INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Unconditional Audio Generation with Generative Adversarial Networks and Cycle Regularization

Jen-Yu Liu[1], *Yu-Hua Chen*[1,2], *Yin-Cheng Yeh*[1], *Yi-Hsuan Yang*[1,2]

[1]Taiwan AI Labs, Taipei, Taiwan
[2]Academia Sinica, Taipei, Taiwan

jyliu@ailabs.tw, cloud60138@citi.sinica.edu.tw, yyeh@ailabs.tw, yang@citi.sinica.edu.tw

Abstract

In a recent paper, we have presented a generative adversarial network (GAN)-based model for unconditional generation of the mel-spectrograms of singing voices. As the generator of the model is designed to take a variable-length sequence of noise vectors as input, it can generate mel-spectrograms of variable length. However, our previous listening test shows that the quality of the generated audio leaves room for improvement. The present paper extends and expands that previous work in the following aspects. First, we employ a hierarchical architecture in the generator to induce some structure in the temporal dimension. Second, we introduce a cycle regularization mechanism to the generator to avoid mode collapse. Third, we evaluate the performance of the new model not only for generating singing voices, but also for generating speech voices. Evaluation result shows that new model outperforms the prior one both objectively and subjectively. We also employ the model to unconditionally generate sequences of piano and violin music and find the result promising. Audio examples, as well as the code for implementing our model, will be publicly available online upon paper publication.

1. Introduction

In the recent development of deep learning, unconditional generation of images has been a popular research topic [1, 2, 3], either for its own artistic value or for being used as a base model for developing models with more fine-grained conditions (e.g., [4]). Unconditional music generation in the symbolic domain, which aims at generating sequences of musical notes in a symbolic format such as the piano roll, has also become an active research topic lately [5, 6, 7].[1]

Unconditional generation of raw audio waveforms, or its time-frequency representations, has received growing attention in recent years as well. A notable example is the *Zero Resource Speech Challenge* ("TTS without T," or text-to-speech without text) organized in some editions of the INTERSPEECH conference [15]. Given a collection of raw audio without text or phoneme labels as the training data, the participants have to discover the subword units of speech [16, 17, 18] in an unsupervised way so as to synthesize novel utterances from novel speakers. While being an interesting and meaningful unconditional audio generation task, the task setup is fairly speech-specific and accordingly such TTS without T models are not readily applicable to generate other types of audio signals, such as instrumental or environmental sounds.

There have been attempts to unconditional generation of general audio, e.g., [19, 20, 21]. However, they all use an autoregressive approach that takes only a noise vector as input and generates sequentially samples of an audio signal (e.g., timesteps or spectral frames), one sample at a time, which might not be efficient at inference time. We are motivated to explore alternative model architectures.

We present such an attempt in a recent work for unconditional singing voice generation [22], aiming to generate improvised singing voices without using not only the phoneme labels (i.e., the lyrics) but also the pitch labels (i.e., the singing melody), in both model training and inference time. It is based on a generative adversarial network (GAN) [1] where the generator takes a variable-length sequence of noise vectors as input, instead of just one noise vector as done in [19, 20, 21]. The mission of the generator is to convert the sequence of input noise vectors to a mel-spectrogram of the corresponding length, with each input vector corresponds to a certain length at the output. Such an architecture therefore has the potential to generate a variable-length audio in a more efficient way. While being an interesting attempt, the user study reported in the previous work (i.e., [22]) suggests that the model presented there is not powerful enough to generated samples with satisfying perceptual quality. Moreover, whether such a GAN-based model can be applied to audio other than singing voices remains unexplored.

We presented in this paper an improved version of our prior model, using a similar GAN architecture that takes multiple noise vectors as input. But, the new generator now employs a hierarchical structure to govern the temporal coherence of the generated samples. Moreover, we regularize the model training process by enforcing cycle consistency between each input noise vector and the corresponding segment in the output mel-spectrogram. We validate both objectively and subjectively that the new model greatly outperforms the prior one, for unconditional generation of not only singing voices but also speech.

We refer to the proposed model architecture as UNAGAN, or unconditional audio generation with GAN. The code and trained models are available at `https://github.com/ciaua/unagan.git`. Samples of the generated sounds of singing, speech, as well as instruments are also provided.

2. Problem Formulation

In the literature, a common approach to audio generation is to first generate acoustic features, such as the mel-spectrograms, and then pass them to a vocoder to generate the corresponding audio waveforms [9, 20]. We follow this practice in this paper, and we focus on generating the mel-spectrograms.

The problem formulation adopted in [19, 20, 21] can be generally described as follows. Given an input noise vector $\mathbf{z} \in \mathbb{R}^N$, where N denotes the length of the vector, they build

[1] Following the convention in the literature, we define unconditional generation as a task that aims at generating things from scratch, i.e., taking nothing but random noises as the input. In contrast, a conditional generation model takes additional input such as class labels [8], text [9], pitch labels [10, 11], or reference audio [12, 13, 14].

http://dx.doi.org/10.21437/Interspeech.2020-1137

a generator $G(\cdot)$ so as to convert the input vector to a sequence of acoustic features (e.g., the mel-spectrograms), $\widehat{\mathbf{X}} \equiv G(\mathbf{z}) \in \mathbb{R}^{K\times T}$, in an auto-regressive manner. Here, K denotes the length of each acoustic feature vector $\widehat{\mathbf{x}}_t$, and T the temporal length of the sequence to be generated.

We intend to use instead a sequence of random noise vectors as the input, namely, $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{T'}]$, where each term in each vector $\mathbf{z}_t$ is sampled from a Gaussian distribution with zero mean and unit variance. And, the length of the input sequence, namely T', is proportional to the length of the target output sequence, namely T. In other words, each $\mathbf{z}_t$ has a direct influence over one (i.e., when $T' = T$) or a few (when $T' < T$) samples of the target output $\widehat{\mathbf{X}}$.

3. Model

We adopt the GAN framework for unconditional generation of audio signals of arbitrary length. Similar to [22], we find that the boundary-equilibrium GAN (BEGAN) [23] works better than other types of GANs for this task, so we also use it here. To further improve the generation quality, we propose a number of changes for the generator and the discriminator, as described below. We start with a brief introduction of BEGAN first.

3.1. Boundary-Equilibrium GAN

In BEGAN [23], the loss functions l_D and l_G for the discriminator $D(\cdot)$ and the generator $G(\cdot)$ are respectively:

$$l_D = L(\mathbf{X}) - \tau_s L(G(\mathbf{Z}))\,, \qquad (1)$$

$$l_G = L(G(\mathbf{Z}))\,, \qquad (2)$$

where $\mathbf{X} \in \mathbb{R}^{K\times T}$ denotes a sequence of acoustic features from a real audio signal sampled from the training data, and $L(\cdot)$ is a function that measures how well the discriminator reconstructs its input. Specifically,

$$L(\mathbf{M}) = \frac{1}{WT}\sum_{w,t} |D(\mathbf{M})_{w,t} - \mathbf{M}_{w,t}|\,, \qquad (3)$$

for an arbitrary $W \times T$ matrix $\mathbf{M}$, where $M_{w,t}$ denotes the (w,t)-th element of a matrix $\mathbf{M}$ (and similarly for $D(\mathbf{M})_{w,t}$). Moreover, the variable τ_s in Eq. (1) is introduced by BEGAN to balance the power of $D(\cdot)$ and $G(\cdot)$ during the learning process. It is dynamically set to be $\tau_{s+1} = \tau_s + \beta(\gamma L(\mathbf{X}) - L(G(\mathbf{Z})))$, for each training step s, with $\tau_s \in [0,1]$. And, β and γ are manually-set hyperparameters. From Eqs. (1) and (2), we see that $D(\cdot)$ and $G(\cdot)$ have contradicting goals, giving rise to the name of adversarial training. Once trained, only $G(\cdot)$ is used at the inference time for generating new content from scratch.

3.2. Hierarchical Structure in the Generator

In our previous work [22], each spectral frame $\widehat{\mathbf{x}}_t$ has its own input noise $\mathbf{z}_t$, i.e., $T' = T$. This allows the generator to generate a sequence of arbitrary length. However, this could also make it difficult for the generator to generate coherent sequences of acoustic features while respecting the input noises, because $[\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{T'}]$ are sampled independently. As a result, the generated sounds tend to be fragmented and do not form complete pronunciations.

To alleviate the aforementioned issue, we make two changes to the architecture of the generator. First, we reduce the number of input noise vectors by a downsampling factor of S. That is, to generate a sequence of length T, we use $T' = \lceil T/S \rceil$ input noise vectors. Due to this change, it is necessary to upsample the output of the intermediate layers of the generator. This incurs the second change to the generator architecture—instead of a pure upsampling operation, we adopt a hierarchical architecture where the generator generates acoustic features from coarse ones to more refined ones, as shown in Figure 1. This kind of processing has been shown effective in several generation models [3, 21].

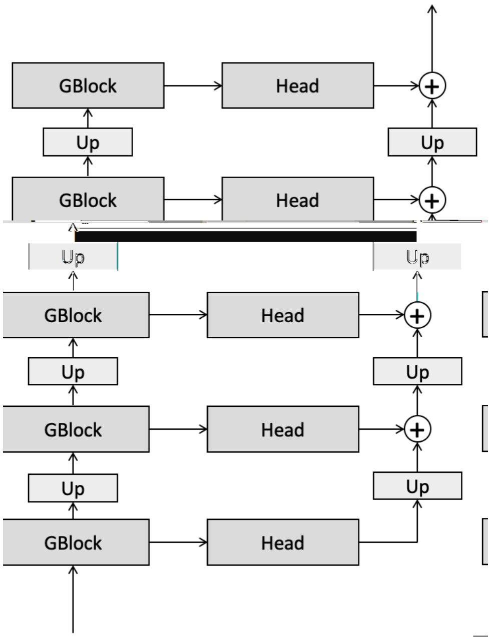

Figure 1: *The hierarchical architecture of the proposed generator. Here, 'GBlock' is a stack of convolution layers and gate recurrent unit layers with skip connections; 'Head' is a convolution layer; and 'Up' is a nearest-neighbor upsampling operation with scaling factor of 2.*

As depicted in Figure 1, in the proposed generator we still employ the 'GBlock' proposed in our prior work [22]. It entails a stack of gate recurrent unit (GRU) layers [24, 25] and grouped convolution layers. Therefore, the major changes are associated with the use of the upsampling layers ('Up'), and the auxiliary convolution layers ('Head') in the skip connections.

3.3. Cycle Regularization in the Generator

Another issue of the previous model [22] is that it seems to suffer from the "mode collapse" issue [26, 27] and generate sounds that are not diverse enough. We propose using a cycle regularization mechanism in the generator to alleviate this.

In cycleGAN [28], a cycle-consistency constraint is enforced on two generators of two different domains, so that the content generated by one generator and the content generated by the other generator have certain degree of correspondence. In our case, one domain is about the acoustic features $\mathbf{x}_t$ and the other domain the input noises $\mathbf{z}_t$. In other words, the cycle consistency is established between the input noise and the target output. We note that Ulyanov *et al.* [29] have used this strategy as an alternative way for generation without the adversarial term.

Specifically, besides the GAN loss l_G, we add the following cycle regularization term to the loss function of the generator:

$$l_C = |E(G(\mathbf{Z})) - \mathbf{Z}|_1 + |G(E(\mathbf{X})) - \mathbf{X}|_1\,, \qquad (4)$$

Table 1: *The architecture of the discriminator. All convolution layers have kernel size 3. 'Ch.' denotes the number of channels.*

	Details	**Ch.**	**Stride**	**Dilation**
2D Block 1	2D Conv Batch norm Leaky ReLU	4	(2, 1)	(1, 2)
2D Block 2	*	16	(2, 1)	(1, 4)
2D Block 3	*	64	(2, 1)	(1, 8)
Flatten				
1D Block 1	1D Conv Batch norm Leaky ReLU	512	1	1
1D Block 2	*	512	1	16
1D Block 3	*	512	1	32
1D Block 4	*	512	1	64
1D Block 5	*	512	1	128
Output	1D Conv	80	1	1

where E, an encoder, predicts $\mathbf{Z}$ from the output of G. In our implementation, E consists of 2D and 1D convolution layers similar to D, but has downsampling operations to match the upsampling operations of G. The loss function of the generator therefore becomes

$$l'_G = l_G + \lambda\, l_C\,, \tag{5}$$

where λ is a tunable hyperparameter.

If the cycle regularization term is met perfectly, it will enforce a one-one and onto mapping between the domain of noises and the domain of acoustic features. In other words, for every acoustic feature vector $\mathbf{x}$, there is a vector $\mathbf{z}$ such that $G(\mathbf{z}) = \mathbf{x}$. This also means that every mode is covered, avoiding mode collapse. However, as the cycle term is not met perfectly in reality, mode collapse may still happen, yet hopefully to a less degree.

Another benefit of this regularization is that it is general-purpose: it can be augmented to any generators and discriminators without changing the architectures.

3.4. The Discriminator

In [22], the discriminator also contains GRUs. We have found that it is possible to use a more efficient discriminator that contains only convolution layers without compromising the performance. Specifically, we use 2D convolution layers followed by 1D convolution layers in this work, as shown in Table 1.

4. Evaluation

To demonstrate that the proposed model works for a diverse set of audio, we apply it to generating singing voices, speech, and solos of piano and violin, each using a different dataset. Examples of the generated sounds can be found in our GitHub repo. Below, we present the implementation details, as well as the objective and subjective studies that validate the effectiveness of the proposed model for generating singing and speech voices.

4.1. Datasets

We employ the following audio datasets in this work.

- **Speech**: We use the LJ Speech dataset [30], which contains 13,100 short audio clips of a single speaker reading passages from books.
- **Singing**: Following our previous work [22], for the singing part we use a collection of 17.4 hours of female voices singing in Jazz. The singing voices are obtained by applying a state-of-the-art blind source separation model [31] to the original songs which contains instrumental background music.
- **Piano**: We use the MAESTRO dataset [32], which contains 23 hours of virtuosic piano performance of classical music recoreded in several years. The MIDI part of the dataset has been increasingly used in symbolic-domain music generation [6]. We use however the audio recordings of the data, using only those from the year of 2004.
- **Violin**: We use an in-house collection of around 16.7 hours of high-fidelity violin solo recordings of classical music from various composers and violinist.

4.2. Model Implementation Details

For fair performance comparison with our prior model [22], we follow as closely its model settings, except for the major changes mentioned in Section 3. We use 20-dimensional noise vectors as the input (i.e., $N = 20$), and 80-dimensional mel-spectraograms as the output acoustic features (i.e., $K = 80$). The parameters of the network are optimized with Adam [33] with 0.0001 learning rate. And, a mini-batch of size 5 is used and 100,000 updates are executed for each model. The training takes around 0.5 1 day on an NVIDIA RTX 2080Ti.

The mel-spectrograms are converted into waveforms using the MelGAN [20] as the vocoder. Except for speech, we train a specific MelGAN for each of the audio types listed in Section 4.1. For speech, we directly use the vocoder trained on LJ Speech provided in the official MelGAN GitHub repository. [2]

4.3. Evaluation Metrics

We employ the following objective metrics for our task.

- **Vocalness** measures whether an audio clip contains human voices. Following [22], we employ the JDC model [34][3] for measuring vocalness. The JDC model regards a frame as being vocal if it has a vocal activation ≥ 0.5 AND if the detected pitch value falls within a reasonable human pitch range (i.e., 73–988 Hz). We define the vocalness of an audio clip as the percentage of non-silent frames that are vocal.[4]
- **Diversity**. Following [27], we employ the following two diversity metrics proposed by Richardson *et al.* [35] to examine the generated mel-spectrograms: statistically-different bins (NDB) and Jensen-Shannon divergence (JSD). They measure diversity by 1) clustering the training data into several clusters, and 2) measuring how well the generated samples fit into those clusters. In other words, it uses the training data as a baseline of diversity and compares the generated samples with this baseline. We use the the official implementation of NDB and JSD.[5]

[2] `https://github.com/descriptinc/melgan-neurips`

[3] `https://github.com/keums/melodyExtraction_JDC`

[4] The non-silent frames are derived by using the `librosa` function 'effects._signal_to_frame_nonsilent.'

[5] `https://github.com/eitanrich/gans-n-gmms/blob/master/utils/ndb.py`

Table 2: *Result of subjective evaluation on a 1-to-5 five point Likert scale; the higher the better.*

Singing	**Non-Hier.**	**Hier.**	**Hier. w/ cycle**
Naturalness	2.55 ± 1.07	**3.40 ± 0.92**	2.60 ± 1.02
Audio Quality	1.90 ± 0.70	**2.75 ± 0.94**	2.45 ± 1.02
Diversity	2.60 ± 0.97	**3.15 ± 0.73**	2.95 ± 0.97
AI Vocalness	2.75 ± 1.26	**3.40 ± 0.86**	2.85 ± 1.15
Speech			
Naturalness	NA	2.42 ± 1.02.	**3.50 ± 1.06**

Table 3: *Result of objective evaluation. The Vocalness [34] are the higher the better, while NDB and JSD [35] are the opposite.*

Singing	**Non-Hier.**	**Hier.**	**Hier. w/ cycle**
Vocalness ↑	0.48 ± 0.11	0.58 ± 0.07	**0.64 ± 0.10**
NDB ↓	**48**	61	50
JSD ↓	**0.04**	0.06	0.05
Speech			
Vocalness ↑	NA	0.35 ± 0.07	**0.49 ± 0.04**
NDB ↓	NA	37	**8**
JSD ↓	NA	0.03	**0.01**

In the subjective evaluation, we ask human listeners to rate the generated samples (on a five-point scale) by the following four metrics for singing, and by Naturalness only for speech.

- **Naturalness** measures whether an audio recording sounds like real singing voices or speeches.
- **Audio Quality** measures the perceptual audio quality.
- **Diversity** measures the diversity of the audio contents across three different samples generated by the same model, and presented to the participant consecutively.
- **AI Vocalness** measures whether the generated singing voices fit the listener's own (and subjective) expectation of vocals from an AI. This metric is included with the assumption that an singing voice generating AI may have its own timbre that cannot be found in human voices.

4.4. Subjective Evaluation Result

We discuss the subjective evaluation first. We solicit non-paid responses from the Internet with an online questionnaire. For the singing part, a subject is asked to listen to three 10-second samples generated by a model and then rate the performance of the model based on an overall impression on the three samples. For the speech part, the subject rates (the Naturalness of) each generated 10-second sample individually.

We compare the following three models for the singing part:

- **Non-Hier.**: The old model originally proposed in [22].
- **Hier.**: The proposed new model using the hierarchical structure but not the cycle regularization.
- **Hier. w/ cycle**: The proposed model with both hierarchical structure and cycle regularization.

For the speech part, we only compare the second two models. We ask a subject to rate in total 6 generated samples, 3 from each model. The ordering of the samples are randomized.

We inform the subjects that the samples are freely generated by machine without following any text, lyrics, or pitch labels.

The responses from 20 subjects are summarized in Table 2. We can see that the generators with a hierarchical architecture greatly outperform the non-hierarchical one in almost all the four metrics for the singing voices, demonstrating the effectiveness of the proposed hierarchical structure. Moreover, while the average rating for the old model is all under '3' (i.e., below average), this is not the case for the new model, except for Audio Quality.

Interestingly, the cycle regularization largely improves the generation quality for the speeches, but not for the singing. We conjecture that there might be two reasons for this discrepancy. First, the training data for the singing voices are the output of a source separation model, which could be noisy, while the training data for the speeches are clean speech data from a single person. The cycle regularization retain the modes but can also retain the modes of the noisy signals at the same time. Second, compared to the singing voices, speeches have a clearer target for the encoder to predict, that is, the phonemes, while the factors that are involved in singing are more complicated [36, 37].

4.5. Objective Evaluation Result

For the objective evaluation, 100 10-second samples are generated by each model with the same random seed. We report the average scores across the 100 samples in Table 3.

For the vocalness, we can see that the hierarchical architectures perform better than the non-hierarchical one for both singing voices and speeches. Using cycle regularization makes relatively moderate difference in the singing voices, but large improvement in speech. This finding seems to be consistent with the result of subjective evaluations.

From the result of NDB and JSD, we can see that hierarchical architectures with cycle regularization improves the diversity for both singing voices and speeches. This is as expected because the cycle regularization is meant to reduce mode collapse.

There are two interesting findings related to diversity. First, for singing, the non-hierarchical architecture actually performs the best among the three models. Our conjecture is that the hierarchical structure enforces a consistency among local features and reduces the degree of freedom. This makes the diversity lower in NDB and JSD, but makes the audio more pleasant to listen to, as shown in the subjective evaluation.

Second, the singing voices with cycle regularization has better *objective* diversity, while those without cycle has better *subjective* diversity, comparing Tables 2 and 3. This may be related to the artifact of source separation again—the diversity caused by the artefacts might be counted by the objective metrics but not perceived as contributing to diversity subjectively.

5. Conclusions

In this paper, we have proposed a new model for unconditional generation of general audio of arbitrary length. It features a hierarchical generator to convert a sequence of random noises into slices of a mel-spectrogram, and a cycle regularizer between the noise input and the output. We have subjectively and objectively validated the effectiveness of the proposed design in different types of voices. For future work, we intend to compare the model against other existing models (e.g., [21]), and to further improve the audio quality of the generated samples.

6. References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[2] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. International Conference on Machine Learning*, 2016, p. 1747–1756.

[3] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," *arXiv preprint*, 2019.

[4] A. v. d. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with PixelCNN decoders," in *Proc. Advances in Neural Information Processing Systems*, 2016, p. 4797–4805.

[5] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Symbolic-domain music generation and accompaniment with multi-track sequential generative adversarial networks," in *Proc. AAAI Conf. Artificial Intelligence*, 2018.

[6] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music Transformer: Generating music with long-term structure," in *Proc. International Conference on Learning Representations*, 2019.

[7] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Generating music with rhythm and harmony," *arXiv preprint arXiv:2002.00212*, 2020.

[8] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[9] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 4779–4783.

[10] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANsynth: Adversarial neural audio synthesis," in *Proc. International Conference on Learning Representations*, 2019.

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

[12] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[13] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proc. International Conference on Learning Representations*, 2020.

[14] J. Parekh, P. Rao, and Y.-H. Yang, "Speech-to-singing conversion in an encoder-decoder framework," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 5535–5539.

[15] E. Dunbar, R. Algayres, J. Karadayi, M. Bernard, J. Benjumea, X. Cao, L. Miskic, C. Dugrain, L. Ondel, A. W. Black, L. Besacier, S. Sakti, and E. Dupoux, "The zero resource speech challenge 2019: TTS without T," in *Proc. INTERSPEECH*, 2019.

[16] K.-Y. Chen, C.-P. Tsai, D.-R. Liu, H.-Y. Lee, and L.-S. Lee, "Completely unsupervised phoneme recognition by a generative adversarial network harmonized with iteratively refined hidden Markov models," in *Proc. INTERSPEECH*, 2019.

[17] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 12, pp. 2041–2053, 2019.

[18] R. Eloff, A. Nortje, B. van Niekerk, A. Govender, L. Nortje, A. Pretorius, E. V. Biljon, E. van der Westhuizen, L. van Staden, and H. Kamper, "Unsupervised acoustic unit discovery for speech synthesis using discrete latent-variable neural networks," in *Proc. INTERSPEECH*, 2019, pp. 1103–1107.

[19] S. Dieleman, A. van den Oord, and K. Simonyan, "The challenge of realistic music generation: modelling raw audio at scale," *arXiv preprint*, jun 2018.

[20] 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," *arXiv preprint arXiv:1910.06711*, 2019.

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

[22] J.-Y. Liu, Y.-H. Chen, Y.-C. Yeh, and Y.-H. Yang, "Score and lyrics-free singing voice generation," in *Proc. International Conference on Computational Creativity*, 2020.

[23] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," *arXiv preprint*, 2017.

[24] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proc. Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014.

[25] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NIPS Workshop on Deep Learning*, 2014.

[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training GANs," in *Proc. Advances in Neural Information Processing Systems*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2234–2242. [Online]. Available: http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf

[27] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019.

[28] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint*, 2017.

[29] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "It takes (only) two: Adversarial generator-encoder networks," Tech. Rep., apr 2018.

[30] K. Ito, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[31] J.-Y. Liu and Y.-H. Yang, "Dilated convolution with dilated GRU for music source separation," in *Proc. International Joint Conference on Artificial Intelligence*, 2019, pp. 4718–4724.

[32] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *Proc. International Confernece on Learning Representations*, 2019.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] S. Kum and J. Nam, "Joint detection and classification of singing voice melody using convolutional recurrent neural networks," *Applied Sciences*, vol. 9, no. 7, 2019, [Online] https://github.com/keums/melodyExtraction_JDC.

[35] E. Richardson and Y. Weiss, "On GANs and GMMs," in *Proc. Advances in Neural Information Processing Systems*, 2018, pp. 5852–5863.

[36] J. Lee, H. Choi, C. Jeon, J. Koo, and K. Lee, "Adversarially trained end-to-end Korean singing voice synthesis system," in *Proc. INTERSPEECH*, G. Kubin and Z. Kacic, Eds., 2019, pp. 2588–2592.

[37] K. Qian, Y. Zhang, S. Chang, D. Cox, and M. Hasegawa-Johnson, "Unsupervised speech decomposition via triple information bottleneck," *arXiv preprint arXiv:2004.11284*, 2020.