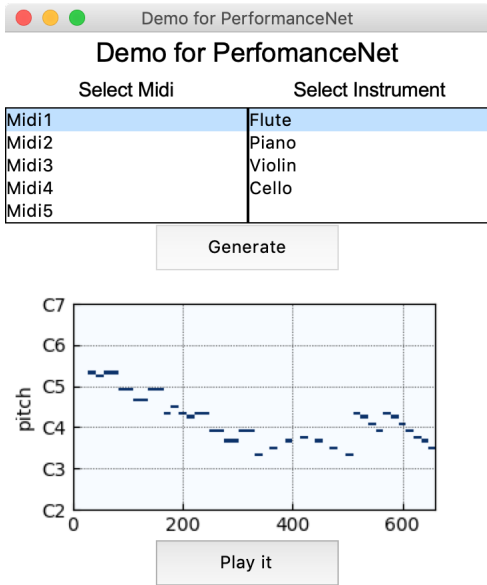


Figure 2: A simple GUI for demonstrating PerformanceNet.

audio generation (or synthesis). When an encoder/decoder architecture is used, as the case in [Engel *et al.*, 2017] (see Figure 1(a)), the model learns in the latent space the embeddings of the musical timbre of different instruments (marked as  $z_{\text{audio}}$  in Figure 1(a)). It is therefore possible to sample from the latent space to create sounds of new instruments, or to interpolate the sounds of existing instruments. To control the pitch, both [Engel *et al.*, 2017] and [Engel *et al.*, 2019] concatenate with the latent code a one-hot vector (marked as  $z_{\text{condition}}$  in Figure 1(a)) representing the pitch of the sound to be generated. The pitch vector is one-hot (i.e., only one element of the vector takes the value one and the rest are zero), as the models synthesize audio one note at a time. The sounds generated can be realistic and expressive, since the model is trained with real-world audio recordings.

An AI performer, on the other hand, learns to convert a music piece from the symbolic domain to the audio domain, assigning performance-level attributes such as changes in velocity automatically to the music and then synthesizing the audio. The model is trained with pairs of the symbolic representation and audio recordings of musical phrases comprising of multiple notes. The input representation used by PerformanceNet, for example, is a symbolic representation called the *pianoroll* [Dong *et al.*, 2018b], a binary, scoresheet-like matrix representing the presence of notes over different time steps for a single instrument. We can extend it to a tensor, i.e., multitrack pianoroll, to represent the score of multiple instruments. The output representation used by PerformanceNet is the (magnitude) spectrogram of the corresponding audio recording, so what PerformanceNet learns is actually a matrix-to-matrix mapping. When an encoder/decoder architecture is used, as the case in PerformanceNet (illustrated in Figure 1(b)), the latent code contains not only timbre but also style and pitch information. Therefore, disentanglement techniques [Hung *et al.*, 2019] may be needed to disentangle these elements, as illustrated in Figure 1(c).

We can now see that a core task of an AI performer is therefore score-informed phrase-level audio generation. Unlike the case of note-level generation, here we need to learn how to connect different notes while playing (e.g., using playing techniques such as *slide*, *hammer-on* and *pull-off* as the case in guitar music [Chen *et al.*, 2015; Su *et al.*, 2019]), and to play the same pitch differently depending on the position of that note in a phrase (e.g., whether it is at the downbeat) [Li *et al.*, 2015]. Moreover, an AI performer holds the potential to learn better the phrase-level attributes of music, and accordingly the playing style of different musicians [Shih *et al.*, 2017]. This might be done, for example, by conditioning the PerformanceNet with an one-hot vector indicating the musician who played that phrase.

### 3 Model Architecture of PerformanceNet

The PerformanceNet consists of two subnets. The first subnet, the ContourNet, uses a convolutional encoder/decoder architecture to roughly convert the pianoroll to the spectrogram. The second subnet, the TextureNet, further improves the result of the ContourNet by refining the details of the partials of each note in the spectra with convolutional layers of a multi-band residual design.<sup>1</sup> The job of the ContourNet is akin to performing *domain translation* [Gatys *et al.*, 2016] (between the symbolic and audio domains of music), whereas the TextureNet is doing *super resolution* [Ledig *et al.*, 2017]. Please see [Wang and Yang, 2019] for more technical details.

### 4 Demo System

For the purpose of demonstration, we build a graphical user interface for PerformanceNet, as depicted in Figure 2. Users can select a MIDI file or upload one. After the score is given, users can choose the instrument to play the piece. The audio can be generated on-the-fly by our model.

### References

- [Brunner *et al.*, 2018] Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 23–27, 2018.
- [Chen *et al.*, 2015] Yuan-Ping Chen, Li Su, and Yi-Hsuan Yang. Electric guitar playing technique detection in real-world recordings based on f0 sequence pattern recognition. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 708–714, 2015.
- [Dong *et al.*, 2018a] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. MuseGAN: Symbolic-domain music generation and accompaniment with multi-

<sup>1</sup>We found that TextureNet’s refinement is two-fold. Firstly, it sharpens the blurred frequency bins close to the fundamental frequency, which contributes to better reconstructed audio quality as pointed out in [Huang *et al.*, 2018]. Secondly, overtones with higher frequencies, which contribute to the perception of realistic timbre, are gradually added to the spectrogram by multi-band residual blocks, demonstrating the coarse-to-fine rendering process. We show figures demonstrating these in our project website.

- track sequential generative adversarial networks. In *Proc. AAAI Conf. Artificial Intelligence*, 2018.
- [Dong *et al.*, 2018b] Hao-Wen Dong, Wen-Yi Hsiao, and Yi-Hsuan Yang. Pypianoroll: Open source Python package for handling multitrack pianoroll. In *Proc. Int. Soc. Music Information Retrieval Conf.*, 2018. Late-breaking paper; [Online] <https://github.com/salu133445/pypianoroll>.
- [Engel *et al.*, 2017] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi. Neural audio synthesis of musical notes with WaveNet autoencoders. *arXiv preprint arXiv:1704.01279*, 2017.
- [Engel *et al.*, 2019] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. GANSynth: Adversarial neural audio synthesis. In *Proc. Int. Conf. Learning Representations*, 2019.
- [Gatys *et al.*, 2016] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.
- [Huang *et al.*, 2018] 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. *CoRR*, abs/1811.09620, 2018.
- [Hung *et al.*, 2019] Yun-Ning Hung, I-Tung Chiang, Yi-An Chen, and Yi-Hsuan Yang. Musical composition style transfer via disentangled timbre representations. In *Proc. Int. Joint Conf. Artificial Intelligence*, 2019.
- [Kim *et al.*, 2019] Jong Wook Kim, Rachel Bittner, Aparna Kumar, and Juan Pablo Bello. Neural music synthesis for flexible timbre control. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pages 176–180, 2019.
- [Ledig *et al.*, 2017] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 4681–4690, 2017.
- [Li *et al.*, 2015] Pei-Ching Li, Li Su, Yi-Hsuan Yang, and Alvin W. Y. Su. Analysis of expressive musical terms in violin using score-informed and expression-based audio features. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 809–815, 2015.
- [Marafioti *et al.*, 2019] Andrés Marafioti, Nicki Holighaus, Nathanaël Perraudin, and Piotr Majdak. Adversarial generation of time-frequency features with application in audio synthesis. *CoRR*, abs/1902.04072, 2019.
- [Shih *et al.*, 2017] Chi-Ching Shih, Pei-Ching Li, Yi-Ju Lin, Alvin W. Y. Su, Li Su, and Yi-Hsuan Yang. Analysis of expressive musical terms in violin using score-informed and expression-based audio features. In *Proc. Int. Conf. Digital Audio Effects*, 2017.
- [Simon *et al.*, 2018] Ian Simon, Adam Roberts, Colin Raffel, Jesse Engel, Curtis Hawthorne, and Douglas Eck. Learning a latent space of multitrack measures. In *Proc. Int. Soc. Music Information Retrieval Conf.*, 2018.
- [Su *et al.*, 2019] Ting-Wei Su, Yuan-Ping Chen, Li Su, and Yi-Hsuan Yang. TENT: Technique-embedded note tracking for real-world guitar solo recordings. *Transactions of the International Society for Music Information Retrieval*, 2019.
- [Wang and Yang, 2019] Bryan Wang and Yi-Hsuan Yang. PerformanceNet: Score-to-audio music generation with multi-band convolutional residual network. In *Proc. AAAI Conf. Artificial Intelligence*, 2019.
- [Yang *et al.*, 2017] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. MidiNet: A convolutional generative adversarial network for symbolic-domain music generation. In *Proc. Int. Soc. Music Information Retrieval Conf.*, 2017.