GANSTRUMENT: ADVERSARIAL INSTRUMENT SOUND SYNTHESIS WITH PITCH-INVARIANT INSTANCE CONDITIONING

Gaku Narita, Junichi Shimizu, and Taketo Akama

Sony Computer Science Laboratories, Tokyo, Japan

ABSTRACT

We propose GANStrument, a generative adversarial model for instrument sound synthesis. Given a one-shot sound as input, it is able to generate pitched instrument sounds that reflect the timbre of the input within an interactive time. By exploiting instance conditioning, GANStrument achieves better fidelity and diversity of synthesized sounds and generalization ability to various inputs. In addition, we introduce an adversarial training scheme for a pitch-invariant feature extractor that significantly improves the pitch accuracy and timbre consistency. Experimental results show that GANStrument outperforms strong baselines that do not use instance conditioning in terms of generation quality and input editability. Qualitative examples are available online[1].

Index Terms— neural synthesizer, generative adversarial networks, adversarial feature extraction

1. INTRODUCTION

Since the advent of computers, many musicians and researchers have explored ways to generate music with computers. There are two main approaches: direct synthesis of music sounds including melody and accompaniment, and single note synthesis followed by playing symbolic music like MIDI. The former enables end-to-end music synthesis, but has low controllability of generation. The latter enables MIDI and timbre to be independently controlled and is compatible with production flows in the music industry. In this paper, we tackled instrument sound synthesis for the latter approach.

Realistic instrument sounds are typically synthesized with samplers that utilize recorded one-shot sounds. Although arbitrary sound materials can be exploited, it is difficult to synthesize a completely new timbre or intelligently combine multiple sounds. In contrast, recently reported deep generative models for audio synthesis [1, 2, 3, 4, 5, 6] have the potential to generate and mix a variety of timbres by exploring the latent space. Our aim is to design a neural synthesizer that combines the flexibility of samplers with the generative power of deep networks, thereby enabling users to freely control the timbre by leveraging existing sound materials. For practical use, it needs to not only be generalized to a variety of inputs but also be able to generate high-quality audio with accurate pitch within an interactive time.

Towards this end, we present GANStrument, a novel neural instrument sound synthesizer. To enable the model to accept various inputs, we focus on instance conditioning [7], a new GAN training scheme for conditioning a model on input features. In addition, we present a pitch-invariant feature extractor based on adversarial training that disentangles the latent space and significantly improves pitch accuracy and timbre consistency. Furthermore, use of the modern GAN architecture, parallel sampling with spectrogram representation, and carefully designed audio inversion enables high-quality audio to be generated within an interactive time.
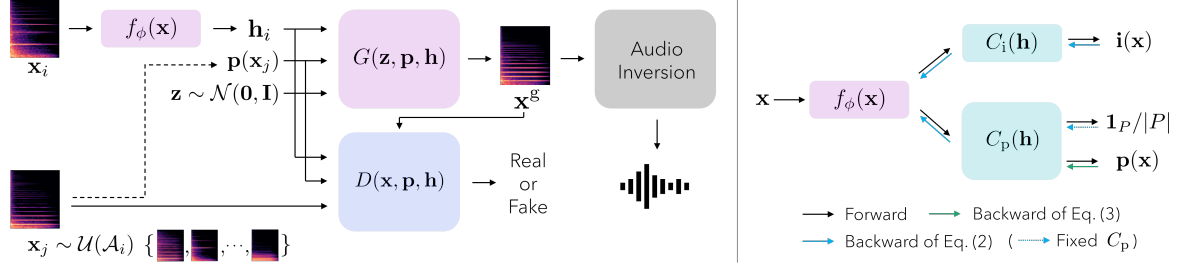
2. METHOD

As depicted in Fig. 1, the input waveform is first transformed into a mel-spectrogram $\mathbf{x}_i$, and its feature $\mathbf{h}_i$ is extracted with the feature extractor f_ϕ. It is fed into the generator G together with pitch $\mathbf{p}$ and noise $\mathbf{z}$ to synthesize a mel-spectrogram $\mathbf{x}^g$, which is transformed into a waveform by optimization-based audio inversion. Feature extractor f_ϕ is first trained by incorporating the pitch-adversarial loss into a standard classification loss. Using a frozen feature extractor, we jointly train the generator G and discriminator D with input neighborhoods as real samples.

2.1. Instance conditioning

Class-conditional GANs partition the entire data distribution into multiple distributions without overlap. In contrast, instance-conditioned GAN [7] partitions the entire data distribution into many overlapping local distributions and thereby model a complex distribution. By conditioning both the generator and discriminator with instance feature $\mathbf{h}_i = f_\phi(\mathbf{x}_i)$ and pitch $\mathbf{p}$, we model a local distribution of instance neighborhood $p(\mathbf{x}|\mathbf{h}_i, \mathbf{p})$ and represent the entire data distribution $p(\mathbf{x})$ as a mixture of these distributions: $\sum_{\mathbf{h}_i} \sum_{\mathbf{p}} p(\mathbf{x}|\mathbf{h}_i, \mathbf{p})p(\mathbf{p}|\mathbf{h}_i)p(\mathbf{h}_i)$.

We follow the training procedure of Casanova et al. [7]. For input $\mathbf{x}_i$, let $\mathcal{A}_i$ be the L_2-based k nearest neighbors of $\mathbf{x}_i$ over the feature space defined by f_ϕ. As shown in Fig. 1, we sample neighborhood data point $\mathbf{x}_j$ from the uniform distribution $\mathcal{U}(\mathcal{A}_i)$. Then $\mathbf{x}_j$ is used as a real sample together with a generated sample $\mathbf{x}^g$ to train the discriminator D, and its corresponding pitch $\mathbf{p}(\mathbf{x}_j)$ is fed into both the generator G and discriminator D for conditioning. Formally, we jointly

---

Fig. 1. Overview of GANStrument. Left side shows training and inference pipeline of generative model. Right side depicts adversarial training scheme of feature extractor.

optimize G and D using the following min-max game:

$$\min_G \max_D \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}), \mathbf{x}_j \sim \mathcal{U}(\mathcal{A}_i)}[\log D(\mathbf{x}_j, \mathbf{p}(\mathbf{x}_j), \mathbf{h}_i)] +$$
$$\mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}), \mathbf{z} \sim p(\mathbf{z})}[\log(1 - D(G(\mathbf{z}, \mathbf{p}(\mathbf{x}_j), \mathbf{h}_i), \mathbf{p}(\mathbf{x}_j), \mathbf{h}_i))]. \quad (1)$$

2.2. Pitch-invariant feature extractor

The simplest way to obtain feature extractor f_ϕ is to train a classifier using labeled data. In our case, for example, we can train an instrument identity classifier on the NSynth dataset [1]. However, features extracted by this classifier contain not only timbre but also pitch information, resulting in a decrease of pitch accuracy, as shown in Sec. 3.4. This is because the generator and discriminator confuse the specified pitch $\mathbf{p}$ with remaining pitch information in $\mathbf{h}$, and the training process starts oscillating.

To solve this problem, we propose a pitch-invariant feature extractor based on an adversarial training scheme that enables disentanglement of timbre and pitch in the latent space. Our training scheme is inspired by previous work such as domain adaptation [8], image manipulation [9], and music domain transfer [10], in which adversarial training was introduced to the bottleneck features. Let $C_i(\mathbf{h})$ and $C_p(\mathbf{h})$ be shallow MLPs that predict instrument identity $\mathbf{i}(\mathbf{x})$ and pitch $\mathbf{p}(\mathbf{x})$, respectively, given a feature $\mathbf{h} = f_\phi(\mathbf{x})$. As shown on the right side of Fig. 1, the following objectives are alternately optimized to obtain f_ϕ:

$$\min_{f_\phi, C_i} \text{CE}(\mathbf{i}(\mathbf{x}), C_i(f_\phi(\mathbf{x}))) + \lambda_{\text{adv}} \text{KL}(\frac{\mathbf{1}_P}{|P|} || C_p(f_\phi(\mathbf{x}))), \quad (2)$$

$$\min_{C_p} \text{CE}(\mathbf{p}(\mathbf{x}), C_p(f_\phi(\mathbf{x}))), \quad (3)$$

where $|P|$ is the number of pitches, $\mathbf{1}_P \in \mathbb{R}^{|P|}$ is the all-one vector, and CE and KL represent cross entropy and Kullback-Leibler divergence, respectively. The first term of Eq. (2) updates f_ϕ and C_i so that instrument identities can be correctly classified, while the second term makes it impossible to classify pitch given $\mathbf{h}$. Eq. (3) updates C_p to maximize the accuracy of pitch classification given $\mathbf{h}$. This adversarial training eventually produces instance feature $\mathbf{h}$, which contains little pitch information. In fact, we re-trained pitch classifier C_p with the frozen f_ϕ using Eq. (3), and the accuracy of pitch classification dropped from 17.4% to 2.6% with our adversarial training scheme while that of instrument identity classifi-

cation remained unchanged at 91.2%, meaning that feature $\mathbf{h}$ preserves timbre information.

2.3. Audio inversion

In the field of speech synthesis, learning-based vocoders have achieved high-quality audio synthesis [11, 12, 13]. However, several studies [14, 15, 16] suggested the difficulty of making neural vocoders generalized to a variety of timbre and pitch, which GANStrument is aimed at generating. On the other hand, MelNet [17] revealed that optimization-based audio inversion can synthesize a variety of audio including music with decent quality by using high-resolution mel-spectrograms (e.g., 256 bins).

Therefore, we use optimization-based audio inversion with high-resolution mel-spectrograms (512 bins). Mel-spectrogram inversion typically consists of a mel-to-linear frequency-scale conversion and phase restoration using the Griffin-Lim algorithm [18]. The frequency-scale conversion could be a bottleneck here because a non-negative least-squares problem $\min_{\mathbf{x}_{\text{lin}}} ||\mathbf{F}_{\text{mel}} \mathbf{x}_{\text{lin}} - \mathbf{x}_{\text{mel}}||^2$ s.t. $\mathbf{x}_{\text{lin}} \geq 0$ must be solved with the computationally demanding L-BFGS-B algorithm [19]. In place of L-BFGS-B, `torchaudio` [20] introduces the first-order gradient method with negative value clipping for faster iteration. However, it requires a sufficient number of iterations due to random initialization.

We propose a simple yet effective initialization scheme. First, we solve an unconstrained least-squares problem $\min_{\mathbf{x}_{\text{lin}}} ||\mathbf{F}_{\text{mel}} \mathbf{x}_{\text{lin}} - \mathbf{x}_{\text{mel}}||^2$ using the divide-and-conquer SVD because the mel filter bank $\mathbf{F}_{\text{mel}}$ is not well-conditioned. Next, the negative values of the solution are clipped, and the clipped solution is set as the initial value for the iterative method of the first-order gradient method. This initialization scheme reduces the number of iterations by a factor of 10.

3. EVALUATION

3.1. Experimental setup

We trained GANStrument on the NSynth dataset [1], a large-scale instrument sound dataset that includes rich annotations such as instrument categories, identities, and pitches. We extracted 88 pitches (MIDI notes 21–108) and used their first 1-s segments with amplitude normalization and exponential fade-out preprocessing. For evaluation, we also used single notes of Good-sounds [21], with the silence intervals trimmed, and used the same preprocessing as the NSynth dataset.

Table 1. Generation quality

conditioning	Nsynth (val set)		Good-sounds	
	FID↓	Pitch↑	FID↓	Pitch↑
pitch	490.1	0.831	1837.0	0.900
pitch + instrument	469.6	0.828	921.6	0.937
pitch + instance (ours)	**212.3**	**0.870**	**507.3**	**0.946**

To compute mel-spectrograms, we used an STFT with a Hann window, a 1024 window size, a 64 hop size, and a 2024 fft size. This was followed by mel-scale conversion with 512 filter banks, resulting in 512×256 mel-spectrograms.

We utilized the StyleGAN2 [22] architecture, a state-of-the-art image synthesis model, for the backbone and used a projection discriminator [23]. For feature extractor f_ϕ, we used an architecture that removes the final layer of the discriminator. We jointly trained the generator and discriminator using the ADAM optimizer with a learning rate of 2.5×10^{-3}, $(\beta_1, \beta_2) = (0.0, 0.99)$, and $\epsilon = 10^{-8}$ for 300k steps with a batch size of 16. For training stability, we exploited the training techniques described by Karras et al. [22] such as R_1 regularization and path length regularization. We used $k = 50$ for the neighborhood search.
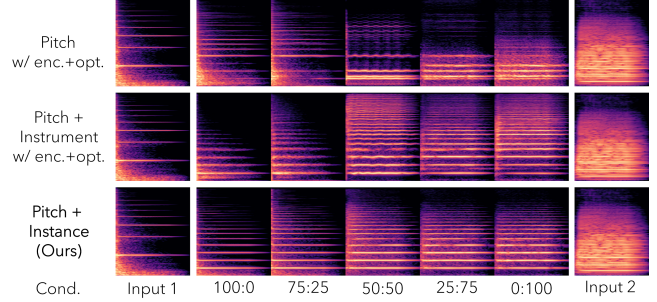
3.2. Generation quality

First, we evaluated the fidelity and diversity as well as pitch accuracy of the generated samples. To validate the proposed approach, we trained two class-conditional GANs as strong baselines: the first model $G_1(\mathbf{z}, \mathbf{p})$ was conditioned on pitch $\mathbf{p}$ and the other $G_2(\mathbf{z}, \mathbf{p}, \mathbf{c})$ was conditioned on both pitch $\mathbf{p}$ and instrument category $\mathbf{c}$ (using 11 NSynth instrument categories). For fair comparison, we used the same architecture and training parameters for these baselines as in Sec 3.1.

To evaluate the Fréchet inception distance (FID) and pitch accuracy, we trained both the instrument category classifier (as an FID feature extractor) and the pitch classifier on the NSynth dataset. The architecture of these classifiers was a slightly modified version of the discriminator. They respectively achieved accuracies of 74.3% and 93.2% (against the test set). For evaluation on Good-sounds, we used the same distributions of pitch and category as in the dataset for fair comparison.

Table 1 shows that GANStrument was superior to the baselines on both datasets, suggesting that GANStrument has not only the ability to model the distribution of training data but also the ability of generalization.

3.3. Editability

Next, we evaluated the editability of the input sounds. To evaluate the faithfulness of reconstruction, we measured the mean squared error (MSE) and pitch accuracy of the synthesized samples. MSE was computed on the feature space defined by the FID feature extractor. To evaluate the ability of exploration in the latent space, we randomly chose two inputs from the dataset and interpolated the corresponding la-

Fig. 2. Examples of interpolation in the latent space.[1]

tent variables using a ratio sampled from uniform distribution $\mathcal{U}(0, 1)$ to generate interpolated samples. We computed the FID between the input and interpolated samples. We randomly chose a conditioning pitch from the 88 pitches.

The baseline models need to invert the inputs into the latent space for editing. Typical approaches to GAN inversion can be categorized into learning-based, optimization-based, and a hybrid of the two [24]. In our experiments, we used learning-based and hybrid approaches, which we found work well. We trained encoders $E_1(\mathbf{x})$ and $E_2(\mathbf{x})$ for the baselines with objectives $\min_{E_1} ||\mathbf{x} - G_1(E_1(\mathbf{x}), \mathbf{p}(\mathbf{x}))||_2^2 + \lambda_{\mathbf{z}}||E_1(\mathbf{x})||_2^2$ and $\min_{E_2} ||\mathbf{x} - G_2(E_2(\mathbf{x}), \mathbf{p}(\mathbf{x}), \mathbf{c}(\mathbf{x}))||_2^2 + \lambda_{\mathbf{z}}||E_2(\mathbf{x})||_2^2$, respectively. The second term is regularization for $\mathbf{z}$ to follow a standard normal distribution, which we found to be essential for the following optimization. In the hybrid approach, we initialized latent variables with $\mathbf{z} = E_{\{1,2\}}(\mathbf{x})$ and minimized L_2 loss with respect to $\mathbf{z}$.

The middle portion of Table 2 shows that the baselines tended to fail in reconstructing the inputs and to sacrifice pitch accuracy, especially for Good-sounds, because they prioritize minimizing the reconstruction error, whereas GANStrument successfully reconstructed the inputs for both seen and unseen datasets. The right side of Table 2 shows that the interpolated samples of the baselines produced a significant decrease in pitch accuracy, which suggests that the interpolated latents could deviate from the data manifold. Our model, in contrast, had better FID and pitch accuracy, demonstrating that it can generate high fidelity samples with accurate pitch by exploring the latent space.

Fig. 2 shows qualitative examples of the interpolation in the latent space. Their inputs were keyboard and brass sounds of the NSynth dataset and noise vectors $\mathbf{z}$ were fixed. The baselines struggled with inverting the inputs and completely failed to mix two sounds. In contrast, GANStrument smoothly interpolated two timbres with accurate pitch.
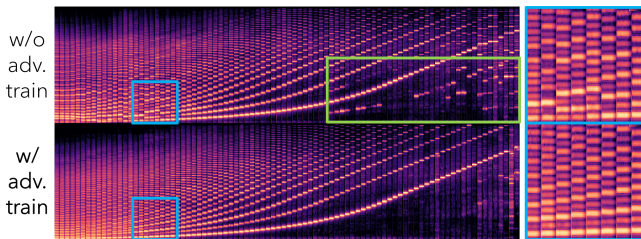
3.4. Ablation study

Next, we conducted an ablation study to evaluate the effectiveness of the proposed pitch-invariant feature extractor. For comparison, we trained an instrument identity classifier as a feature extractor f_ϕ using only the first term of Eq. (2). Table 3 shows that our approach significantly improved pitch accuracy. Fig. 3 shows mel-spectrograms of 88 pitches generated with the input of a saxophone sound of the Good-sounds dataset and a fixed noise $\mathbf{z}$. The feature extractor without ad-

Table 2. Editability

conditioning	inversion	reconstruction				interpolation			
		Nsynth (val set)		Good-sounds		Nsynth (val set)		Good-sounds	
		MSE↓	Pitch↑	MSE↓	Pitch↑	FID↓	Pitch↑	FID↓	Pitch↑
pitch	enc.	6.53	0.669	8.23	0.384	1451.7	0.296	2298.5	0.251
	enc. + opt.	6.34	0.629	8.44	0.126	1183.0	0.314	2292.4	0.214
pitch + instrument	enc.	4.32	0.793	5.09	0.655	709.8	0.594	679.3	0.585
	enc. + opt.	3.49	0.778	**3.12**	0.167	601.2	0.534	610.0	0.442
pitch + instance (ours)	-	**1.79**	**0.904**	3.28	**0.944**	**252.2**	**0.883**	**477.4**	**0.883**

Table 3. Ablation study: feature extractor

feature extractor	Nsynth (train set)		Nsynth (val set)	
	FID↓	Pitch↑	FID↓	Pitch↑
w/o adv. training	95.3	0.731	**191.4**	0.757
w/ adv. training	**90.4**	**0.834**	212.3	**0.870**

Fig. 3. Ablation study: difference between feature extractor without and with adversarial training.[1]
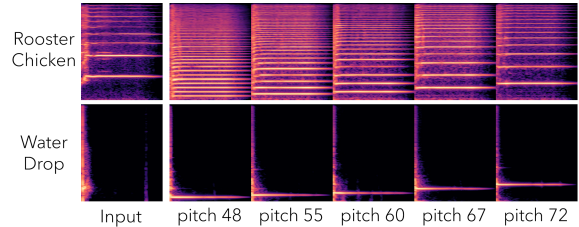
versarial training produced inaccurate pitches, especially in higher tones as shown in the green box, as well as timbre inconsistency, as shown in the blue box. The pitch-invariant feature extractor, in contrast, produced stable pitches with timbre consistency.

3.5. Non-instrument sound inputs

Fig. 4 shows qualitative results with the input of non-instrument sounds such as a rooster and dropping water. The synthesized sounds reflected the input timbres and produced stable pitch like musical instruments. These results demonstrate that GANStrument has generalization ability to non-instrument sounds to some extent and is able to exploit a variety of sound materials to design the timbre as the traditional samplers do. Note that the additional examples are available online[1].

3.6. Timing

Finally, we measured the generation timing on a middle-range CPU (Intel Core i7-7800X, 3.50 GHz). The total time was 1.62 s, where the inference of the feature extractor f_ϕ and generator G took 0.31 and 0.35 s, respectively, and mel-to-linear scale conversion 0.60 s, Griffin-Lim algorithm 0.36 s. These results show that the improved mel-to-linear scale conversion described in Sec. 2.3 plays an important role in interactive generation.

Fig. 4. Examples of non-instrument sound inputs.[1]

4. RELATED WORK

NSynth [1] uses a WaveNet [25]-based autoencoder to directly synthesize the waveforms of instrument sounds. While it is capable of inference with a trained encoder, autoregressive sampling makes generation slow and prone to artifacts. GANSynth [2] improves generation speed and quality by using an image synthesis model and a spectrogram with phase information. However, it does not accept inputs, making it difficult to explore desired timbre in a complex latent space. Luo et al. [3] proposed disentangling timbre and pitch using a Gaussian mixture VAE, but the simple architecture and autoencoder-based training make audio quality insufficient.

DDSP [5] and its subsequent work [6] achieve fast and interpretable generation by incorporating additive synthesis and wavetable synthesis into autoencoders. However, their inputs should have clear pitch and the generated timbre is basically limited to a combination of integral multiples of the fundamental frequency [5]. Leveraging the domain knowledge like these studies is complementary to our work and left for future work to further improve our model.

5. CONCLUSION

Our novel neural synthesizer, GANStrument, generates pitched instrument sounds reflecting one-shot input timbre within an interactive time. It incorporates two key features: 1) instance conditioning, resulting in better generation quality and generalization ability to various inputs and 2) pitch-invariant feature extraction based on adversarial training, resulting in significantly improved pitch accuracy and timbre consistency. Experimental results demonstrated the effectiveness of this approach. We believe that GANStrument will enable users to generate novel instrument sounds as well as freely explore the desired timbre by utilizing a variety of sound materials.

6. REFERENCES

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

[2] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," in *Proc. ICLR*, 2018.

[3] Y. Luo, K. Agres, and D. Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders," in *Proc. ISMIR*, 2019.

[4] J. Nistal, S. Lattner, and G. Richard, "Drumgan: Synthesis of drum sounds with timbral feature conditioning using generative adversarial networks," in *Proc. ISMIR*, 2020.

[5] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," in *Proc. ICLR*, 2020.

[6] S. Shan, L. Hantrakul, J. Chen, M. Avent, and D. Trevelyan, "Differentiable wavetable synthesis," in *Proc. ICASSP*, 2022.

[7] A. Casanova, M. Careil, J. Verbeek, M. Drozdzal, and A. Romero-Soriano, "Instance-conditioned gan," in *Proc. NeurIPS*, 2021.

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[9] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato, "Fader networks: Manipulating images by sliding attributes," in *Proc. NeurIPS*, 2017.

[10] A. P. Noam Mor, Lior Wold and Y. Taigman, "A universal music translation network," in *Proc. ICLR*, 2019.

[11] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, et al., "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. ICASSP*, 2018.

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

[13] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeurIPS*, 2020.

[14] S.-g. Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, "Bigvgan: A universal neural vocoder with large-scale training," *preprint arXiv:2206.04658*, 2022.

[15] D.-Y. Wu, W.-Y. Hsiao, F.-R. Yang, O. Friedman, W. Jackson, S. Bruzenak, Y.-W. Liu, and Y.-H. Yang, "Ddsp-based singing vocoders: A new subtractive-based synthesizer and a comprehensive evaluation," in *Proc. ISMIR*, 2022.

[16] J. Lee, H. Lim, C. Lee, I. Jang, and H.-G. Kang, "Adversarial audio synthesis using a harmonic-percussive discriminator," in *Proc. ICASSP*, 2022.

[17] S. Vasquez and M. Lewis, "Melnet: A generative model for audio in the frequency domain," *preprint arXiv:1906.01083*, 2019.

[18] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[19] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on scientific computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[20] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, "Torchaudio: Building blocks for audio and speech processing," *preprint arXiv:2110.15018*, 2021.

[21] O. Romani Picas, H. Parra Rodriguez, D. Dabiri, H. Tokuda, W. Hariya, K. Oishi, and X. Serra, "A real-time system for measuring sound goodness in instrumental sounds," in *Audio Engineering Society Convention 138*, 2015.

[22] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proc. CVPR*, 2020.

[23] T. Miyato and M. Koyama, "cgans with projection discriminator," in *Proc. ICLR*, 2018.

[24] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, "Gan inversion: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[25] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *preprint arXiv:1609.03499*, 2016.