Reproduction of MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis

Yifei Zhao
yifei.zhao@mail.mcgill.ca

Yichao Yang
yichao.yang@mail.mcgill.ca

Yang Gao
yang.gao6@mail.mcgill.ca

Abstract

Raw audio generation is a challenging task due to its complicated time-dependent structure. Previous work (by Thibault de Boissiere et al.[1]) has shown that it is possible to use Generative adversarial networks (GANs) to generate high quality coherent waveforms by introducing a set of architectural changes and simple training techniques. In this project, we use the LJ Speech Dataset for training and testing the model. We evaluate various components of the model through ablation studies. It has shown in the results section that the number of residual blocks in the generator plays a more important role than other components (e.g. the number of discriminators in the model). After ablation studies, we accordingly combine those modified parts that improve the performance and generate audio files of better quality (The PESQ-MOS score is 0.5 higher than the score of baseline). To interpret the results of different experiments, we introduce 'Perceptual Evaluation of Speech Quality' (or PESQ) to evaluate the quality of the generated audio files.

1 Introduction

Raw audio is a complex, highly structured sequential data modality. Its structure at different timescales can show both short and long term dependencies. Therefore, most approaches simplify the problem by modeling a lower-resolution representation that can be efficiently computed from the raw temporal signal.

As introduced in our reproducing paper[1], in this project we use Generative adversarial networks (GANs) to model raw audio files. GANs currently constitute one of the dominant paradigms for generative modeling of images, and they are able to produce high-delity samples that are almost indistinguishable from real data. However, their application to audio generation tasks has seen relatively limited success so far. In the many previous works, authors explored methods of raw waveform generation with GANs and demonstrated that adversarially trained feed-forward generators are indeed able to synthesize speech audio.[2]

1.1 A Brief Description of the Paper

The main task of our reproducing paper [1] is to implement MelGAN, a non-autoregressive feed-forward convolutional architecture to perform audio waveform generation in a GAN setup.

For speech synthesis, the authors introduced several techniques deployed by MelGAN, such as exponentially increasing dilated convolution layers, weight normalization in all layers of the generator, multiple discriminator blocks to learn frequency features. This gives us some intuitions to investigate the original model. As discussed in the approach and results section, we change the number of discriminator blocks, let stride size increase exponentially and try our model with 2 normalization techniques. Some modifications indeed improve the performance of our model.

1.2 Experiment Set-up

The first goal of this work is to reproduce the results presented in the paper [1] by running the released code provided by the authors.

We then explore how different components affect model performance. for instance, we test the model performance by changing the number of layers in discriminator, activation functions in the Generator, the learning rate of the optimization function in the model and so on. In this process, we perform ablation studies to understand the model's robustness.

Furthermore, we attempt to modify the model by using different architectures and hyperparameters. Instead of using loops to construct neural network layers with fixed parameters, we construct each layer with different parameters and try to improve the overall performance.

1.3 Dataset Information

In this experiment, we choose the LJ Speech Dataset, one of the most popular datasets for audio modeling training experiments. This is a public domain speech dataset consisting of 13,100 short audio clips of a single speaker reading passages from 7 non-fiction books. A transcription is provided for each clip. Clips vary in length from 1 to 10 seconds and have a total length of approximately 24 hours.

2 Related Works

The optimizer is an important component in the model, we refer several papers related to different kinds of optimizers during the experiments: Vaishnavh et al. [3] proposed a regularized gradient descent method when training GANs model, which addressed the mode collapse issue and guaranteed the local stability with the speeding up convergence. Yann N. Dauphin et al. [4] discussed the properties of the RMSPROP optimizer to solve non-convex task. The paper by Xiangyu Zeng et al.[5] introduced how Adamax optimizer works and compared to Adam optimizer. Based on these previous work, we make experiments to test how different optimizers behave in the model.

Dominik Scherer et al. [6] discussed that generally a maximum pooling operation significantly outperforms subsampling operations. This gives us an intuition to replace average pooling operation with maximum pooling operation. It indeed improved the performance significantly.

We also change the padding operation in the experiment. T Sree Harsha and Dr. Balaji Srinivasan [7] showed 3 padding techniques that used in convolutional neural network. Moreover, Guilin Liu et al. [8] discussed the padding techniques for partial convolution. Based on these work, we make a comparison between replication and reflection padding operations.

Sitao Xiang and Hao Li [9] attach importance to the effects of normalization techniques on the performance of GANs, particularly the weight normalization technique is more favorable in image generating scenario than the model using batch normalization technique. While Takeru Miyato et al. [10] put forward a novel normalization method: spectral normalization, and showed that it is capable of generating images of better or equal quality. Based on these materials, we further explore the effects of various normalization techniques on the performance of MelGAN model.

To interpret the performances of our model during experiments, we use a standard metric *Perceptual Evaluation of Speech Quality* (PESQ) proposed by Antony W. Rix et al. [11] to calculate Mean Opinion Score (MOS). We then calculate the average score of 8 generated samples.

Additionally, the International Telecommunication Union(ITU) [12] presents other methods to evaluate the audio quality such as the Outlier ratio, Pearson correlation coefficient and RMSE. Instead of using the loss function to evaluate the importance of different feature components, we introduce the Pearson correlation coefficient.

3 Proposed Approach

3.1 Dataset and Set-up

As is introduced in the previous section, in this project we choose the LJ Speech Dataset, which consists of 13,100 short audio clips in total. As the paper suggested, we choose 13,091 wave files as the training set and use the remaining 10 wave files as test datasets.

3.2 Preprocessing

Modeling raw audio is a particularly challenging problem, thus instead of modeling the raw audio directly, we choose mel-spectrogram inversion as an intermediate representation. A visualization of mel-spectrogram inversion is shown in figure 1

Extraction: The initial stage consists of the decomposing waveform into time and frequency using the Short-Time Fourier Transformation (STFT). Then the phase information is discarded leaving only the linear-amplitude magnitude spectrogram. The linear-spaced frequency bins of the resultant spectrogram are then compressed to fewer bins which are equally-spaced on a logarithmic scare (Mel scale).
Inversion: To heuristically invert this procedure (vocode), Firstly, logarithmic amplitudes are converted to linear ones. Then, an appropriate magnitude spectrogram is estimated from the mel-spectrogram. Finally, appropriate phase information is estimated from the magnitude spectrogram, and the inverse STFT is used to render audio.

Unless otherwise specified, throughout this paper we operate on waveforms sampled at 22050Hz using an STFT with a window size of 1024 and a hop size of 256. We compress magnitude spectrograms to 80 bins equally spaced along the mel scale from 125 Hz to 76000Hz.

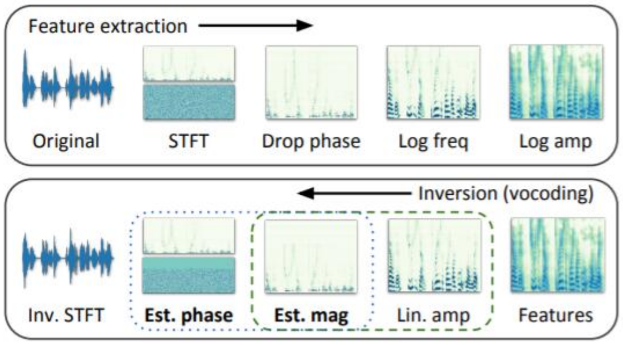


Figure 1: Mel-Spectrogram Inversion

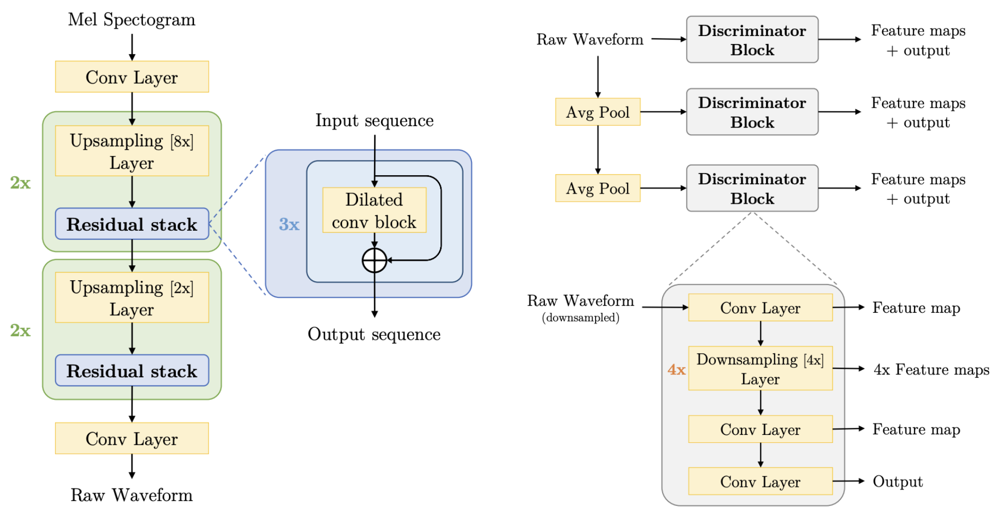
3.3 Original Model

The original model consists of a generator and a discriminator architecture for mel-spectrogram inversion. The overall structure is shown in figure 2, weight normalization is applied in all layers of generator and discriminator.

3.3.1 Generator

The generator is a fully convolutional feed-forward network with mel-spectrogram s as input and generates raw waveform x as output. The mel-spectrogram is at a $256\times$ lower temporal resolution, which is upsampled by a stack of transposed convolutional layers and residual blocks with dilated convolutions.

Induced receptive field: The authors designed the generator to put an inductive bias that there is long range correlation among the audio time steps. Since residual blocks with dilation after each upsampling layer are added to the generator, temporally far output activations of each subsequent layer have significant overlapping inputs. And since the receptive field of a stack of dilated convolution layers increases exponentially with the number of layers, incorporating these in the generator



(a) Generator (b) Discriminator

Figure 2: raw model architecture

efficiently increases the induced receptive fields of each output time-step, which leads to a better long range correlation.

3.3.2 Discriminator

Multi-Scale Architecture: 3 discriminators (D_1, D_2, D_3) with identical network structure but different audio scales are introduced in the discriminator architecture. D_1 operates on the scale of the generated audio files, D_2 and D_3 operate generated audio files downsampled by a factor of 2 and 4 respectively.

Window-based objective: Grouped convolutions are used to allow the larger kernel sizes for a small number of parameters. The window-based discriminator learns to classify between the distribution of small audio chunks. Since the discriminator loss is computed over the overlapping windows where each window is very large (equal to the receptive field of the discriminator), the MelGAN model learns to maintain coherence across patches.

3.3.3 Feature Matching

The object of feature matching is to minimize the L1 distance between the discriminator feature maps of real and synthetic audio files. Intuitively, this can be seen as a learned similarity metric, where a discriminator learns a feature space that discriminates the fake data from real data.

3.4 Modified Model

Due to our computational and resource constraints, the scope of our reproduction is narrowed down from that of the original paper. In particular, instead of running the model for 3,000 epochs as the author proposed, we only run the code for 100 epochs, but we use the same dataset of the same size. So in our experiments, the baseline is the reconstructed audio files generated from 100 epochs without any modification on the original model. Based on this, we perform rigours ablation study as follows:

Residual blocks: The architecture of the generator consists of 4 transposed convolutional layers to upsample the input sequence and each transposed convolutional layer is followed by 3 residual blocks with dilated convolutions. Thus, we investigate the influence of the different number of residual blocks on the performance of the model. We also investigate the effects of different activation functions used in the residual block.

Discriminator: The original model uses 3 discriminators and each discriminator consists of 4 convolutional layers. To understand the importance of its different components, we change the number of convolutional layers and discriminators. Apart from changing the activation function, we modify the strides size, type of pooling layer and method of padding.

Normalization technique: The original model applies the weight normalization technique on both the generator and discriminator. We examine the effects of normalization on the performance of the model by just removing weight normalization and replacing it with spectral normalization.

Optimizer: We try several types of optimizers in the generator and discriminator. Specifically, Adam optimizer, Stochastic Gradient Decent (SGD) optimizer, Rprop optimizer, and RMSprop optimizer are applied to the model.

Loss function: L1 loss function is replaced by the reversed Pearson correlation coefficient since we expect the increasing reversed correlation coefficient to reflect the decreasing discriminator loss. L2 loss function is also applied and compared.

4 Results

As mentioned earlier, we run the model for 100 epochs on the original LJ dataset with 13,091 training samples and 10 test samples. We implement 9 ablation studies and hyper-parameters exploration experiments, each experiment generating eight audio files and the measured PESQ is the average of the generated audio quality.

We conduct Mean Opinion Score (MOS) tests to compare the performance of our modified model to competing architectures. However, instead of conducting surveys by asking many people to evaluate the quality of generated audios which seems to be subjective, we use a standard algorithm *Perceptual Evaluation of Speech Quality* (PESQ) proposed by Antony W. Rix et al. [11] to compute MOS, and then calculate the mean of 10 generated samples.

4.1 Reproduction and Baseline

The original architecture of the MelGAN model contains 3 residual blocks, 3 discriminators and each with 9 convolutional layers. As the following tables show, the quality results evaluated by PESQ is 1.75 for PESQ-MOS and 1.457 for MOS-LQO (based on 100 epochs).

The PESQ-MOS scores range from 0 to 4.5 and MOS-LQO scores range from 1.02 to 4.56 The value of PESQ-MOS is the mapping results from PESQ to MOS, and the value of MOS-LQO is the mapping results from MOS to LQO (Listening Quality Objective). Both can be interpretations of the quality of the audio files.

4.2 Ablation Study and Experiments

During ablation studies, we modified several components of the original model:

- Generator: activation function, residual block layer, several normalization techniques
- Discriminator: the number of discriminators, stride size, pooling layer, padding operation
- Optimizers and their learning rates
- Loss Functions: the Pearson correlation coefficient, L2 loss function

4.2.1 Generator

Experiment 1: In this section, different activation functions, various number of residual blocks, with or without weight normalization technique are applied to investigate their influence on the original model. As the **Table** 1 is shown below, the weight normalization technique slightly influences the performance of the model by decreasing the PESQ-MOS, while changing the activation function from LeakyReLU to ReLU results in worse audio quality. Indeed, using ReLU activation function produces metallic noise. Adding the number of residual blocks actually contributes to better audio quality. When the weight normalization technique is replaced with the spectrum normalization, the PESQ score significantly decreases by roughly 0.9 compared to the baseline model.

Table 1: Generator with normalization and different number residual blocks

Table head	Baseline	ReLU	w/o WN[1]	w/ SN[2]	w/ BN[3]	1 RB[4]	2 RBs	4 RBs
PESQ-MOS[5]	1.750	1.286	1.715	1.254	1.274	0.934	1.645	2.221
MOS-LQO[6]	1.457	1.242	1.436	1.194	1.102	1.259	1.542	1.828

[1] WN: Weight normalization.
[2] SN: Spectral normalization.
[3] BN: Batch normalization.
[4] RB: Residual block.
[5] PESQ-MOS: Mapping PESQ to MOS.
[6] MOS-LQO: Mapping raw result scores to MOS-LQO (Listening Quality Objective).

Experiment 2: Furthermore, drop out function is introduced to the model, however, it further pollutes the generated audio, shown as **Table** 2. (Baseline does not contain drop out function)

Table 2: Drop out layer

	Baseline	Drop out (p=0.2)	Drop out (p=0.5)
PESQ-MOS	1.750	1.668	1.535
MOS-LQO	1.457	1.410	1.358

4.2.2 Discriminator

Experiment 3: The number of discriminators plays an important role in the generative adversarial network, thus, the value is changed from 3 to 1, 2 and 4. As shown in **Table** 3, simply increasing the number of discriminators does not guarantee a better audio quality. Reducing one or two discriminators results in similar PESQ scores to the baseline. Overall, the number of discriminators shows a slight influence on model performance.

Table 3: Ablation study of the number of discriminators

	Baseline	1 disc	2 disc	4 disc[1]
PESQ-MOS	1.750	1.685	1.722	1.614
MOS-LQO	1.457	1.387	1.420	1.381

[1] 'n disc' means the number of discriminator blocks is n

Experiment 4: The stride size on each convolutional layer is increased exponentially by a factor of 4. The results in **Table** 4 show that instead of using the down-sampling method before each discriminator, using one discriminator with increasing stride size can be an alternative as the down-sampling method, which slightly increases the PESQ score.

Table 4: Ablation study on the strides size

	(4,4,4,4)[1]	(4,8,12,16)	(2,2,4,8)	(4,8,12,16,20,24,28,32)[2]
PESQ-MOS	1.685	1.776	1.241	1.780
MOS-LQO	1.387	1.473	1.227	1.478

[1] for here all models have only one discriminator.
[2] the form (n_1,n_2,...,n_k) means there are k 1D convolution layers with stride size on each layer equal to n_1,n_2,...,n_k respectively.

Experiment 5: The effects of different methods of padding and pooling on the performance of the model are studied. The original model applies the average pooling technique combined with reflection padding operation. While in our experiments, we compare the PESQ scores by using maximum pooling layers and reflection padding layers. As shown in **Table** 5, the maximum pooling technique slightly improves the performance of the MelGan model, and it is noticeable that the model using replication padding technique achieves better performance than the original model. However, combining maximum pooling operation and replication padding technique does not generate better audio quality.

Table 5: Effects of Padding and Pooling Function

	Baseline	w/ max pooling	w/ replication padding	w/ max+replication[1]
PESQ-MOS	1.750	1.881	2.053	1.931
MOS-LQO	1.457	1.542	1.675	1.608

[1] 'max+replication' means combination of max pooling technique and replication padding method.

As is discussed in the related work section, pooling techniques aim to 'shrink' (summarize) the feature locally while padding operations 'expand' the feature to catch more details of information. These two operations act in the opposite direction, which is why combining them cannot make an improvement.

4.2.3 Optimizer

Experiment 6: Initially, we deploy the original Adam optimizer and vary the value of learning rate from 10^{-2} to 10^{-6}, the results can be seen in **Table** 6. The reason is that for a large learning rate it is hard to converge due to its large step size of gradient descent update, while if the learning rate is small it needs more epochs and iterations to approach the local minima. That is why our results are better when the learning rate is close to 10^{-4}, which is the most appropriate value for here.

Table 6: learning rate comparison

	Baseline[1]	$lr = 10^{-2}$	$lr = 10^{-3}$	$lr = 5 \times 10^{-4}$	$lr = 10^{-5}$	$lr = 10^{-6}$
PESQ-MOS	1.750	0.434	0.455	1.733	0.834	0.449
MOS-LQO	1.457	1.070	1.072	1.447	1.126	1.072

[1] Baseline has learning rate (lr, for short) = 10^{-4}

Moreover, the generator with a large learning rate such as 10^{-2}, generates audio files that are polluted with deep and chaotic noise, while the generated file from the generator with less learning rate like 10^{-5} contains sharper and high-frequency noise, compared to file generated by the baseline model.

Experiment 7: Then we try different optimizers and the result is compared with the baseline, which is shown in the **Table** 7. The overall performances are worse than that of the baseline. One reason is that some optimizers (e.g. SGD) relay on the concave loss function to converge to the optimum point, which is not the case in the GANs. The other reason is the hyper-parameters: each optimizer contains too many hyperparameters that need to be tuned(e.g. learning rate, momentum), especially for SGD and RMSPROP.

Table 7: Optimizer comparison

	Baseline(Adam)	SGD	Rprop	RMSprop
PESQ-MOS	1.750	0.885	1.026	1.064
MOS-LQO	1.457	1.14	1.167	1.176

4.2.4 Loss function

Experiment 8: As a method of evaluating the audio quality, the Pearson correlation coefficient replaces L1 loss function as a feature matching loss function. The larger coefficient indicates a stronger correlation between the measured two quantity, therefore we choose the inverse of the coefficient as the feature matching loss function. However the results (**Table** 8) show that the Pearson correlation coefficient results in a low PESQ score. The reason is that the compacted range of coefficient is between -1 and +1, which is hard to describe the variation of the loss.

4.3 Improvement

Experiment 9: Based on the above experiments, we reduce the number of discriminators to 1 and try different architectures. As the **Table** 9 shows, replacing the average pooling technique with

Table 8: Pearson correlation coefficient

	Baseline(L1 Loss)	Pearson correlation coefficient[1]	L2 Loss
PESQ-MOS	1.750	1.479	0.806
MOS-LQO	1.457	1.317	1.128

[1] 'Peason Correlation is implemented as a feature matching loss function'

maximum pooling technique contributes to a comparable result, and using the replication padding method improves the PESQ score, while combining two techniques does not further improve the audio quality.

Additionally, adding one more extra residual block to the generator significantly improves the audio quality.

Table 9: Improvement (using 1 discriminator)

	Baseline (3 discs[6])	Baseline (1 disc)	MP[1] (1 disc)	RP[2] (1 disc)	MP+RP[3] (1 disc)	MP+4 RBs[4] (1 disc)	RP+4 RBs[5] (1 disc)
PESQ-MOS	1.750	1.685	1.732	1.915	1.701	2.015	2.254
MOS-LQO	1.457	1.387	1.426	1.692	1.388	1.787	1.892

[1] 'MP' means max pooling technique.
[2] 'RP' means replication padding method.
[3] 'MP+RP' means a combination of max pooling technique and replication padding method.
[4] 'MP+4 RBs' means a combination of max pooling technique and 4 residual blocks.
[5] 'RP+4 RBs' means a combination of replication padding technique and 4 residual blocks.
[6] 'n disc' means the model contains n discriminators.

5 Discussion and Conclusion

In this project, we successfully reproduce the task of the paper [1] and do experiments on different components of the original model on LJ Speech dataset. As explained above, due to the limitation of computational resources, we train 100 epochs instead of 3000 epochs in each experiment.

Accordingly, we show quantitative results obtained by introducing PESQ algorithm and briefly discuss the quality of those generated audio files. During the ablation study and these experiments, we discover that the number of residual blocks is more crucial to synthesize high quality audio than other factors. We also show that our modified models can improve the performance significantly (with replication padding operations on 4 residual blocks).

In particular, replacing the average pooling layer with max pooling layer and replacing reflection padding with replication padding improves the performance significantly, while combining them produces worse results. As discussed in the results section, these two operations act in the opposite direction, which is the reason why combining them cannot improve. How to balance these two techniques to achieve the best performance can be an investigation direction in the future.

The Pearson correlation coefficient can be used as a loss function during feature matching. However, it needs to be scaled to be able to describe a larger range of variation of the feature matching loss, which is why our attempts with the Pearson correlation coefficient do not improve the performance. How to properly use it can also be a future investigation topic.

6 Statement of Contribution

1. Yifei Zhao (260900278)
   yifei.zhao@mail.mcgill.ca
   code reproduction, code modification, code testing, report writing

2. Yichao Yang (260764629)
   yichao.yang@mail.mcgill.ca
   code reproduction, code modification, code testing, report writing
3. Yang Gao(260906623)
   yang.gao6@mail.mcgill.ca
   code reproduction, code modification, code testing, report writing

References

[1] T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, A. Courville, K. Kumar, and R. Kumaré, "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis", 2019.

[2] M. Binkowskié, "HIGH FIDELITY SPEECH SYNTHESIS WITH ADVERSARIAL NETWORKS", 2019.

[3] N. Vaishnavh and J. Z. Kolteré, "Gradient descent GAN optimization is locally stable", 2017.

[4] Y. N. Dauphin, H. de Vries, J. Chung, and Y. Bengioé, "RMSProp and equilibrated adaptive learning rates for non-convex optimization", 2015.

[5] X. Zeng, Z. Zhang, and D. Wangé, "AdaMax Online Training for Speech Recognition", 2016.

[6] D. Scherer, A. Muller, and S. B. é, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition", 2010.

[7] T. S. Harsha and D. B. Srinivasané, "Reconstruction of Flows using Convolutional neural networks", 2019.

[8] G. Liu, K. J. Shih, T.-C. Wang, F. A. Reda, K. Sapra, Z. Yu, A. Tao, and B. Catanzaroé, "Partial Convolution based Padding", in *arXiv preprint arXiv:1811.11718*, 2018.

[9] S. Xiang and H. Lié, "On the Effects of Batch and Weight Normalization in Generative Adversarial", 2017.

[10] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshidaé, "SPECTRAL NORMALIZATION FOR GENERATIVE ADVERSARIAL NETWORKS", 2018.

[11] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstraé, "PERCEPTUAL EVALUATION OF SPEECH QUALITY (PESQ) - A NEW METHOD FOR SPEECH QUALITY ASSESSMENT OF TELEPHONE NETWORKS AND CODECS", 2001.

[12] T. é, "Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models", *P.1401 : Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models*, 2012. [Online]. Available: `https://www.itu.int/rec/T-REC-P.1401-201207-I/en`.

7 APPENDIX

7.1 Training objective

To train the GAN, we use the hinge loss version of the GAN objective (Lim Ye, 2017; Miyato et al., 2018). We also experimented with the least-squares (LSGAN) formulation (Mao et al., 2017) and noticed slight improvements with the hinge version.

$$\min_{D_k} \mathbb{E}_x \left[\min\left(0, 1 - D_k(x)\right)\right] + \mathbb{E}_{s,z}\left[\min\left(0, 1 + D_k(G(s,z))\right)\right], \forall k = 1,2,3$$
$$\min_G \mathbb{E}_{s,z}\left[\sum_{k=1,2,3} -D_k(G(s,z))\right]$$

where *x* represents the raw wave form, *s* represents the conditioning information (eg. mel-spectrogram) and z represents the Gaussian noise vector.

7.2 Feature Matching

In addition to the discriminator's signal, we use a feature matching objective (Larsen et al., 2015) to train the generator. This objective minimizes the L1 distance between the discriminator feature maps of real and synthetic audio. Intuitively, this can be seen as a learned similarity metric, where a discriminator learns a feature space that discriminates the fake data from realdata. It is worth noting that we do not use any loss in the raw audio space. This is counter to other conditional GANs (Isola et al., 2017) where L1 loss is used to match conditionally generated images and their corresponding ground-truths, to enforce global coherence. In fact, in our case adding L1 loss in audio space introduces audible noise that hurts audio quality.

$$\mathcal{L}_{\mathrm{FM}}\left(G, D_k\right)=\mathbb{E}_{x, s \sim p_{\text{data}}}\left[\sum_{i=1}^{T} \frac{1}{N_i}\left\|D_k^{(i)}(x)-D_k^{(i)}(G(s))\right\|_1\right]$$

For simplicity of notation, $D_k^{(i)}$ represents the i^{th} layer feature mapping output of the k^{th} discriminator block, N_i denotes the number of units in each layer. Feature matching is similar to the perceptual loss. In the MelGan model, feature matching is applied to each intermediate layer of all discriminator blocks. The following objective is used to train the generator.

$$\min_{G}\left(\mathbb{E}_{s, z}\left[\sum_{k=1,2,3} -D_k(G(s, z))\right]+\lambda \sum_{k=1}^{3} \mathcal{L}_{\mathrm{FM}}\left(G, D_k\right)\right)$$