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

Audio Dequantization for High Fidelity Audio Generation in Flow-based Neural Vocoder

Hyun-Wook Yoon[1], *Sang-Hoon Lee*[2], *Hyeong-Rae Noh*[2], *Seong-Whan Lee*[2,3]

[1]Department of Computer and Radio Communications Engineering, Korea University, Seoul, Korea
[2]Department of Brain and Cognitive Engineering, Korea University, Seoul, Korea
[3]Department of Artificial Intelligence, Korea University, Seoul, Korea
{hw_yoon, sh_lee, hr_noh, sw.lee}@korea.ac.kr

Abstract

In recent works, a flow-based neural vocoder has shown significant improvement in real-time speech generation task. The sequence of invertible flow operations allows the model to convert samples from simple distribution to audio samples. However, training a continuous density model on discrete audio data can degrade model performance due to the topological difference between latent and actual distribution. To resolve this problem, we propose audio dequantization methods in flow-based neural vocoder for high fidelity audio generation. Data dequantization is a well-known method in image generation but has not yet been studied in the audio domain. For this reason, we implement various audio dequantization methods in flow-based neural vocoder and investigate the effect on the generated audio. We conduct various objective performance assessments and subjective evaluation to show that audio dequantization can improve audio generation quality. From our experiments, using audio dequantization produces waveform audio with better harmonic structure and fewer digital artifacts.

Index Terms: audio synthesis, neural vocoder, flow-based generative models, data dequantization, deep learning

1. Introduction

Most speech synthesis models take two-stage procedures to generate waveform audio from the text. First stage generates spectrogram conditioned on linguistic features such as text or phoneme. [1–5] In second stage, generally refer to as vocoder stage, audio samples are generated through model capable of estimating audio samples from the acoustic features. Traditional approaches estimated audio samples either directly from the spectral density model [6] or hand-crafted acoustic model [7,8], but these approaches tended to produce low-quality audio.

After the emergence of the WaveNet [9], models that generate audio samples on previously generated samples had shown exceptional works in the field. [10–12]. Nevertheless, dilated causal convolution networks used in the model require sequential generation process during the inference, which infers that real-time speech synthesis is hard to achieve because parallel inference can't be utilized. For this reason, generating high-quality waveform audio in real-time has become a challenging task.

To overcome the structural limitation of the auto-regressive model, most of the recent works are focused on non-autoregressive models such as knowledge distillation [13, 14], generative adversarial network [15–19], and flow-based generative model [20, 21]. We focus on the flow-based generative model since it can model highly flexible approximate posterior distribution in variational inference [22]. The transformation from a single data-point to a Gaussian noise is one-to-one, which makes the parallel generation possible. However, we have to acknowledge that audio samples are discrete data. In other words, naive modeling of a continuous probability density on discrete data can produce arbitrary high likelihood on discrete location [23, 24]. This can lead to degraded generation performance in flow-based neural vocoder. Therefore, *dequantization* is required before the transformation.

In this paper, we present various audio dequantization schemes that can be implemented in the flow-based neural vocoder. In image generation, adding continuous noise to datapoints to *dequantize* the data is commonly used. However, to the best of our knowledge, the effectiveness of data dequantization in audio domain is still an unknown area, so further investigation is needed. Unlike pixels of the image, audio samples are bounded to signed integer. To overcome this domain issue, we either normalize range of noise values or range of audio samples with different normalization method. In addition, we adapt flow block from flow-based neural vocoder to generate more flexible noises known as *variational dequantization* [25].

2. Flow-based Neural Vocoder

FloWaveNet [21] and WaveGlow [20] are two pioneers in flow-based neural vocoders. Both models are based on normalizing flow [22]. Two main contributions that they share are training simplicity and faster generation. Since they use a single invertible flow network repeatedly, model structure is intuitive. Moreover, optimization can be easily done with a single log-likelihood loss function. During inference, random noises of equal length to the product of frames of mel-spectrogram and hop-size are sampled from the spherical Gaussian distribution and simultaneously converted to audio samples. As a result, flow-based neural vocoders can produce waveform signal as fast as other non-autoregressive models.

In general, flow-based neural vocoder requires three steps: squeeze, flow, and shuffle. In the squeezing step, the temporal dimension of the feature is reduced whereas channels of the feature are increased. According to FloWaveNet [21], this operation increase the size of the receptive field like dilated convolutions layer from the WaveNet [9]. During the flow step, multiple blocks of flow operate affine transformation on the half of input vectors. In detail, half input vectors are used to predict shift and scale parameters for the other half in each flow operation.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00079, Department of Artificial Intelligence, Korea University), the Magellan Division of Netmarble Corporation, and the Seoul R&BD Program(CY190019).

http://dx.doi.org/10.21437/Interspeech.2020-1226

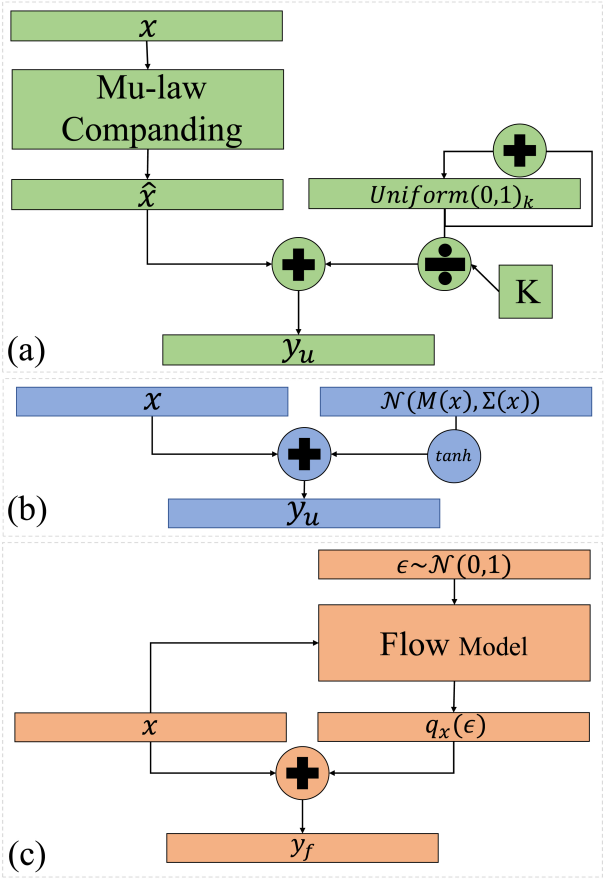

Figure 1: *Examples of audio dequantization for flow-based neural vocoder. (a) represents uniform dequantization. (b) represents Gaussian dequantization. (c) represents variational dequantization.*

Lastly, the shuffle step mixes elements of data, giving flexibility in transformation.

Although both models have similar concepts, they use different techniques in detail. FloWaveNet [21] defines one squeeze operation and multiple flow operations as a single context block and duplicates this context block to form the flow process. The model also implements the activation normalization layer suggested in Glow [26] before the coupling layer to stabilize training. After each flow, the model simply swap odd and even elements of vectors to shuffle features. WaveGlow [20] operates the squeezing step only once during the process. Also, the model adapts invertible 1x1 convolution from Glow to operate the shuffle step before each flow operation.

3. Audio Dequantization

A raw audio is stored in computer digitally. In other words, values of the audio are formed as discrete representations. Therefore, naively transforming audio samples into Gaussian noise can lead to arbitrary high likelihood on values of data in flow-based neural vocoder. To resolve this issue, we adapt the idea of adding noise to each of the data-point to *dequantize* discrete distribution data in image generation task [23]. In image, a pixel x is represented as a single discrete value in $\{0, 1, \ldots, 255\}$, so dequantized data y can be formulated as $y = x + u$, where u represents D components of noise bounded to $[0, 1)^D$.

Unlike image, raw audio encoded in 16-bit WAV contains 15-bit of negative and positive integer values, which can be represented as $\{-32,767, \ldots, 32,767\}$. To apply data dequantization to raw audio, either range of audio samples must be compressed to 8-bit unsigned integer, or the range of dequantized data has to be within the range of $(-1, 1)^D$. For the proper audio dequantization, we present three different methods in the following sections.

3.1. Uniform Dequantization

In [23], authors note that optimizing the continuous model $p_{model}(y)$ on the dequantized data $y \sim p_{data}$ can closely optimize the discrete model $P_{model}(x)$ on the origianl data $x \sim P_{data}$ through Jensen's inequality, which can be formulated as below:

$$\int p_{data}(y) \log p_{model}(y) dy \tag{1}$$

$$= \sum_x P_{data}(x) \int_{[0,1)^D} \log p_{model}(x+u) du \tag{2}$$

$$\leq \sum_x P_{data}(x) \log \int_{[0,1)^D} p_{model}(x+u) du \tag{3}$$

$$= \mathbb{E}_{x \sim P_{data}} [\log P_{model}(x)] \tag{4}$$

Since the uniform noise is bounded to $[0, 1)^D$, values of audio samples have to be bounded to unsigned integer. For this purpose, we preprocess raw audio with nonlinear companding method called '*mu-law companding*' [27]. This method can significantly reduce the range of audio samples while minimizing the quantization error. Redistribution equation of the method can be expressed as:

$$\hat{x} = sign(x)\left(\frac{ln(1+\mu|x|)}{ln(1+\mu)}\right) \tag{5}$$

where $sign$ function represents sign of value x, and μ represents integers that x values are mapped. We set μ to 255 to apply 8-bit *mu-law companding*. Then, we add random noise from uniformly distributed function formulated as $Unif(0, 1)$.

We assume that companding audio with lossy compression can possibly produce noisy output. Therefore, we implement *iw(importance-weighted) dequantization* proposed in [24] to improve generation quality. In the paper, authors demonstrate that sampling noise multiple times can directly approximate the objective log-likelihood, which can lead to better log-likelihood performance. As a result, we define uniformly dequantized data y_u as:

$$y_u = \hat{x} + \frac{1}{K} \sum_{k=1}^{K} Unif(0, 1)_k \tag{6}$$

where $Unif(0, 1)$ defines noise sampled uniformly from $[0, 1)^D$. We set K to 10.

To compare the performance of model depending on *iw dequantization*, we refer to uniform dequantization with *iw dequantization* as *Uniform_IW* and only uniform deqauntization model as *Uniform*.

3.2. Gaussian Dequantization

In flow-based neural vocoder, discrete data distribution is transformed into a spherical Gaussian distribution. In other words,

dequantizing data distribution to normal distribution can be more optimal choice. With this though in mind, we formulate Gaussian dequantization motivated from logistic-normal distributions [28]. Random noise samples are generated from normal distribution formulated as $\mathcal{N}(\mu, \sigma^2)$, where mean and variance are calculated from the given data batch. To properly implement in audio domain, we apply a hyperbolic tangent function to normalize noise boundary at $(-1, 1)^D$. As a result, normally dequantized data y_n can be formulated as:

$$y_n = x + \tanh(\mathcal{N}(M(x_b)), \Sigma(x_b)) \tag{7}$$

where $M(x_b)$ and $\Sigma(x_b)$ represent mean and variance of batch group x_b.

To compare model performance between conventional and improved method, we refer to the conventional method suggested in [28] as *Gaussian_Sig* and proposed method as *Gaussian_Tanh*.

3.3. Variational Dequantization

Instead of adding noise from known distribution, noise distribution can be formulated through a neural network such as a flow-based network. Flow++ [25] suggests that if the noise samples u are generated from conditional probability model $q(u|x)$, probability distribution of original data can be estimated as follows:

$$P_{model}(x) := \int_{[0,1)^D} q(u|x) \frac{p_{model}(x+u)}{q(u|x)} du \tag{8}$$

Then, we can obtain the variational lower-bound on the log-likelihood function by applying Jensen's inequality as below:

$$\mathbb{E}_{x \sim P_{data}} \left[\log P_{model}(x)\right] \tag{9}$$

$$= \mathbb{E}_{x \sim P_{data}} \left[\log \int_{[0,1)^D} q(u|x) \frac{p_{model}(x+u)}{q(u|x)} du\right] \tag{10}$$

$$\geq \mathbb{E}_{x \sim P_{data}} \left[\int_{[0,1)^D} q(u|x) \log \frac{p_{model}(x+u)}{q(u|x)} du\right] \tag{11}$$

$$= \mathbb{E}_{x \sim P_{data}} \mathbb{E}_{u \sim q(u|x)} \left[\log \frac{p_{model}(x+u)}{q(u|x)}\right] \tag{12}$$

As a result, dequantized data y_p from variational dequantization can be defined as:

$$y_p = x + q_x(\epsilon) \tag{13}$$

where $\epsilon \sim p(\epsilon) = \mathcal{N}(\epsilon; 0, I)$.

To implement variational dequantization in flow-based neural vocoder, we modify flow model from FloWaveNet [21]. We set initial input as 1-dimensional noise vector generated from spherical Gaussian distribution $\mathcal{N}(\epsilon; 0, I)$, where the length is equal to target audio. In each context block, single squeeze step and multiple flow steps are operated. In each flow step, an affine transformation conditioned on target audio is applied to the half of squeezed vector. At the end, the vector is flattened, and hyperbolic tangent function is applied to fit range of audio domain. Negative log likelihood from the dequantizer is trained jointly with the flow-based neural vocoder.

We set a total of 16 flow stacks as *Flow_Shallow* and 48 flow stacks as *Flow_Dense* to examine whether the dept of dequantization model is critical to the model performance.

Table 1: *Mean opinion score (MOS) results with 95% confidence intervals on 150 randomly selected sentences in test set.*

Methods	MOS	95% CI
Ground Truth	4.489	±0.016
Baseline [21]	2.898	±0.043
Uniform_IW	3.054	±0.041
Gaussian_Tanh	3.029	±0.041
Flow_Dense	3.267	±0.036

4. Experimental Results and Analysis

4.1. Experimental Settings

We set FloWaveNet [21] as our baseline model and trained baseline with 6 different dequantization methods. Each model was trained with VCTK-Corpus [29] containing 109 native speakers English dataset. Since FloWaveNet and WaveGlow [20] evaluated with only a single speaker dataset, we expanded the experiment on the model to a multimodal case where audio generation is much harder due to the larger variation among different speakers. In the dataset, we withdrew some corrupted audio files and used 44,070 audio clips. For each speaker, 70% of data were used as training data, 20% as validation data, and rest of them as test data. All clips were down-sampled from 48,000Hz to 22,050Hz. From each audio clip, 16,000 chunks were randomly extracted.

All models were trained on 4 Nvidia Titan Xp GPUs with a batch size of 8. We used Adam optimizer with a step size of 1×10^{-3} and set the learning rate decay in every 200K iterations with a factor of 0.5. We trained each model for 600K iterations.

4.2. Subjective Evaluation

For subjective evaluation, we conducted a subjective 5 scale MOS test on Amazon Mechanical Turk[1]. Each participant was suggested to wear either earbuds or headphones for the eligible testing. Then they had to listen to 5 audio clips at least twice and rated naturalness of audio on a scale of 1 to 5 with 0.5 point increments. We explicitly instructed the participants to focus on the quality of audio. We collected approximately 3,000 samples for the evaluation. In the Table 1, models implemented audio dequantization show higher MOS than the baseline model which show that audio dequantization can improve audio quality. Except for the real audio, variational dequantization with a deeper layer receives the highest MOS. This shows that injecting noise with more complex distribution can produce more natural audio.

4.3. Objective Evaluation

Audio generated from the baseline model tended to have digital artifacts such as reverberation, trembling sound, and periodic noise. We assumed that the occurrence of these artifacts was due to the unnatural collapsing from the continuous density model on discrete data-points. To prove that audio dequantization can remove such artifacts, we conducted several quantitative evaluations in signal processing to compare the quality of audio. First we randomly selected 400 sentences in test set. Then, we conducted mel-cepstral distortion (MCD) [30], global signal-to-noise ratio (GSNR) [31], segmental signal-to-noise ratio (SSNR) [11], and root mean square error of fundamental frequency (RMSE_{f0}) [11]. All equations for the evaluation can be

[1] https://www.mturk.com/

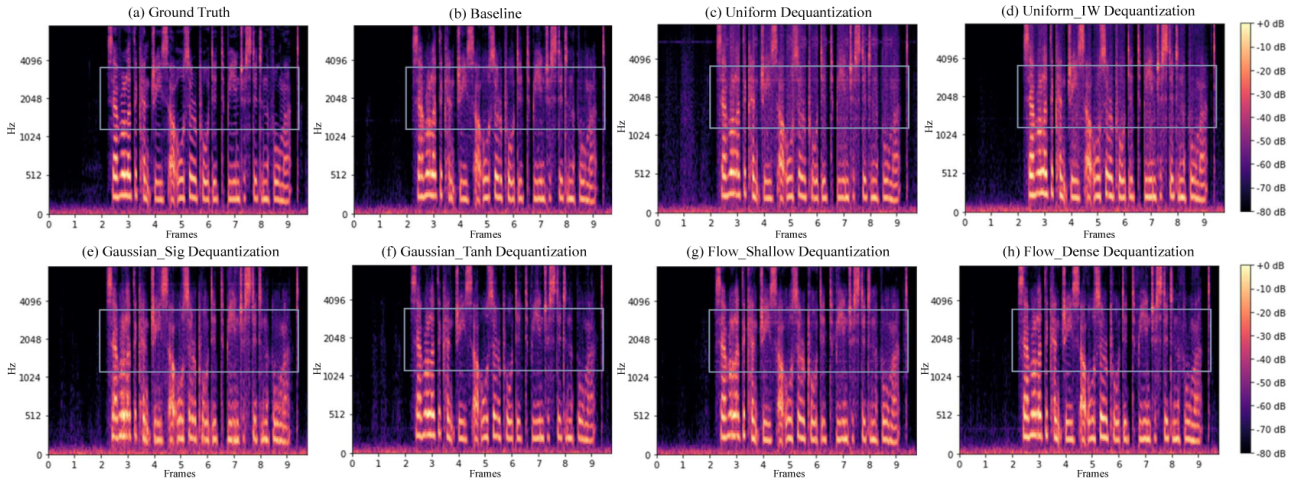

Figure 2: *Mel-spectrogram converted from audio samples generated by baseline [21] and proposed dequantization models. The blue bounding-box indicates the area where periodic noise appears and harmonic frequencies are presented.*

Table 2: *MCD (dB) results with 95% confidence intervals.*

Methods	**MCD$_{13}$**	**95% CI**
Baseline [21]	3.455	±0.032
Uniform	4.139	±0.041
Uniform_IW	3.567	±0.032
Gaussian_Sig	3.739	±0.039
Gaussian_Tanh	3.355	±0.031
Flow_Shallow	3.484	±0.032
Flow_Dense	3.401	±0.032

calculated as follows:

$$MCD_{13}[dB] = \frac{1}{T}\sum_{t=0}^{T-1}\sqrt{\sum_{k=1}^{K}(c_{t,k} - c'_{t,k})^2} \quad (14)$$

$$GSNR[dB] = 10log\frac{\sigma_s^2}{\sigma_r^2} \quad (15)$$

$$SSNR[dB] = 10log_{10}(\frac{\sum_{n=0}^{M} x_s(n)^2}{\sum_{n=0}^{M}(x_s(n) - y_r(n))^2}) \quad (16)$$

$$RMSE_{f0}[cent] = 1200\sqrt{(log_2(F_r) - log_2(F_s))^2} \quad (17)$$

where $c_{t,k}$, $c'_{t,k}$ represent original and synthesized k-th mel frequency cepstral coefficient (MFCC) of t-th frame, σ_s^2, σ_r^2 represent power of speech signal and noise, $x_s(n)$, $y_r(n)$ represent raw and synthesized waveform sample at time n, and F_r, F_s represent fundamental frequency of raw and synthesized waveform.

In MCD, we compared all models including *Uniform* and *Gaussian_Sig* to see the improvement within the modification. In Table 2, dequantizations with modified methods show better performance than the conventional methods. *Gaussian_Tanh* and *Flow_Dense* score relatively lower MCD than baseline, which shows that both models can produce better audio quality than the baseline model. There was no significant performance difference between the two variational dequantization methods. *Uniform_IW* shows slightly higher MCD than the baseline because of the remaining audible noise generated from mu-law companding.

Table 3: *GSNR (dB), SSNR (dB), and RMSE$_{f0}$ (Hz) result. Higher is better for SNR and lower is better for RMSE$_{f0}$*

Methods	**GSNR**	**SSNR**	**RMSE$_{f0}$**
Baseline [21]	-2.127	-2.284	44.881
Uniform_IW	-1.902	-1.990	38.359
Gaussian_Tanh	-2.112	-2.186	37.208
Flow_Dense	-2.048	-2.141	44.066

Table 3 presents the result of SNR and RMSE$_{f0}$. All proposed methods show higher SNR than baseline, indicating that audio dequantization can help reducing noise. In addition, *Uniform_IW* and *Gaussian_Tanh* dequantization show better performance in modeling fundamental frequency, while *Flow_Dense* dequantization shows a comparable result with the baseline model. We also visualized test outputs for qualitative evaluation in Figure 2. Figure 2(f) and Figure 2(h) show clearer harmonic structures than Figure 2(b). Although Figure 2(d) shows less clear harmonic structures than other approaches, we can see that periodic noises are reduced significantly. We provide audio results on our online demo webpage.[2]

5. Conclusions

In this paper, we proposed various audio dequantization schemes that can be implemented in flow-based neural vocoder. For the uniform dequantization, we compressed the range of audio domain to match with conventional uniform dequantization method by using *mu-law companding* compression. In addition, we implemented *iw dequantization* to resolve the noise issue that occurs from the lossy compression. For the Gaussian dequantization, we applied hyperbolic tangent normalization on data-oriented Gaussian noise to properly fit the data within the audio range. Lastly, we modified flow block in flow-based neural vocoder to construct variational dequantization model to apply more flexible noise. From the experiments, we demonstrate that implementing audio dequantization can supplement the flow-based neural vocoder to produce better audio quality with fewer artifacts.

[2] https://claudin92.github.io/deqflow_webdemo/

6. References

[1] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems*, 2018, pp. 4480–4490.

[2] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Interspeech*, 2017, pp. 4006–4010.

[3] J. Park, K. Han, Y. Jeong, and S. W. Lee, "Phonemic-level duration control using attention alignment for natural speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 5896–5900.

[4] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "Voiceloop: Voice fitting and synthesis via a phonological loop," in *International Conference on Learning Representations*, 2018.

[5] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 4779–4783.

[6] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1984, pp. 236–243.

[7] H. Kawahara, "Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds," *Acoustical Science and Technology*, vol. 27, no. 6, pp. 349–353, 2006.

[8] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[9] 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," *arXiv preprint arXiv:1609.03499*, 2016.

[10] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent wavenet vocoder." in *Interspeech*, 2017, pp. 1118–1122.

[11] T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, and T. Toda, "An investigation of multi-speaker training for wavenet vocoder," in *IEEE Automatic Speech Recognition and Understanding Workshop*, 2017, pp. 712–718.

[12] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman *et al.*, "Deep voice: Real-time neural text-to-speech," in *International Conference on Machine Learning*, vol. 70. JMLR. org, 2017, pp. 195–204.

[13] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," in *International Conference on Learning Representations*, 2018.

[14] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," in *International Conference on Machine Learning*, 2018, pp. 3918–3926.

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

[16] P. Neekhara, C. Donahue, M. Puckette, S. Dubnov, and J. McAuley, "Expediting tts synthesis with adversarial vocoding," in *Interspeech*, 2019, pp. 186–190.

[17] R. Yamamoto, E. Song, and J.-M. Kim, "Probability density distillation with generative adversarial networks for high-quality parallel waveform generation," in *Interspeech*, 2019, pp. 699–703.

[18] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 881–14 892.

[19] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 6199–6203.

[20] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 3617–3621.

[21] S. Kim, S.-g. Lee, J. Song, J. Kim, and S. Yoon, "Flowavenet: A generative flow for raw audio," in *International Conference on Machine Learning*, 2019, pp. 3370–3378.

[22] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International Conference on Machine Learning*, 2015.

[23] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," in *International Conference on Learning Representations*, 2015.

[24] E. Hoogeboom, T. S. Cohen, and J. M. Tomczak, "Learning discrete distributions by dequantization," *arXiv preprint arXiv:2001.11235*, 2020.

[25] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, "Flow++: Improving flow-based generative models with variational dequantization and architecture design," in *International Conference on Machine Learning*, 2019, pp. 2722–2730.

[26] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 215–10 224.

[27] T. Yoshimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Mel-cepstrum-based quantization noise shaping applied to neural-network-based speech waveform synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 7, pp. 1177–1184, 2018.

[28] S. M. S. J Atchison, "Logistic-normal distributions: Some properties and uses," 1980.

[29] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2016.

[30] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1, 1993, pp. 125–128.

[31] M. Vondrasek and P. Pollak, "Methods for speech snr estimation: Evaluation tool and analysis of vad dependency," *Radioengineering*, vol. 14, no. 1, pp. 6–11, 2005.