研究概要報告書 【サウンド技術振興部門】

研究題目	Deep-learning-based neural source-filtering models for fast and high-quality music signal generation	報告書作成者	Xin Wang
研究従事者	Erica Cooper, Yi Zhao		
研究目的	How to generate high-quality music audio waveforms from a digital system is one of the intensively studied research topics on digital audio generation. Before the latest era of deep learning, music audio generation mainly uses sampled-based or physical-model-based approaches. The former uses a dataset of pre-recorded audio waveforms of one instrument (e.g., the 88 notes on a grand piano), and it generates a music audio waveform by concatenating individual notes. The physical-model-based approach, however, simulates the audio generation process through a physic model of the instrument. Both the sample-based and physical-model-based approaches can produce high-quality music audios, but they require laborious human efforts when they are adopted to model a new instrument: the sample-based approach needs human effort to build the sample dataset, and the physical-model-based approach requires numerical simulation. In recent years, many deep-learning-based approaches are proposed to learn a generative model of the music audio in a data-driven manner. The users do not need to annotate and build the audio sample database, neither do they need to do physical simulation. What they need to do is to provide the training data, define the DNN architecture, and train the DNN with a specified criterion. The deep learning methods make it convenient to build applications for music audio generation. However, the convenience is not free. Users need to spend more time searching for a good DNN architecture, and there is no general guideline for that. A sub-optimal DNN architecture may produce low-quality sound or require too much computation time. This project contributes to the recent research on deep-learning-based music audio generation and explains our proposed DNN models and training recipes. Although the initial goal when submitting this proposal is to *build a single neural-network-based model to produce high-quality sounds of multiple types of instruments,* we have partially achieved this goal by the time when this proposal was accepted [1], and we call the proposed model as neural source-filter (NSF) waveform model [2]. In this project, we further extend this research and try our best practices to ***build an NSF model to produce high-quality sounds for polyphonic instruments such as a piano.*** We will release the Pytorch code for this project so that everyone can try to build a DNN-based piano audio generation model. Although we focus on piano, the code and methods can be applied to other instruments as we have demonstrated in [1]. [1] Zhao, Y., Wang, X., Juvela, L. & Yamagishi, J. Transferring neural speech waveform synthesizers to musical instrument sounds generation. in *Proc. ICASSP* 6269-6273 (IEEE, 2020). doi:10.1109/ICASSP40776.2020.9053047 [2] Wang, X., Takaki, S. & Yamagishi, J. Autoregressive Neural F0 Model for Statistical Parametric Speech Synthesis. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **26**, 1406-1419 (2018)		

研究内容

Application scenarios: for music audio generation, we need to define the system input from which the music audio is generated. Here, we borrow the terminologies from text-to-speech synthesis and define the following scenarios: 1) the copy-synthesis scenario where the input is the acoustic features of the music audio; 2) the pipeline synthesis scenario where the input is MIDI, and an acoustic model is used to convert the MIDI into acoustic features for the NSF model; 3) the end-to-end synthesis scenario where the NSF directly convert the MIDI input into the music audio. These scenarios are illustrated in Figure 1. By evaluating the NSF model under these three scenarios, we can understand how well the NSF model can produce piano sound for various applications.

Research focus: *The core of this project is to find good neural network architectures for the NSF-based piano waveform model.* The original NSF model consists of three modules: a condition module that transforms and up-samples the input features, a source module that produces a sine-excitation signal given the fundamental frequency (F0) of the audio, and a neural filter module that transforms the excitation signal into the output audio through multiple dilated convolution blocks. However, the design of the original NSF is mainly for speech waveforms and may not be the best for polyphonic instruments because:

◎ The original source module only accepts a single F0 at one time and can only be used for monophonic instruments. However, the piano is polyphonic, and the notes played simultaneously (i.e., chords) cannot be represented by a single F0 value;

◎ The original neural filter module uses a harmonic-plus-noise (HNS) structure, and it simulates the human speech production process.

Accordingly, we revised the NSF model architecture and proposed the NSF piano waveform model, which is illustrated in Figure 2:

◎ The original source module is removed, and we can use either Gaussian noise or a sine-excitation signal produced from MIDI API;

◎ The neural filter module is simplified and only contains a single branch of convolution blocks.

◎ A new type of acoustic feature called MIDI filter-bank feature is proposed as an alternative to the widely used Mel-spectrogram. As Figure 3 plots, this feature uses a filter bank defined on the MIDI notes, and it is more compatible with the MIDI piano roll input.

Experiments: we evaluated the NSF's performance in the three scenarios. We also included strong baselines: a sample-based approach (Fluidsynth https://www.fluidsynth.org/), a physical model-based approach (Pianoteq https://www.modartt.com/pianoteq), and another deep-learning-based approach (PFNet [3]). For the acoustic model, which is not the topic of this project, we utilized the famous TTS model called Tacotron [4] and another deep neural network from PFNet. Given the audios generated from the systems, we conducted a large-scale listening test and asked more than 200 amateur human listeners to rate the quality of the generated audios.

[3] Wang, B. & Yang, Y.-H. PerformanceNet: Score-to-audio music generation with multi-band convolutional residual network. in *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 33 1174–1181 (2019).

[4] Wang, Y. *et al.* Tacotron: Towards End-to-End Speech Synthesis. in *Proc. Interspeech* 4006–4010 (2017).

研究概要報告書【サウンド技術振興部門】

研究のポイント	As explained in 研究内容, the points we need to consider include: ◎ How to define an NSF source module for polyphonic music audios; ◎ How to revise the NSF neural filter module so that it can be used for music; ◎ Whether we can find better acoustic features for NSF-based MIDI-to-audio applications. Meanwhile, it is important to testify the model performance with the following points into consideration: ◎ Varied but strong baselines. We included both sampled-based (FluidSynth https://www.fluidsynth.org/), physical model-based (Pianoteq, https://www.modartt.com/pianoteq), and another deep learning model called PFNet; ◎ A large-scale listening test. In this project, we conducted a listening test and asked more than 200 (paid) participants to evaluate the audio quality. We also conducted statistical analysis to further justify the experimental results; ◎ A large-scale dataset. We used the MAESTRO dataset, which contains more than 200 hours' piano audio and aligned MIDI data.
研究結果	Detailed results are illustrated in Figure 4. ◎ For scenario 1 – piano audio synthesis from acoustic features: the NSF piano waveform model with MIDI filter-bank acoustic feature and excitation from MIDI-API (i.e., **abs-mfb-sin**) achieved a quality score (MOS) of 3.87, which is close to the original audio data in MASESTRO dataset (MOS=4.04). In fact, the difference of the MOS scores (3.87 vs 4.04) is not statistically significant (cf. the submitted paper). **This result indicates that the proposed NSF piano waveform model can synthesize high-quality piano audios. The proposed MIDI filter-bank feature is also effective;** ◎ For scenario 2 & 3 – piano audio synthesis from MIDI data: audios generated from the NSF piano waveform model degraded when the acoustic features are predicted from Tacotron or PFNet model (cf. scenario 2 in Figure 4, where the highest MOS is 3.19), or when the NSF piano waveform model uses MIDI piano roll as input (cf. scenario 3 in Figure 4, where the highest MOS is 2.83). Audio samples can be found on this page https://nii-yamagishilab.github.io/samples-xin/main-midi2audio.html#label-midi2audio. We will release the Pytorch code recipe and code when it is ready. The submitted paper is available on Arxiv: http://arxiv.org/abs/2104.12292.
今後の課題	The experimental results suggest that the NSF piano waveform model can generate high-quality piano sounds when the input acoustic features are also high in quality. Therefore, one future research topic is to improve the performance of the acoustic model. Another future work is to use the proposed NSF models for other polyphonic instruments, such as Cello. Although there is no large-scale database for other polyphonic instruments, our team is preparing a new database from professional performers.

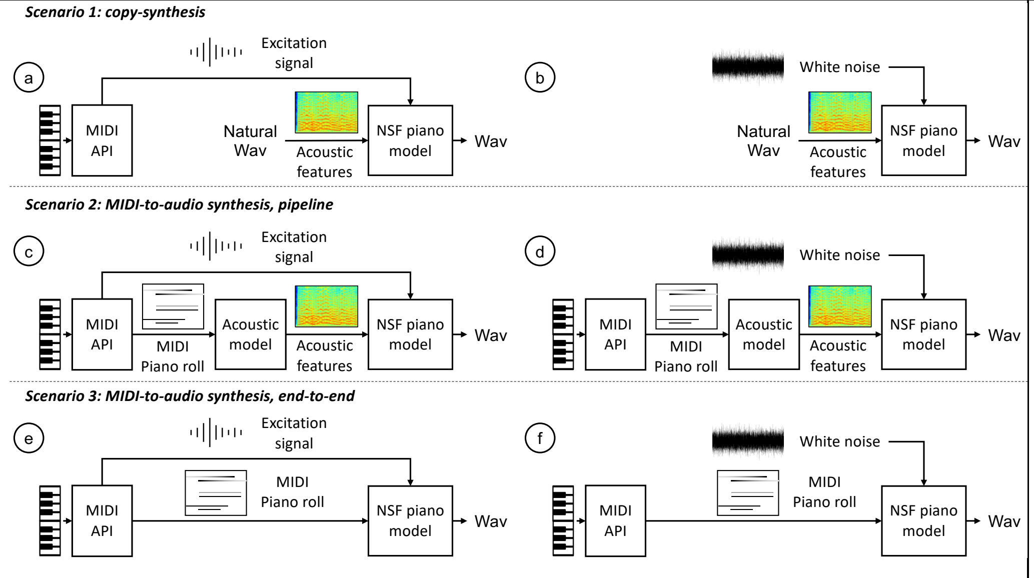


Figure 1. Three experimental scenarios, and six types of systems from ⓐ to ⓕ.

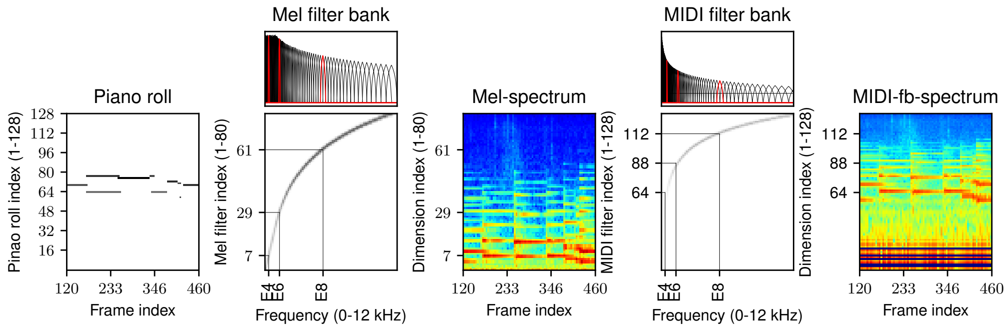


Figure 3. Comparison between Mel-spectrum (bottom middle) and proposed MIDI-based filter-bank feature (bottom right) of one piano audio waveform. The filter banks (top) are also plotted, together with the frequency-warping function (2nd to left and 2nd to right, bottom). Note that the MIDI filter-bank based feature is similar to the piano roll (bottom left) of the audio.

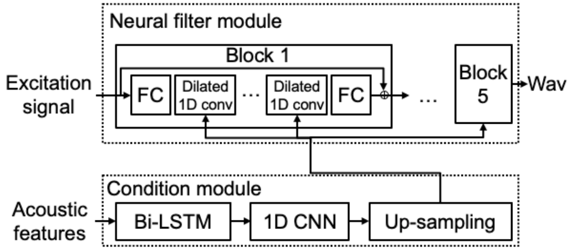
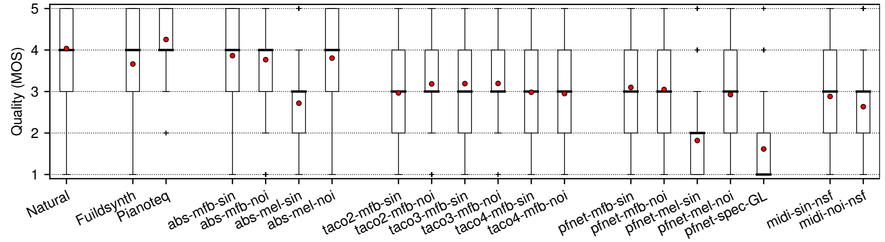


Figure 2. Proposed NSF piano waveform model. Block from 2 to 5 use the same structure as Block 1. FC denotes a fully connected layer. Dilated 1D conv block is detailed in [1]



	System ID	Type	Acoustic model	Acoustic feature	Excit. signal	Wave. model	Pitch mismatch		MOS (mean)
							note	chord	
Reference, baselines	Natural		-	-	-	-	-	-	4.04
	Fluidsynth	-	Sample-based MIDI-to-audio software				5.20	6.77	3.66
	Pianoteq	-	Physical-model MIDI-to-audio software				4.82	6.50	4.25
Scenario 1	abs-mfb-sin	a	-	midi-fb	sine	NSF	-	-	3.87
	abs-mfb-noi	b	-	midi-fb	noise	NSF	-	-	3.77
	abs-mel-sin	a	-	mel-spc.	sine	NSF	-	-	2.72
	abs-mel-noi	b	-	mel-spc.	noise	NSF	-	-	3.81
Scenario 2	taco2-mfb-sin	c	taco2	midi-fb	sine	NSF	4.61	6.34	2.97
	taco2-mfb-noi	d	taco2	midi-fb	noise	NSF	4.66	6.36	3.18
	taco3-mfb-sin	c	taco3	midi-fb	sine	NSF	4.78	6.48	3.19
	taco3-mfb-noi	d	taco3	midi-fb	noise	NSF	4.89	6.53	3.19
	taco4-mfb-sin	c	taco4	midi-fb	sine	NSF	4.86	6.39	2.98
	taco4-mfb-noi	d	taco4	midi-fb	noise	NSF	4.97	6.42	2.95
	pfnet-mfb-sin	c	PFNet	midi-fb	sine	NSF	5.59	7.14	3.10
	pfnet-mfb-noi	d	PFNet	midi-fb	noise	NSF	5.78	7.26	3.05
	pfnet-mel-sin	c	PFNet	mel-spec.	sine	NSF	5.66	7.17	1.82
	pfnet-mel-noi	d	PFNet	mel-spec.	noise	NSF	5.74	7.25	2.93
	pfnet-spec-GL	-	PFNet	spec.	-	GL	-	-	1.62
Scenario 3	midi-sin-nsf	e	-	-	sine	NSF	4.32	6.40	2.88
	midi-noi-nsf	f	-	-	noise	NSF	4.40	6.08	2.63

Figure 4. Scatter plot of Listening test scores (top) and table of systems details (bottom). **Mean score of each system (MOS) is denoted by a red dot, and a higher mean score denotes a better performance**. System type a - f are shown in Figure 1.