DEEP PERFORMER: SCORE-TO-AUDIO MUSIC PERFORMANCE SYNTHESIS

Hao-Wen Dong [1,2] [*]   *Cong Zhou* [1]   *Taylor Berg-Kirkpatrick* [2]   *Julian McAuley* [2]

[1] Dolby Laboratories   [2] University of California San Diego

ABSTRACT

Music performance synthesis aims to synthesize a musical score into a natural performance. In this paper, we borrow recent advances in text-to-speech synthesis and present the Deep Performer—a novel system for score-to-audio music performance synthesis. Unlike speech, music often contains polyphony and long notes. Hence, we propose two new techniques for handling polyphonic inputs and providing a fine-grained conditioning in a transformer encoder-decoder model. To train our proposed system, we present a new violin dataset consisting of paired recordings and scores along with estimated alignments between them. We show that our proposed model can synthesize music with clear polyphony and harmonic structures. In a listening test, we achieve competitive quality against the baseline model, a conditional generative audio model, in terms of pitch accuracy, timbre and noise level. Moreover, our proposed model significantly outperforms the baseline on an existing piano dataset in overall quality.

Index Terms— Audio synthesis, computer music, music information retrieval, machine learning, neural network

1. INTRODUCTION

Music synthesis is a complex process that involves both the physics behind a musical instrument and the art of music performance. It remains challenging for a machine to synthesize a natural performance for several reasons. First, it requires a computational model for interpreting and phrasing a musical score. Second, it requires either an explicit or implicit model of the physics and acoustics by which a musical instrument sounds. Third, it requires an understanding of different playing techniques and styles for a musical instrument. While most existing systems only address one of these three challenges at a time, we aim to tackle all these challenges with a data-driven approach using machine learning in this work. We present the Deep Performer—a novel three-stage system for score-to-audio music synthesis, as illustrated in Fig. 1.

Prior work has studied music synthesis via various approaches. One line of research focuses on generating realistic samples of musical notes [1–3], while in this work we aim to generate the full performance. Some approach music synthesis by conditioning generative audio models with aligned
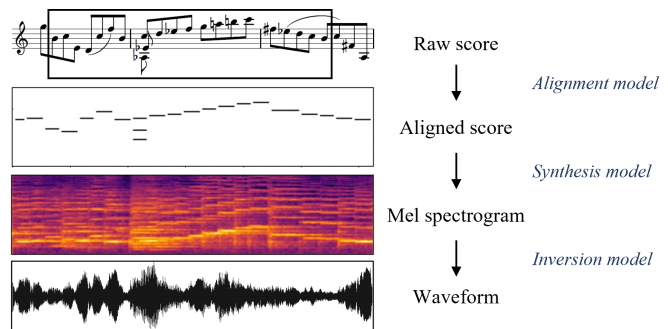

Fig. 1: An overview of the proposed three-stage pipeline for score-to-audio music performance synthesis.

piano rolls [4,5], which we will include as the baseline model in our experiments. Others study synthesizing audio from the fundamental frequency (F0) contour and loudness curve extracted from a recording [6,7], or from lyrics and demo singing audio [8]. On the other hand, some use neural networks to generate expressive timing and dynamics from raw scores [9]. Many have also studied inverting mel spectrograms back to waveforms [10–13], including Hifi-GAN [13], which we will use as the inversion model in our proposed system. To the best of our knowledge, prior work on deep neural network based music synthesis either requires an input with expressive timing [4–8,14–16] or allows only monophonic (i.e., one pitch at a time) inputs [6, 8, 17]. Our proposed system represents the first that allows unaligned, polyphonic scores as inputs.

In light of the similarity between text-to-speech (TTS) and score-to-audio synthesis, we borrow recent advances from TTS synthesis [18] to music synthesis and propose a three-stage system for score-to-audio music synthesis. Despite the similarity, music synthesis differs from speech synthesis in that music often contains polyphony, and that long notes are common in music. In order to handle polyphonic music, we propose a new *polyphonic mixer* for aligning the encoder and decoder in a transformer encoder-decoder network [19]. To provide a fine-grained conditioning to the model, we propose a new *note-wise positional encoding* so that the model can learn to behave differently at the beginning, middle and end of a note. Due to the lack of a proper dataset for training a score-to-audio music synthesis model, we collect and release a new dataset of 6.5 hours of high-quality violin recordings along with their scores and estimated alignments. Through our experiments, we show the effectiveness of our proposed

---

[*]Work done during an internship at Dolby. Contact: `hwdong@ucsd.edu`

system both qualitatively and quantitatively. Finally, we conduct a subjective listening test to compare our proposed model against a baseline model that uses Hifi-GAN [13] to synthesize the waveform directly from an aligned piano roll. Audio samples can be found on our project website.[1]

2. METHODS

We illustrate in Fig. 1 the proposed three-stage system for score-to-audio music synthesis, which consists of the following three components: (1) an *alignment model* that predicts the expressive timing for each note from a musical score, (2) a *synthesis model* that synthesizes the mel spectrogram from the aligned score, and (3) an *inversion model* that generates the audio waveform given the synthesized mel spectrogram.

2.1. Alignment model

The alignment model consists of a transformer encoder that takes as inputs a sequence of notes and the tempo, followed by a fully-connected layer that outputs the onset and duration of each note. The input score uses metric time with a musically-meaningful unit, e.g., quarter notes, while the output alignment is in the unit of frames. Each note is specified by its pitch, onset, duration and (optional) velocity. In addition, we provide the performer IDs so that the model can learn the different playing styles of performers. The alignment model is trained to minimize the mean squared error (MSE) between the ground truth and predicted onsets and durations, in frames.

2.2. Synthesis model

Given the similarity between TTS and score-to-audio synthesis, we propose a transformer encoder-decoder model for our synthesis model based on [20]. In [20], each text embedding produced by the encoder is expanded multiple times according to its duration, and then the expanded text embeddings are concatenated to obtain the frame embeddings to be fed to the decoder. This is called the state expansion mechanism [20,21]. However, unlike speech, music often contains polyphony. In order to handle polyphonic inputs, we propose the *polyphonic mixer*. As illustrated in Fig. 2, the encoder first encodes the input notes into a sequence of note embeddings. Then, the polyphonic mixer mixes the note embeddings into a sequence of frame embeddings by summing up the note embeddings for the same frame according to their onsets and durations. Finally, the decoder decodes the frame embeddings into a sequence of mel spectrogram frames.

In the state expansion mechanism [20,21], the output vectors remain constant for the duration of a note, and the positional information within each note is missing. However, we argue that such note-wise positional information is critical for the model to behave differently at the beginning, middle and
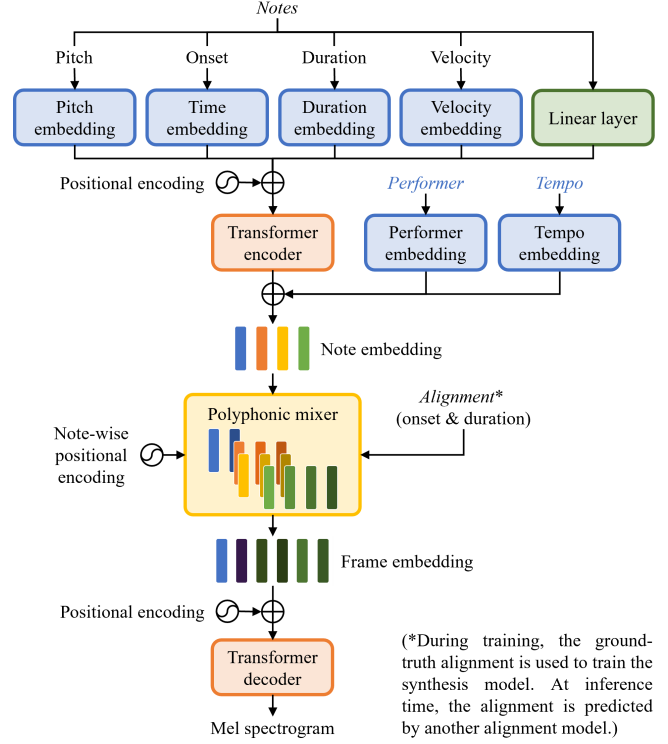
Fig. 2: An illustration of the proposed synthesis model.

end of a note. Hence, we propose the *note-wise positional encoding* to provide a fine-grained conditioning to the decoder. Mathematically, let $p \in [0, 1]$ be the relative position within a note. For a note embedding $\mathbf{v}_{\text{note}}$, we have the corresponding frame embedding at position p as $\mathbf{v}_{\text{frame}} = (1 + p\mathbf{w}) \odot \mathbf{v}_{\text{note}}$, where $\mathbf{w}$ is a learnable vector initialized to small random numbers so that $\mathbf{v}_{\text{frame}} \approx \mathbf{v}_{\text{note}}$ initially. The synthesis model is trained to minimize the MSE between the synthesized mel spectrograms and the ground truth, in log scale.

2.3. Inversion model

Prior work has studied various approaches for synthesizing waveforms from mel spectrograms [10–13]. In this work, we adopt the state-of-the-art Hifi-GAN model [13] as our inversion model. We note that the proposed three-stage pipeline allows us to use different datasets for training the models. For example, training the inversion model does not require aligned data and thus it can be trained on a larger dataset as unaligned data are relatively easier to acquire.

3. DATA

Due to the lack of a dataset that provides paired audios and scores with fine alignments for training our proposed system, we compile a new dataset of 6.5 hours of professional violin recordings along with their scores and estimated alignments. For copyright concern, we choose Bach's sonatas and partitas
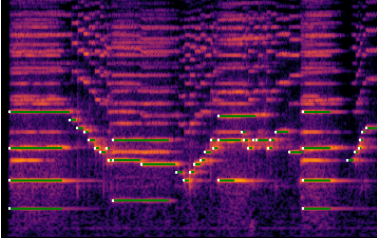
Fig. 3: An example of the constant-Q spectrogram of the first 20 seconds of a violin recording and the estimated onsets (white dots) and durations (green lines).

for solo violin (BWV 1001–1006) for the ease to acquire high-quality public recordings from the web. The dataset consists of performances by 17 violinists recorded in various recording setups. To acquire the alignment between a recording and its score, we synthesize the scores using FluidSynth [22], an open-source software synthesizer, with MuseScore General SoundFont [23] and perform dynamic time warping on the constant-Q spectrogram of the synthesized audio and that of the recording. We present in Fig. 3 an example of the dataset and its estimated alignment. To facilitate future research on score-to-audio music synthesis, we release the dataset and the source code for the alignment process to the public.[2] As discussed in Section 2.3, the inversion model does not require aligned data for training, and thus we also collect an internal dataset of 156 hours of commercial recordings to train the inversion model. Apart from violin, we also consider the MAESTRO dataset [5], which contains 200 hours of piano recordings with finely-aligned MIDI recordings for 10 competition years of the International Piano-e-Competition [24]. However, since it does not provide the raw scores, we can only train the synthesis and inversion models on this dataset.

4. EXPERIMENTS & RESULTS

4.1. Implementation details

We use 3 transformer layers in the encoder for the alignment model. The synthesis model shares the same encoder architecture as the alignment model and has 6 transformer layers in the decoder. We use 128 dimensions for all embeddings. For the inversion model, we use the same network architecture as the Hifi-GAN v2 model in [13]. We use velocity information only for the piano dataset as it is only available in this dataset. Since performer information is unavailable for the piano dataset, we use the competition years as the performer IDs. We use a temporal resolution of 24 time steps per quarter note for the scores. We downsample the audios to 16 kHz mono and use a hop size of 256 in spectrogram computation, i.e., a temporal resolution of 16 ms. The audios are sliced into 5-second clips for training, where 10% of them are reserved for validation purpose. We use the Adam optimizer [25] with a batch size of 16. Unlike [20], we train the alignment and synthesis models separately as we find that joint training hinders convergence. We train the alignment model for 10K steps and

Fig. 4: Examples of the alignments predicted by (a) the constant-tempo baseline model and (b) Deep Performer, our proposed model. (c) shows the input score.
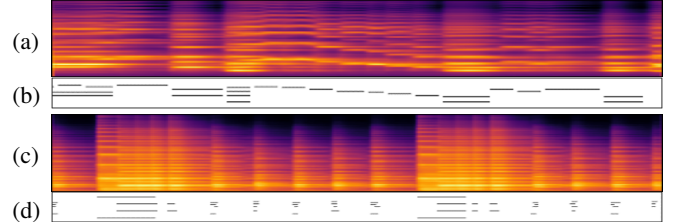

Fig. 5: Examples of the mel spectrograms, in log scale, synthesized by our proposed model for (a) violin and (c) piano. (b) and (d) show the input scores for (a) and (c), respectively.

all the synthesis models for 100K (violin) and 250K (piano) steps. For each dataset, the inversion model is trained for 1M steps and shared by different synthesis models. We base our implementation on the code kindly released in [13, 26]. We use pretty_midi [27] and MusPy [28] to process the scores.

4.2. Qualitative and quantitative results

We show in Fig. 4 an example of the alignment predicted by our proposed alignment model alongside that generated by assuming a constant tempo. We can see that our proposed model is able to predict realistic timing and insert rests between notes. To showcase the effectiveness of the proposed polyphonic mixer, we present in Fig. 5 examples of the synthesized mel spectrograms for two polyphonic scores, where we can observe clear harmonic structures and polyphony.
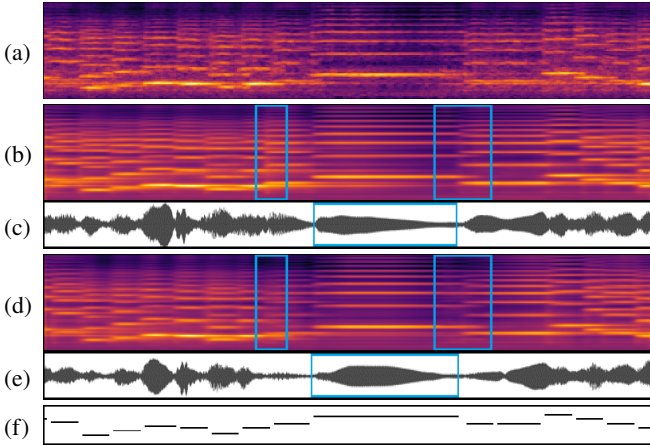
Next, we compare our proposed synthesis model against a baseline model that uses a Hifi-GAN [13] to synthesize the waveform directly from an aligned piano roll. For a fair comparison, we condition this model with the performer IDs and provide a *position roll* that encodes the note-wise position information. (A position roll is similar to a piano roll, but the values decrease linearly from 1 to 0, from the beginning of a note to its end.) As can be seen from Fig. 6(a) and (b), our proposed model produces smoother contours and clearer harmonic structures, especially on the high frequency end, while the baseline model generates sharper yet noisier results. Table 2 shows the final MSE between the synthesized mel spectrograms and the ground truths. We can see that our proposed model achieves a lower MSE than the baseline model on both datasets. Finally, due to the reduced temporal resolution of a mel spectrogram compared to that of a waveform, our proposed model is faster in training than the baseline model. Audio samples can be found on our project website.[1]

Table 1: Results of the subjective listening test. The mean opinion scores (MOS) and 95% confidence intervals are reported.

	Violin				Piano
	Pitch accuracy	Timbre	Noise level	Overall	Overall
Hifi-GAN baseline	4.02 ± 0.31	3.13 ± 0.26	2.51 ± 0.29	2.57 ± 0.22	1.49 ± 0.17
Deep Performer (ours)	4.22 ± 0.30	3.26 ± 0.30	2.67 ± 0.31	2.58 ± 0.21	2.17 ± 0.24
- without note-wise positional encoding	4.13 ± 0.29	3.24 ± 0.27	2.52 ± 0.29	2.61 ± 0.23	2.37 ± 0.23
- without performer embedding	3.05 ± 0.52	2.54 ± 0.42	2.04 ± 0.31	2.01 ± 0.25	2.26 ± 0.25
- without encoder (using piano roll input)	4.30 ± 0.36	2.91 ± 0.28	2.39 ± 0.28	2.22 ± 0.18	1.43 ± 0.16

Table 2: Comparisons of the final MSE between the synthesized mel spectrograms and the ground truths, in log scales.

	Violin	Piano
Hifi-GAN baseline	0.892	0.722
Deep Performer (ours)	0.700	0.436
- without note-wise positional encoding	0.700	0.433
- without performer embedding	1.030	0.523
- without encoder (using piano roll input)	0.844	0.621

Fig. 6: Examples of the mel spectrograms, in log scale, synthesized by (a) the baseline model, (b) our proposed synthesis model, and (d) our proposed synthesis model without the note-wise positional encoding. (c) and (e) show the waveforms for (b) and (d), respectively. (f) shows the input score.

4.3. Subjective listening test

To further evaluate our proposed system, we conduct a subjective listening test with 15 participants recruited from our social networks, where 14 of them plays a musical instrument. We randomly choose 5 musical scores from each dataset and synthesize them with different models. The participants are instructed to rate the synthesized audios in a 5-point Likert scale in terms of pitch accuracy, timbre and noise level as well as the overall quality. We report the results in Table 1. We can see that our proposed model significantly outperforms the baseline model on the piano dataset and achieves comparable performance to the baseline on the violin dataset.

4.4. Ablation study

To measure the contributions of different components of the proposed model, we consider three ablated versions of our model. The first removes the note-wise positional encoding. The second removes the performer embedding. The third removes the encoder and uses piano rolls and position rolls (see Section 4.2) as the inputs to the decoder, while keeping the performer embedding. As we can see from Fig. 6(b)–(e), note-wise positional encoding help the model produce clearer note transitions and a more realistic waveform envelope (see the highlighted regions). We also report in Tables 1 and 2 the results for these ablated models. We can see that the performer embedding significantly improves the quality across all criteria. While we show above the effectiveness of the note-wise positional encoding, its impact does not reach statistical significance in our subjective listening test, possibly overshadowed by the artifacts produced by the models. Finally, including an encoder network improves the quality significantly, suggesting that the encoder can learn a more effective representation of the score as compared to the piano roll representation.

5. CONCLUSION

We presented a novel three-stage system for sythesizing natural music performance from unaligned musical scores. We proposed the polyphonic mixer for aligning the encoder and decoder with polyphonic inputs. In addition, we also proposed the note-wise positional encoding for providing a fined-grained conditioning to the synthesis model. Through the subjective listening test, we show that our proposed model significantly outperforms the baseline model on the piano dataset and achieves competitive quality against the baseline on the violin dataset. For future work, we plan to utilize the articulation marks and ornaments on scores to better model playing techniques [29, 30], disentangle the timbre from room acoustics to enhance controllability [6], and incorporate adversarial losses [31, 32] to improve the sharpness of the results.

6. ACKNOWLEDGMENTS

7. REFERENCES

[1] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, "Neural audio synthesis of musical notes with WaveNet autoencoders," in *ICML*, 2017.

[2] A. Défossez, N. Zeghidour, N. Usunier, L. Bottou, and F. Bach, "SING: Symbol-to-instrument neural generator," in *NeurIPS*, 2018.

[3] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial neural audio synthesis," in *ICLR*, 2019.

[4] R. Manzelli, V. Thakkar, A. Siahkamari, and B. Kulis, "Conditioning deep generative raw audio models for structured automatic music," in *ISMIR*, 2018.

[5] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *ICLR*, 2019.

[6] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *ICLR*, 2020.

[7] B. Hayes, C. Saitis, and G. Fazekas, "Neural waveshaping synthesis," in *ISMIR*, 2021.

[8] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. Liu, "DeepSinger: Singing voice synthesis with data mined from the web," in *KDD*, 2020.

[9] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Computing and Applications*, vol. 32, pp. 955–967, 2020.

[10] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *ICASSP*, 2018.

[11] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *ICASSP*, 2019.

[12] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, and A. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *NeurIPS*, 2019.

[13] J. B. Jungil Kong, Jaehyeon Kim, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *NeurIPS*, 2010.

[14] B. Wang and Y.-H. Yang, "PerformanceNet: Score-to-audio music generation with multi-band convolutional residual network," in *AAAI*, 2019.

[15] F. Schimbinschi, C. Walder, S. M. Erfani, and J. Bailey, "SynthNet: Learning to synthesize music end-to-end," in *IJCAI*, 2019.

[16] J. W. Kim, R. Bittner, A. Kumar, and J. P. Bello, "Neural music synthesis for flexible timbre control," in *ICASSP*, 2019.

[17] Y. Wu, E. Manilow, Y. Deng, R. Swavely, K. Kastner, T. Cooijmans, A. Courville, C.-Z. A. Huang, and J. Engel, "MIDI-DDSP: Detailed control of musical performance via hierarchical modeling," in *ICLR*, 2022.

[18] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," *arXiv preprint arXiv:2106.15561*, 2021.

[19] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *INTERSPEECH*, 2017.

[20] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, robust and controllable text to speech," in *NeurIPS*, 2019.

[21] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei, D. Su, and D. Yu, "DurIAN: Duration informed attention network for multimodal synthesis," in *INTERSPEECH*, 2020.

[22] "FluidSynth," `https://www.fluidsynth.org/`.

[23] "Musescore general soundfont," `https://musescore.org/en/handbook/3/soundfonts-and-sfz-files`.

[24] "International Piano-e-competition," `https://www.piano-e-competition.com/`.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[26] C.-M. Chien, J.-H. Lin, C.-Y. Huang, P.-C. Hsu, and H.-Y. Lee, "Investigating on incorporating pretrained and learnable speaker representations for multi-speaker multi-style text-to-speech," in *ICASSP*, 2021.

[27] C. Raffel and D. P. W. Ellis, "Intuitive analysis, creation and manipulation of MIDI data with pretty_midi," in *ISMIR Late-Breaking Demos*, 2014.

[28] H.-W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, "MusPy: A toolkit for symbolic music generation," in *ISMIR*, 2020.

[29] C.-H. Yang, P.-C. Li, A. W. Y. Su, L. Su, and Y.-H. Yang, "Automatic violin synthesis using expressive musical term features," in *DAFx*, 2016.

[30] C.-C. Shih, P.-C. Li, Y.-J. Lin, Y.-L. Wang, A. W. Y. Su, L. Su, and Y.-H. Yang, "Analysis and synthesis of the violin playing style of Heifetz and Oistrakh," in *DAFx*, 2017.

[31] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.

[32] J. Yang, J.-S. Bae, T. Bak, Y. Kim, and H.-Y. Cho, "GANSpeech: Adversarial training for high-fidelity multi-speaker speech synthesis," in *INTERSPEECH*, 2021.