CONDITIONING TRICK FOR TRAINING STABLE GANS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper we propose a conditioning trick, called difference departure from normality, applied on the generator network in response to instability issues during GAN training. We force the generator to get closer to the departure from normality function of real samples computed in the spectral domain of Schur decomposition. This binding makes the generator amenable to truncation and does not limit exploring all the possible modes. We slightly modify the BigGAN architecture incorporating residual network for synthesizing 2D representations of audio signals which enables reconstructing high quality sounds with some preserved phase information. Additionally, the proposed conditional training scenario makes a trade-off between fidelity and variety for the generated spectrograms. The experimental results on UrbanSound8k and ESC-50 environmental sound datasets and the Mozilla common voice dataset have shown that the proposed GAN configuration with the conditioning trick remarkably outperforms baseline architectures, according to three objective metrics: inception score, Fréchet inception distance, and signal-to-noise ratio.

1 INTRODUCTION

Generative models have been widely used in several audio and speech processing tasks such as speaker verification (Reynolds et al., 2000), enhancement (Chehrehsa & Moir, 2016), synthesis (Raitio et al., 2010), etc. In the last few years, many generative adversarial network (GAN) architectures have been introduced (Bollepalli et al., 2019; Sriram et al., 2018; Hosseini-Asl et al., 2018; 2019) and they have contributed for tackling such challenging tasks. Furthermore, GANs have been employed in high-level data augmentation for supervised, semi-supervised, and unsupervised audio and speech classification (Hu et al., 2018; Donahue et al., 2018). In augmentation with paired transformations, cycle-consistent GANs have been developed for environmental sound classification (Esmaeilpour et al., 2020b) as well as for more sophisticated tasks such as voice conversion (Fang et al., 2018). A typical GAN runs a minimax game between generator and discriminator networks, where the latter should distinguish the real sample from the generated example. Following such a baseline GAN, many other variants in terms of similarity metric, loss function, and architectures have been introduced such as the Wasserstein GAN (Arjovsky et al., 2017), GANs with least squares loss (Mao et al., 2017), attention-based architectures for adversarially training (Zhang et al., 2019), etc. However, training and tuning such advanced GAN models have always been a mounting concern and a difficult challenge due to their instability in training (Salimans et al., 2016; Arjovsky & Bottou, 2017; Yang et al., 2017).

Our main contribution in this paper is in response to this issue. We propose to condition the generator for minimizing dissimilarity among generated and real samples using a constraining metric, called difference departure from normality (DDFN), computed in the spectral domain of Schur decomposition (Van Loan & Golub, 1983). This binding improves stability of the generator and does not limitate exploring all the possible modes during training. Without loss of generality, our focus in this paper is evaluating the performance of the DDFN metric in comparison with other regularization approaches on yielding more stable GANs for synthesizing high quality audio/speech signals. However, we also report some results on image generation in Appendix A.

As a common practice in audio/speech processing domain, which often uses 2D representations (spectrograms) over 1D signals (e.g., DeepSpeech (Hannun et al., 2014) and Kaldi (Povey et al., 2011) systems), our synthesis approach is also based on spectrograms. Technically, spectrograms are generated from the magnitude of a time-domain signal, while its phase information is discarded

(Lang & Forinash, 1998). Spectrogram either from short-time Fourier transformation (STFT) and its variants such as Mel-frequency cepstral coefficients (Hossan et al., 2010) or discrete wavelet transform (DWT) is a standard and common signal representation in signal processing due to its lower dimensionality which densely encodes local frequency information (Mallat, 1999; Rioul & Vetterli, 1991). Therefore, algorithms require much less training parameters on spectrograms and advanced architectures developed in the computer vision domain can be easier generalized to them compared to 1D signals. Signal synthesis with spectrograms also enquires phase matrices. Since our main goal in this paper is synthesizing high quality signals with less audible noises, we do not use any phase vector approximation algorithm. Instead of that, we use original phase vectors derived from real samples available in the datasets. Therefore, for synthesizing high quality signals, we need to synthesize high quality spectrograms.

For objectively measuring the impact of our conditioning trick using the DDFN metric on the baseline models and our slightly modified BigGAN architecture (Brock et al., 2019), we compute both the inception score (IS) (Salimans et al., 2016) and Fréchet inception distance (FID) (Heusel et al., 2017). Then, we reconstruct audio signals with some preserved original phase information and measure the signal-to-noise ratio (SNR). Our experimental results on two environmental sound datasets and on a subset of Mozilla common voice dataset have shown considerable improvement in generating diverse signals with high fidelity. This paper is organized as follows. In Section 2, we review some related works and in Section 3, we explain theories for constraining the generator. We provide our experimental results and associated discussions in Section 4.

2 BACKGROUND

The generator network $G(\mathbf{z}; \theta_g)$ commonly learns to map from a random probability distribution $p_z \sim \mathcal{N}(0, I)$ or $\mathcal{U}[-1, 1]$ to the generator probability domain p_g for $\mathbf{z} \in \mathbb{R}^{d_z}$ and the discriminator network $D(\mathbf{x}; \theta_d)$ should maximize $\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\left[\log\left(1 - D(G(\mathbf{z}))\right)\right]$ against the generator as defined in (1) (Goodfellow et al., 2014).

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}\left[\log D(\mathbf{x})\right] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\left[\log\left(1 - D(G(\mathbf{z}))\right)\right] \tag{1}$$

where θ_g and θ_d denote the training parameters of the generator and discriminator networks, respectively. These networks are often modeled by convolutional neural networks with different architectures (Radford et al., 2015). The optimization problem of (1) not only requires carefully-tuned hyperparameters, but also is very unstable and often collapses during training (Salimans et al., 2016; Thanh-Tung et al., 2019). Generally, there are two approaches to address this issue. First, changing the optimization functions mainly for the generator (Fedus et al., 2017; Zhang et al., 2019; Karras et al., 2018; Dumoulin et al., 2016; Nowozin et al., 2016; Salimans et al., 2016; Chen et al., 2016; Sønderby et al., 2016; Odena et al., 2017). Likewise, changing the similarity metric from Jensen-Shannon divergence (JSD) to Wasserstein loss (Arjovsky et al., 2017) and Pearson χ^2 divergence in least-square GAN (Mao et al., 2017) have also been proposed. Second, constraining the discriminator network to provide meaningful gradients everywhere to $G(\mathbf{z}; \theta_g)$ (Mescheder et al., 2018; Miyato et al., 2018; Gulrajani et al., 2017; Kodali et al., 2017).

There are two approaches that are relevant to our work for improving stability in GAN training. The first one is focused on bijective mapping between p_g and real sample distribution p_r using another network such as an autoencoder (Donahue et al., 2016). This ensures a correlation between generated and real samples through a regularization function similar to $\min \|\mathbf{z} - Rec(G(\mathbf{z}))\|_2 + H(\mathbf{z}, Rec(\mathbf{x}))$ where Rec denotes the reconstructor network (autoencoder) and H is the entropy loss (Srivastava et al., 2017). Besides, some variational autoencoder schemes have been also proposed for avoiding instability (Kingma & Welling, 2014; Rezende et al., 2014). The effectiveness of explicit regularization of loss functions with autoencoder loss has been studied by Che et al. (2016). They have introduced several costly metrics for estimating modes and enhancing quality of the generated samples. A similar metric for encoding random samples from p_r to p_g has been developed by Larsen et al. (2015). Both of these approaches incorporate $\mathcal{L}\left[\mathbf{x}, G(Rec(\mathbf{x}))\right]$ which is a pixel-wise loss followed by another regularization term. The second approach is spectral normalization (Miyato et al., 2018) which conditions the discriminator to support Lipschitz continuity using singular value decomposition. This approach regularizes θ_d (the weight matrix of the discriminator) towards the direction of the top rank (first) singular value. Inspired by Zhang et al. (2019), it has been shown

that this regularization can be more effectively implemented for $G(\mathbf{z};\theta_g)$ as (Brock et al., 2019):

$$\theta_g = \theta_g - \max(0, \sigma_0 - \sigma_{clamp}) v_0 u_0^\top \tag{2}$$

where θ_g denotes the weight matrix of the generator and $v_0\sigma_0 u_0^\top$ forms the first basis matrix of θ_g after decomposition. The threshold σ_{clamp} can be set to a predefined value (which implies an additional hyperparameter) or $\sigma_{clamp} = \mathrm{sg}(\sigma_1)$; where sg stands for the stop-gradient operation. This normalization can be generalized to other subsequent singular values, efficiently computed by Alrnoldi method (Golub & Van der Vorst, 2000).

While instability is a common problem when training GANs with any dataset, it is critical for 2D representations of audio and speech, mainly due to the properties of Fourier or wavelet transforms (Esmaeilpour et al., 2020b). In the next section, we explain how to control the generator for producing high fidelity spectrograms through correctly conditioning it.

3 CONDITIONING THE GENERATOR

In light of correlating p_g to p_r and spectral normalization in response to the instability issue in adversarial training, we condition $G(\mathbf{z};\theta_g)$ in another spectral domain to enhance the stability and also in order to provide better gradients everywhere to the discriminator. Our conditioning trick is fundamentally different from employing an autoencoder or regularizing either θ_g or θ_d through normalizing their singular value(s). Instead of normalizing the top rank eigenvalues of θ_g in spectral normalization (Zhang et al., 2019), we asymptotically correlate p_g to p_r in such a way that it forces the generator to follow Schur spectral distribution p_r. In fact, we force the generator to get closer to the departure from normality function computed for the original spectrograms in p_r defined in the spectral domain of the generalized Schur decomposition (Van Loan & Golub, 1983). This binding makes the generator amenable to the truncation trick (Brock et al., 2019) and the discriminator might converge in fewer iterations. Additionally, it neither limits spanning the entire possible modes nor loses sample variety.

Lemma. For an input sample (spectrogram) $\mathbf{x}_i$ from a given distribution, there exists a unitary representation $Q \in \mathbb{C}^{n\times n}$ in such a way that (Golub & Van Loan, 2012):

$$Q^H \mathbf{x}_i Q = V + S \tag{3}$$

where Q^H denotes the conjugate transpose of Q in vector space of Schur decomposition, $S = \{s_i \mid i = 0 : n-1\} \in \mathbb{C}^{n\times n}$ is an upper triangular matrix, and $V = \mathrm{diag}(\lambda_0, \lambda_1, \cdots, \lambda_{n-1})$ contains eigenvalues of $\mathbf{x}_i$ (λ_i denotes an independent eigenvalue for $\mathbf{x}_i$). In this unitary vector space, which might also yield a quasi-upper triangular representation for S, $Q = [q_0 \mid q_1 \mid q_2 \mid \cdots \mid q_{n-1}]$ provides the pencil of $\overrightarrow{q_i} - \lambda_i \overrightarrow{q_{i+1}}$ for $i \leq n-2$ which is also known as basis vector for $\mathbf{x}_i$. According to the perpendicularity of the achieved pencils and the support matrix S, we write (Golub & Van Loan, 2012):

$$\mathbf{x}_i q_k \approx \lambda_k q_k + \sum_{i=0}^{n-1} s_{ik} q_i, \quad k = 0 : n-1. \tag{4}$$

Therefore with the general assumption of quasi-upper triangular subspaces with the normal span of $\{q_0, q_1, \cdots, q_k\}$ for $k = 0 : n-1$, we can conclude that the choice of S should be independent of Q (Golub & Van Loan, 2012). Accordingly, we can compute its Frobenius norm using λ_i as:

$$\|S\|_F^2 = \|\mathbf{x}_i\|_F^2 - \sum_{i=0}^{n-1} |\lambda_i|^2 \equiv \Delta^2(\mathbf{x}_i) \tag{5}$$

where Δ^2 is known as departure from normality (DFN (Golub & Van Loan, 2012)). For ensuring the correlation between $\mathbf{x}_r$ and $\mathbf{x}_g$ randomly drawn from p_r and p_g in their designated vector spaces (span of q_is), we should expect $\left|\Delta^2(\mathbf{x}_j) - \Delta^2(\mathbf{x}_i)\right| < \epsilon$ for a small enough ϵ. Such a condition which we refer to as difference departure from normality (DDFN) also ensures the consistency of corresponding pencils and contributes to the generalized form of Schur decomposition with $Q_k^H \mathbf{x}_r Z_k = R_k(V_k + S_k)$ and $Q_k^H \mathbf{x}_g Z_k = R_k(V_k + S_k)$, where $R_k = Q_k^H(\mathbf{x}_r \mathbf{x}_{g_k}^{-1} Q_k)$ and Z_k is also unitary and supports for $\lim_{i\to\infty}(Q_{k_i}, Z_{k_i}) = (Q, Z)$ (Golub & Van Loan, 2012). The intuition behind exploiting these basis vectors is providing pencils of $\overrightarrow{\mathbf{x}_{r_i}} - \lambda_i \overrightarrow{\mathbf{x}_{g_i}}$ for learning the

original distribution p_r by the generator. The derivable pencils are not necessarily normal in the span of their associated subspaces, however, their linear combination imparts p_r to p_g (in the closed form). Furthermore, diagonal values in V constitute coefficient of basis vectors (sub-pencils in their manifolds) and represent local properties of the given input sample.

Proposition. The DDFN metric in the form $\left|\Delta^2(\mathbf{x}_g) - \Delta^2(\mathbf{x}_r)\right| < \epsilon$ ensures the correlation of $\mathbf{x}_g \sim p_g$ and $\mathbf{x}_r \sim p_r$ in the spectral domain with a measurable error term ϵ.

Proof. Since $\Delta^2(\cdot)$ is differentiable in its designated subspaces, we can find an upper bound for ϵ. For all $x_i := \max(\text{diag}(S_i))$ we assume $g_r(x_i) = \Delta^2(\mathbf{x}_i) \in \mathbb{C}^{n+1}$ with degree $n+1$ and $g_g(x_i) = \Delta^2(G(\mathbf{z}_i)) \in \mathbb{C}^n$ with degree n are differentiable over the interval $[\varpi_{inf}, \varpi_{sup}]$, therefore we can approximate the error function as (Phillips, 2003):

$$e(x) = g_r(x) - g_{g,n}(x) = \frac{g_r^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \tag{6}$$

where $\xi \in (\varpi_{inf}, \varpi_{sup})$ with the marginal condition using the second derivative $|\ddot{g}_r(x)| < \varrho$ for $0 \leq \varrho << 1$ and $n = 1$ we write:

$$g_r(x) - g_{g,1}(x) = \underbrace{(x - x_i)(x - x_{i+1})}_{g_f(x)} \frac{\ddot{g}_r(\xi)}{2!} \Rightarrow \dot{g}_f(x) = 2x - (x_i + x_{i+1}) = 0 \Rightarrow x = \frac{x_i + x_{i+1}}{2}$$

$$g_f\left(\frac{x_i + x_{i+1}}{2}\right) = -\left(\frac{x_{i+1} - x_i}{2}\right)^2 \Rightarrow |e(x)| \leq \frac{\ddot{g}_r(\xi)}{8} (x_{i+1} - x_i)^2 \quad \square \tag{7}$$

Herein, $|e(x)|$ measures how the dynamic $\Delta_g^2(\cdot)$ is close to the static $\Delta_r^2(\cdot)$ and equivalently measures the correlation between p_g and p_r. We experimentally demonstrate that minimizing the generator $G(\mathbf{z})$, subject to:

$$\left|\mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} \Delta^2(G(\mathbf{z})) - \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})} \Delta^2(\mathbf{x})\right| < \epsilon \tag{8}$$

yields high quality spectrograms and improves stability.

3.1 Fast DFN Approximation

Computing $\sum_{i=0}^{n-1} |\lambda_i|^2$ as defined in (5) can be computationally prohibitive since it involves recursive multiplication of pencils. In response to this issue, $f_V(\lambda_i)$ can be defined as a polynomial function originating from the absolute product of eigenvalues (Edelman, 1993; Van Loan & Golub, 1983):

$$\sum_{i=0}^{n-1} |\lambda_i|^2 \cong \sum_{i=0}^{n-1} \left(\prod_{i=0}^{n-1} f_V(\lambda_i) \prod_{i \neq j}^{n-1} |\lambda_i - \lambda_j| \underbrace{e^{-\sum \lambda_i^2/2}}_{\text{Normal distribution}} \right) \tag{9}$$

where it imparts a normal distribution and closely relates to the joint density function of the eigenvalues for a random symmetric matrix $\mho_{k \times k} := (\varrho + \varrho^\top)/2$. We derive ϱ by downsampling a given input spectrogram (i.e., $\mathbf{x}_{n \times n}$) by a factor of two. Every single element of the derived Toeplitz matrix ($\mho$) should be also distributed according to a normal distribution (Van Loan & Golub, 1983). The probability distribution of $\mho$ which is also known as Gaussian orthogonal ensemble (Alt et al., 1995; Wu et al., 1990) for the eigenvalues of λ_i has the form of (Edelman, 1993):

$$\frac{2^{-k/2}}{\prod_{i=1}^{k} \nu(i/2)} \prod_{i \neq j}^{k-1} |\lambda_i - \lambda_j| \, e^{-0.5 \sum \lambda_i^2} \tag{10}$$

where $\nu(\cdot)$ denotes Minkowski function (Panti, 2008; Thompson & Thompson, 1996). Accordingly, the summation of (9) for fast approximation (5) is the expectation over the determinants of $\mho$ as (Edelman, 1993):

$$\frac{1}{k!} \sum_{i=0}^{n-1} \left(\prod_{i=0}^{n-1} f_V(\lambda_i) \prod_{i \neq j}^{n-1} |\lambda_i - \lambda_j| \, e^{-\sum \lambda_i^2/2} \right) \approx \left\{ 2^{k/2} \prod_{i=0}^{k-1} \nu(i/2) \right\} \mathbb{E}_{\mho} \det \left(\underbrace{\mho}_{f_V(\lambda_i)} \right) \tag{11}$$

Although there are some analytical approaches for closely approximating $f_V(\lambda_i)$, we simply use $\mho$ since it is normally distributed over $\mathbf{x}$ and $\lambda_i \to 0$ for $i > 0.5 \cdot k$. Therefore $\det(\mho)$ gives an acceptable approximation.

3.2 INTUITION FOR THE DDFN METRIC

The DDFN metric as shown in (8) regularizes the generator network to craft spectrograms within the manifolds of real samples which consequently binds its learning behavior and results to better stability. The DFN metric (5) computes a single real value for a spectrogram relative to its eigenvalues in the subspace of Schur decomposition. According to (5), the DFN measures the depth between the Frobenius norm of a spectrogram (mean energy (Golub & Van Loan, 2012)) and the summation of all its possible eigenvalues (which encode structural components (Golub & Van Loan, 2012)). This depth is in fact a single-value descriptor for a spectrogram representing its geometrical position in a manifold. Therefore, the more similar two spectrograms become, the closer their DFN values (descriptors) are expected so that lie in the same manifold. In other words, minimizing over the DDFN metric, forces the generator to produce spectrograms within the manifold of real spectrograms.

Constraining the generator using the DDFN metric has some advantages. First, unlike the spectral normalization, it does not manipulate weight vectors of the model. Therefore, it does not negatively affect the computational complexity of the model during backpropagation in each epoch. Second, we compute this metric for all the training samples and this helps to span all possible varieties in the dataset. Third, the value for ϵ in (8) is justifiable according to the required quality for the spectrogram. This helps to make a better trade-off between more samples variety (by increasing ϵ) and higher quality (by decreasing ϵ). In Section 4, we experimentally prove the effectiveness of employing DDFN metric on GAN architectures for improving stability.

4 EXPERIMENTAL RESULTS

In this section, we provide details of our experiments on two benchmarking environmental sound datasets: UrbanSound8K (Salamon et al., 2014) and ESC-50 (Piczak, 2015). We have also conducted a brief ablation study for Mozilla common voice (MCV)[1]. The first two datasets include 8,732 and 2,000 short environmental audio signals ($\leq$5 sec) organized in 10 and 50 different classes, respectively. MCV consists of 4,257 recorded hours of multi-language speeches ($\leq$7 sec) and the corresponding text transcriptions. For generating DWT spectrograms, we use complex Morlet mother function with static sampling frequency of 16 kHz and frame length of 50 ms with 50% overlapping. We represented each spectrogram using three different visualizations of linear, logarithmic, and logarithmic-real magnitude scales (see Appendix D)(Esmaeilpour et al., 2020a). For STFT spectrograms, we set the number of filters to 2,048 with the hop length of 1,024 and reflect padding in overlapping audio chunks of 50 ms (with ratio 0.5) (Esmaeilpour et al., 2020a). Meanwhile, we apply pitch shifting (Salamon & Bello, 2017) with scales $0.75, 0.9, 1.15$, and 1.5 for audio signals with length ≤ 2 seconds for data augmentation (Esmaeilpour et al., 2020).

Our baseline model is the SA-GAN architecture (Zhang et al., 2019) using the hinge loss objective for smooth training (Lim & Ye, 2017). Since we employ class-conditional (CC) learning similar to Mirza & Osindero (2014), the generator receives CC batch-norm as suggested by De Vries et al. (2017). For the discriminator, we run the identical procedure with projection (Miyato & Koyama, 2018) following the same optimization procedure in the baseline SA-GAN, however with some modifications in the number of channels and applying spectral regularization for the generator network. We have evaluated different options for initializing both networks, such as Glorot (Glorot & Bengio, 2010) and variants of $\mathcal{N}(0, I)$ (e.g., $\mathcal{N}(0, 0.02I)$) (Radford et al., 2016). Since this choice noticeably affects the training performance (Brock et al., 2019) and requires to be evaluated individually according to properties of the dataset, we eventually used orthogonal regularization (Saxe et al., 2014).

Figure 1 depicts the performance of four GAN architectures trained on logarithmic DWT spectrograms. All these models undergo collapse at different iterations which forces to early stop the training. All the subsequent quality measurements on the performance of these models will be conducted on the checkpoints prior to collapse. We compute the $\left|\Delta^2(\mathbf{x}_g) - \Delta^2(\mathbf{x}_r)\right|$ for each model to demonstrate collapse without imposing the DDFN condition on their generators (see (8)). These plots corroborate the positive impact of spectral normalization (Zhang et al., 2019) on delaying collapse at further iterations. The best performance of this normalization is when it is applied only on

[1] https://voice.mozilla.org/en/datasets

$D(\mathbf{x};\theta_d)$ (see Figures 1(c) and 1(d)), though incorporating it into both $G(\mathbf{z};\theta_g)$ and $D(\mathbf{x};\theta_d)$ still outperforms the baseline SA-GAN.

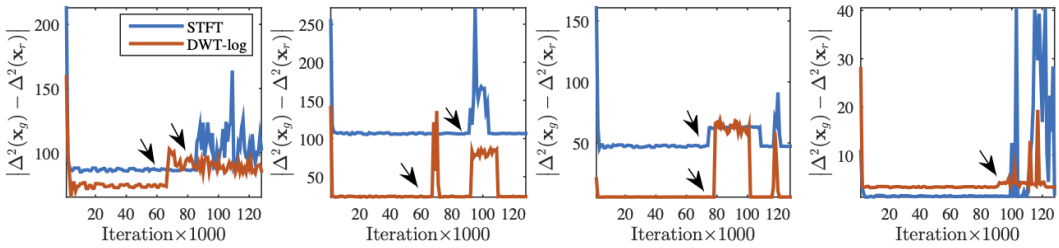


Figure 1: The typical plot of the DDFN measures for a random non-convolution layer in $G(\mathbf{z};\theta_g)$ trained on STFT and logarithmic DWT representations of UrbanSound8K dataset. (a) The baseline SA-GAN (Zhang et al., 2019), (b) the BigGAN architecture (Brock et al., 2019) with spectral normalization in both $G(\mathbf{z};\theta_g)$ and $D(\mathbf{x};\theta_d)$, (c) the BigGAN configuration with spectral normalization only in $D(\mathbf{x};\theta_d)$, and (d) the scaled up (by a factor of 2) configuration for (c). Arrows in these sub-figures refer to collapse onsets.

Furthermore, we have evaluated the impact of increasing the batch size on the performance of GANs during training while monitoring the DDFN measure along iterations. We scaled up the batch size to 256 and 512 aiming at covering more modes and providing better gradients to both $G(\mathbf{z};\theta_g)$ and $D(\mathbf{x};\theta_d)$ (Brock et al., 2019). Figure 1(d) depicts this effect for the BigGAN with batch size of 512 and regularized θ_d using spectral normalization. According to this graph although this GAN keeps its stability until about 100k iterations, it undergoes complete collapse afterward. Moreover, the DDFN measure is considerably reduced compared to other three GANs which indicates convergence in fewer iterations. We additionally scaled up the number of channels (width) in each layer of the networks by a factor of 2 at the cost of doubling the number of required parameters. However this did not rectify nor delay the collapse at higher iterations.

We slightly modify the BigGAN architecture for class conditional (CC) training (see Appendix C). Three major settings in CC training are required to adapt our benchmarking datasets. Firstly setting the conditional vector (c-embedding) and secondly, the skip connection (direct skip-z) from the given noise vector according to the designated probability distribution. While using separate layers for c-embeddings has been proposed by Miyato et al. (2018), we also found shared embeddings (Perez et al., 2018; Brock et al., 2019) outperforms the latter. For the skip-z connection probe, we firstly evaluated splitting $z \sim \mathcal{N}(0, I)$ and $z \sim \mathcal{N}(0, 0.02I)$ vectors into smaller chunks (we set to 20) and concatenating them into c-embeddings in each level of resolution as suggested by Brock et al. (2019), then explored a couple of other variants (Denton et al., 2015; Goodfellow et al., 2014). We ended up to concatenate the vector z in its entire dimension to c-embeddings (Brock et al., 2019). Finally, the third setting is the number of units in each hidden layer of the networks (the channel multiplier). Increasing this hyperparameter relatively affects the IS and FID scores for the model.

Our objective evaluations are summarized in Table 1 and the number of iterations without collapse indicate considerable stability improvement compared to the baseline GANs depicted in Figure 1. This table also shows that for the majority of the cases, the DDFN outperforms the orthogonal regularization (Brock et al., 2017) in delaying potential collapse. Additionally, it compares the effect of batch size and number of channels on the performance of our slightly modified BigGAN architecture using spectral normalization and our introduced DDFN measure. According to this table, doubling the batch size considerably improves both the IS and FID. Among different configurations for the GAN, models trained on three visualizations of DWT spectrograms dominantly outperform the STFT. We conjecture that this is due to the complexity of Morlet mother function (Young, 2012) compared to sinusoidal transform in STFT.

4.1 Orthogonal Regularization vs. DDFN

The orthogonal regularization (Brock et al., 2017) has been proposed for smooth training, which to some extent, makes a better trade-off between sample variety and fidelity by constraining over θ_d

Dataset	Orthogonal Regularization	Iteration $\times 1000$	IS	FID
US8K	False ($batch = 256$)	$132, 136$ ($ch. = 64$)	43.12 ± 1.46 ($\downarrow 3.50, \uparrow 1.14, -$)	44.57 ± 2.93 ($\uparrow 1.67, \downarrow 2.95, \uparrow 22.23$)
	True ($batch = 256$)	$138, 144$ ($ch. = 96$)	46.89 ± 0.12 ($\downarrow 2.64, \uparrow 1.54, -$)	29.06 ± 1.18 ($\uparrow 5.28, \downarrow 6.71, \downarrow 5.35$)
	True ($batch = 256$)	$106, 108$ ($ch. = 96$)	38.55 ± 1.09 ($-, \downarrow 1.18, \uparrow 0.53$)	49.91 ± 1.52 ($\downarrow 2.36, \uparrow 1.82, -$)
	False ($batch = 512$)	$134, 147$ ($ch. = 64$)	49.61 ± 0.12 ($\downarrow 2.41, \uparrow 1.78, -$)	$\mathbf{21.75 \pm 0.31}$ ($\downarrow 2.09, \downarrow 3.14, -$)
	True ($batch = 512$)	$133, 151$ ($ch. = 96$)	$\mathbf{52.36 \pm 1.19}$ ($\uparrow 1.94, \uparrow 2.72, \downarrow 27.19$)	21.93 ± 1.11 ($\downarrow 1.40, -, \uparrow 14.18$)
	True ($batch = 512$)	$107, 110$ ($ch. = 96$)	40.76 ± 0.69 ($\uparrow 1.03, \downarrow 0.93, -$)	33.56 ± 1.25 ($\downarrow 1.33, \downarrow 2.39, \uparrow 1.05$)
ESC-50	False ($batch = 256$)	$148, 152$ ($ch. = 64$)	68.76 ± 3.14 ($\uparrow 0.64, \uparrow 5.63, -$)	34.61 ± 2.23 ($\downarrow 2.16, \uparrow 1.77, -$)
	True ($batch = 256$)	$143, 156$ ($ch. = 96$)	71.94 ± 1.42 ($-, \uparrow 7.91, \downarrow 19.68$)	29.81 ± 3.15 ($\uparrow 3.62, \downarrow 2.13, \uparrow 2.73$)
	True ($batch = 256$)	$107, 115$ ($ch. = 96$)	68.79 ± 1.37 ($\downarrow 1.22, \uparrow 2.27, \downarrow 4.66$)	34.09 ± 0.85 ($\uparrow 3.21, \downarrow 2.76, \downarrow 1.79$)
	False ($batch = 512$)	$139, 155$ ($ch. = 64$)	71.46 ± 0.26 ($\downarrow 1.47, \downarrow 2.38, -$)	29.03 ± 1.05 ($\downarrow 6.91, \downarrow 4.63, -$)
	True ($batch = 512$)	$145, 161$ ($ch. = 96$)	$\mathbf{73.69 \pm 1.47}$ ($\uparrow 3.55, \uparrow 1.95, -$)	$\mathbf{23.03 \pm 2.47}$ ($\uparrow 2.79, \uparrow 1.31, \uparrow 21.84$)
	True ($batch = 512$)	$109, 116$ ($ch. = 96$)	69.81 ± 2.45 ($\uparrow 1.19, \downarrow 1.87, \downarrow 7.45$)	36.07 ± 2.68 ($\downarrow 1.29, -, \uparrow 1.32$)

Table 1: Comparison of the inception score (higher is better) and Fréchet inception distance (lower is better) for our proposed modification to BigGAN architecture trained on logarithmic DWT spectrograms (resolution 128×128). Unhighlighted rows refer to the models with running spectral normalization and DDFN on $D(\mathbf{x}; \theta_d)$ and $G(\mathbf{z}; \theta_g)$, respectively. Highlighted rows denote the models with orthogonal regularization and spectral normalization without DDFN. Scores are averaged over 5k generated spectrograms over 10 different runs for each dataset. The left values in the iteration column are associated with the highest achieved IS/FID before initial collapse, and the right values refer to the extreme collapse onsets.Values inside parenthesis in the last two columns correspond to linear-DWT, logarithmic-real-DWT, and STFT representations. Accordingly, up-/down-ward arrows denote increase/decrease in scores relative to the logarithmic DWT. The best results are shown in boldface.

as:

$$R_\beta = \beta \left\| \theta_d^\top \theta_d - I \right\|_F^2 \tag{12}$$

where β is a small hyperparameter. This regularization prohibitively affects the generator and results to early collapse at about 25k iterations while trained on logarithmic DWT spectrograms. Recently an improved version of this regularization has been introduced (Brock et al., 2019):

$$R_\beta = \beta \left\| \theta_d^\top \theta_d \odot (\mathbf{1} - I) \right\|_F^2 \tag{13}$$

where $\mathbf{1}$ refers to a matrix with constant values of 1. This regularization term binds to minimize local similarity among filters. In our experiment we found out $\beta \in (10^{-4}, 10^{-5}]$ is the best range for STFT and DWT spectrograms except for the logarithmic-real. Upon running several exploratory experiments, we set $\beta = 10^{-4}$ for it. Unlike orthogonal regularization, the DDFN does not involve weight matrix manipulations at each layer. This considerably reduces the computational cost for training and makes the discriminator converge in fewer iterations and remarkably delays the collapse at higher iterations. Table 1 shows the positive effect of orthogonal regularization while used with minimized DDFN measure (see 8).

4.2 Spectrogram Fidelity and variety

Similar to Brock et al. (2019), our modification to the BigGAN architecture makes the model amenable to truncation for $p_r \sim \mathcal{N}(0, \alpha I)$ where the threshold $\alpha \in [0, 1)$. Smaller values for α

negatively affects the total number of spectrogram variety however, considerably boosts the quality of the generated samples. This trade-off (see a relevant study in Marchesi (2017)) also affects the actual value of ϵ in computing the DDFN measure (8). Averaged over different experiments on logarithmic DWT representations, for $\alpha \leq 0.5$ we achieve $\epsilon \leq 37.16$. This indicates higher quality of the generated spectrogram at the cost of losing sample variety. Likewise, increasing ϵ expands the sample variety but with degraded sample quality (see Figure 2). Our GAN architecture for $\alpha \leq 0.2$ generates oversmoothed spectrograms as a result of a poorly conditioned model and increases (explodes) ϵ toward higher values.

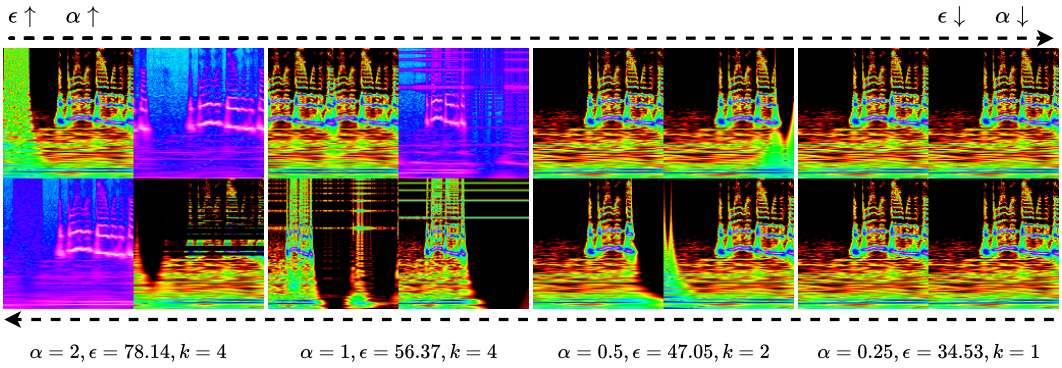


Figure 2: Generated logarithmic DWT spectrograms using our modified BigGAN model with spectral normalization and the DDFN measure. From left to right: decreasing ϵ and α makes $G(\mathbf{z}; \theta_g)$ to generate spectrograms with higher qualities and lower varieties. In this typical example, k denotes the total number of spectrogram variety in each batch.

The truncation trick directly increases the IS and specifically for $\alpha = 0.25$ we noticed 12.3% improvement in generating high quality logarithmic DWT spectrograms compared to $\alpha = 2$ or $\mathcal{U}[-1, 1]$. Moreover, a slight reduction in α positively impacts the FID by reducing the score 10% on average, however, when $\alpha \to 0$ the FID sharply increases. Whereas FID, the IS does not penalize sample variety, however, it rewards precision and this turns out to be a biased objective evaluation on the quality of generated spectrograms (Brock et al., 2019). Moreover, interpretation and quality analysis of the generated spectrograms could be very difficult to human eyes. To address this issue, we reconstruct the audio signals from them and measure the SNR. This requires access to the phase information for each spectrogram (Koerich et al., 2020) and unfortunately we could not successfully train our model with phase vectors. Although, there are some approaches for phase approximation (Leeb & Henk, 1989; Mulgrew, 2013) or reconstructing signal almost without phase information (Kumar et al., 2019) using GAN, we noticed they often introduce ambient and background noises to the reconstructed signal (see Appendix D). We eventually opted to utilize subjective phase vectors from original known samples. Figure 3 shows that spectrograms with lower values of α and ϵ better reconstruct audio signals in terms of quality (see Appendix D for additional information).

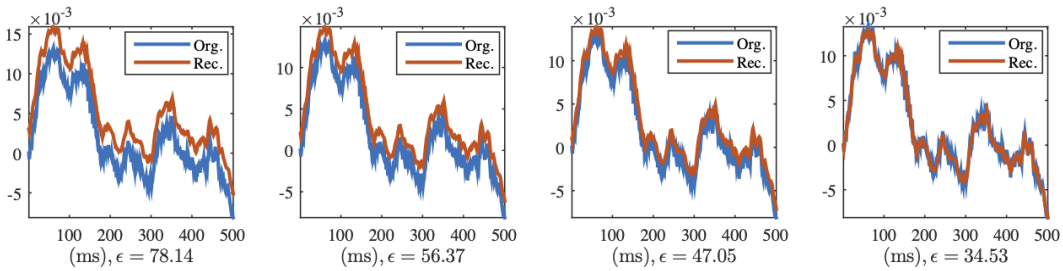


Figure 3: Comparison of original and reconstructed sounds from the generated logarithmic DWT spectrograms with identical phase matrix randomly drawn from an original speech signal from MCV. For better visualization we have shown the first 500 ms from the entire three second-length signal.

Dataset	Model	Res.	IS	FID	SNR_{db}
MCV	SA-GAN (baseline)	128	–	59.14 ± 2.24 ($\downarrow 4.54, \downarrow 1.46, -$)	41.91
	BigGAN	128	–	54.95 ± 1.04 ($\uparrow 2.14, \downarrow 3.48, -$)	46.93
	BigGAN (+DDFN)	128	35.57 ± 2.92 ($\uparrow 1.02, \uparrow 3.46, -$)	51.01 ± 2.18 ($\downarrow 3.52, \downarrow 2.11, -$)	53.72
	Proposed (+DDFN)	128	50.79 ± 0.14 ($\downarrow 3.30, \downarrow 2.48, -$)	43.37 ± 1.65 ($\downarrow 6.88, \downarrow 4.99, -$)	56.91
		256	$\mathbf{56.20 \pm 1.73}$ ($\downarrow 4.03, \downarrow 2.95, -$)	$\mathbf{36.31 \pm 3.63}$ ($\downarrow 7.59, \downarrow 4.36, \uparrow 23.48$)	**62.89**
US8K	SA-GAN (baseline)	256	34.72 ± 0.23 ($\uparrow 1.17, \uparrow 2.39, -$)	–	34.12
	BigGAN	256	39.16 ± 5.07 ($\downarrow 4.56, \downarrow 3.94, -$)	37.34 ± 3.53 ($\downarrow 5.21, \downarrow 2.75, \downarrow 16.18$)	48.14
	BigGAN (+DDFN)	256	48.62 ± 0.23 ($-, \uparrow 2.39, -$)	25.63 ± 2.42 ($\uparrow 2.17, \uparrow 0.94, -$)	55.17
	Proposed (+DDFN)	256	$\mathbf{57.91 \pm 2.19}$ ($\downarrow 7.48, \downarrow 4.88, \downarrow 15.34$)	$\mathbf{16.33 \pm 1.87}$ ($\downarrow 3.42, \downarrow 4.88, -$)	**63.24**

Table 2: Comparison of average IS, FID and SNR for different models trained on logarithmic DWT spectrograms. Outperforming values are in bold. Spectral normalization applied only on $D(\mathbf{x}; \theta_d)$ for all the BigGANs and its modified variant (the proposed architecture). Res. stands for resolution. For supplementary results see Appendix B.

4.3 Summary and Ablation Study

In this section we compare the best performance of our proposed model with other GANs considering 256×256 and 128×128 input resolutions, as shown in Table 2. We use the ablated model for MCV dataset on 100k randomly selected recordings shorter than five seconds. Moreover, we objectively compare the quality of the reconstructed signals using the SNR[2] (Kereliuk et al., 2015) A high SNR means the reconstructed signal has low noise. Table 2 summarizes the results on four spectrogram representations averaged over 5k generated spectrogram at 10 different runs for each dataset. Doubling the resolution to 256×256 improves both the IS and the FID, however roughly doubles the number of training parameters. Adding this note to the truncation trade-off, we can conclude that for synthesizing high quality sounds, we need to tune three major hyperparameters: α, large batch sizes, and high resolutions. Table 2 also shows the results of an ablation study on UrbanSound8K and the conclusions are similar to those on MCV.

5 Conclusion

In this paper, we have proposed a conditioning trick for the generator network based on the DDFN metric in the spectral domain of Schur decomposition. We have experimentally demonstrated its positive impact in improving model stability at larger iterations for both baseline self-attention and on our slightly modified BigGAN architecture for class-conditional learning. We have also shown that our training scenario makes the model amenable to truncation and this helps to make a trade-off between spectrogram quality and variety. According to three objective metrics of IS, FID, and SNR, models conditioned with the DDFN outperform baselines in generating high quality spectrograms and less noisy reconstructed audio and speech signals. Our proposed conditioning trick which has been essentially developed for audio/speech signal synthesis purposes, is generalizable to GANs on different modalities. We demonstrate the effectiveness of this approach on CIFAR-10 dataset for natural image generation as discussed in Appendix A.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-

[2] $\text{SNR}_{db}(\mathbf{x}_r, \mathbf{x}_g) = 20 \log_{10} Pw(\mathbf{x})/Pw(\mathbf{x})$ where $Pw(.)$ denotes the power of the signal.

scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283, 2016.

J. I. Agbinya. Discrete wavelet transform techniques in speech processing. In *Proceedings of Digital Processing Applications (TENCON '96)*, volume 2, pp. 514–519 vol.2, 1996.

H Alt, H-D Gräf, HL Harney, R Hofferbert, H Lengeler, A Richter, P Schardt, and HA Weidenmüller. Gaussian orthogonal ensemble statistics in a microwave stadium billiard with chaotic dynamics: Porter-thomas distribution and algebraic decay of time correlations. *Physical review letters*, 74 (1):62, 1995.

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. arxiv e-prints, art. *arXiv preprint arXiv:1701.04862*, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Mohammed Bahoura and Jean Rouat. Wavelet speech enhancement based on the teager energy operator. *IEEE signal processing letters*, 8(1):10–12, 2001.

Bajibabu Bollepalli, Lauri Juvela, and Paavo Alku. Generative adversarial network-based glottal waveform model for statistical parametric speech synthesis. *arXiv preprint arXiv:1903.05955*, 2019.

Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Cneural photo editing with introspective adversarial networks. In *International conference on machine learning*, 2017.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=B1xsqj09Fm`.

Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*, 2016.

Sarang Chehrehsa and Tom James Moir. Speech enhancement using maximum a-posteriori and gaussian mixture models for speech and noise periodogram estimation. *Computer Speech & Language*, 36:58–71, 2016.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pp. 6594–6604, 2017.

Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pp. 1486–1494, 2015.

Chris Donahue, Bo Li, and Rohit Prabhavalkar. Exploring speech enhancement with generative adversarial networks for robust speech recognition. In *IEEE Intl Conf on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5024–5028, 2018.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Alan Edelman. The circular law and the probability that a random matrix has k real eigenvalues. *preprint*, 1993.

M. Esmaeilpour, P. Cardinal, and A. L. Koerich. A robust approach for securing audio classification against adversarial attacks. *IEEE Transactions on Information Forensics and Security*, 15:2147–2159, 2020.

Mohammad Esmaeilpour, Patrick Cardinal, and Alessandro Lameiras Koerich. From sound representation to model robustness. *arXiv preprint arXiv:2007.13703*, pp. 1–12, 2020a.

Mohammad Esmaeilpour, Patrick Cardinal, and Alessandro Lameiras Koerich. Unsupervised feature learning for environmental sound classification using weighted cycle-consistent generative adversarial network. *Applied Soft Computing*, 86:105912, 2020b.

Fuming Fang, Junichi Yamagishi, Isao Echizen, and Jaime Lorenzo-Trueba. High-quality nonparallel voice conversion based on cycle-consistent adversarial network. In *IEEE Intl Conf on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5279–5283, 2018.

William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. *arXiv preprint arXiv:1710.08446*, 2017.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Gene H Golub and Henk A Van der Vorst. Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1-2):35–65, 2000.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2012.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

M. A. Hossan, S. Memon, and M. A. Gregory. A novel approach for mfcc feature extraction. In *2010 4th International Conference on Signal Processing and Communication Systems*, pp. 1–5, 2010.

Ehsan Hosseini-Asl, Yingbo Zhou, Caiming Xiong, and Richard Socher. A multi-discriminator cyclegan for unsupervised non-parallel speech domain adaptation. *arXiv preprint arXiv:1804.00522*, 2018.

Ehsan Hosseini-Asl, Yingbo Zhou, Caiming Xiong, and Richard Socher. Augmented cyclic adversarial learning for low resource domain adaptation. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=B1G9doA9F7`.

Hu Hu, Tian Tan, and Yanmin Qian. Generative adversarial networks based data augmentation for noise robust speech recognition. In *IEEE Intl Conf on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5044–5048, 2018.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Corey Kereliuk, Bob L Sturm, and Jan Larsen. Deep learning and music adversaries. *IEEE Transactions on Multimedia*, 17(11):2059–2071, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. In *2nd Intl Conf on Learning Representations, ICLR*, volume 19, 2014.

Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

K. M. Koerich, M. Esmaeilpour, S. Abdoli, A. S. Britto Jr., and A. L. Koerich. Cross-representation transferability of adversarial attacks: From spectrograms to audio waveforms. In *IEEE Intl J Conf on Neural Networks*, pp. 1–8, 2020.

Martin Krawczyk and Timo Gerkmann. Stft phase reconstruction in voiced speech for an improved single-channel speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1931–1940, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In *Advances in Neural Information Processing Systems*, pp. 14910–14921, 2019.

W Christopher Lang and Kyle Forinash. Time-frequency analysis with the continuous wavelet transform. *American journal of physics*, 66(9):794–797, 1998.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

F Leeb and T Henk. Simultaneous amplitude and phase approximation for fir filters. *International journal of circuit theory and applications*, 17(3):363–374, 1989.

Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *IEEE Intl Conf onComputer Vision (ICCV)*, pp. 2813–2821, 2017.

Marco Marchesi. Megapixel size image creation using generative adversarial networks. *arXiv preprint arXiv:1706.00082*, 2017.

Yoshiki Masuyama, Kohei Yatabe, Yuma Koizumi, Yasuhiro Oikawa, and Noboru Harada. Deep griffin–lim iteration. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 61–65. IEEE, 2019.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? *arXiv preprint arXiv:1801.04406*, 2018.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ByS1VpgRZ.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Bernard Mulgrew. The stationary phase approximation, time-frequency decomposition and auditory processing. *IEEE transactions on signal processing*, 62(1):56–68, 2013.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pp. 271–279, 2016.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pp. 2642–2651, 2017.

Giovanni Panti. Multidimensional continued fractions and a minkowski function. *Monatshefte für Mathematik*, 154(3):247–264, 2008.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3942–3951. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16528.

George M Phillips. *Interpolation and approximation by polynomials*, volume 14. Springer Science & Business Media, 2003.

Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proc. 23rd ACM international conference on Multimedia*, pp. 1015–1018. ACM, 2015.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06434.

Tuomo Raitio, Antti Suni, Junichi Yamagishi, Hannu Pulakka, Jani Nurminen, Martti Vainio, and Paavo Alku. Hmm-based speech synthesis utilizing glottal inverse filtering. *IEEE transactions on audio, speech, and language processing*, 19(1):153–165, 2010.

Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Olivier Rioul and Martin Vetterli. Wavelets and signal processing. *IEEE signal processing magazine*, 8(4):14–38, 1991.

J. Salamon, C. Jacoby, and J. P. Bello. A dataset and taxonomy for urban sound research. In *22nd ACM Intl Conf on Multimedia*, Orlando, FL, USA, Nov. 2014.

Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6120.

Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

Anuroop Sriram, Heewoo Jun, Yashesh Gaur, and Sanjeev Satheesh. Robust speech recognition using generative adversarial networks. In *IEEE Intl Conf on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5639–5643, 2018.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pp. 3308–3318, 2017.

Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. Improving generalization and stability of generative adversarial networks. *arXiv preprint arXiv:1902.03984*, 2019.

Anthony C Thompson and Anthony C Thompson. *Minkowski geometry*. Cambridge University Press, 1996.

C. F. Van Loan and G. H. Golub. *Matrix computations*. Johns Hopkins University Press, 1983.

Hua Wu, Michel Vallières, Donald WL Sprung, et al. Gaussian-orthogonal-ensemble level statistics in a one-dimensional system. *Physical Review A*, 42(3):1027, 1990.

Shan Yang, Lei Xie, Xiao Chen, Xiaoyan Lou, Xuan Zhu, Dongyan Huang, and Haizhou Li. Statistical parametric speech synthesis using generative adversarial networks under a multi-task learning framework. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 685–691, 2017.

Randy K Young. *Wavelet theory and its applications*, volume 189. Springer Science & Business Media, 2012.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pp. 7354–7363. PMLR, 2019.

A APPENDIX

Without loss of generality, the proposed conditioning trick is also applicable to GANs for image generation. Therefore, we run an ablation study on CIFAR-10 (Krizhevsky et al., 2009) dataset, which includes 6k samples organized into 10 classes. This dataset has been one of the common benchmarking datasets for GAN analysis.

Similar to Table 2, we measure the effectiveness of the DDFN metric in training three GAN configurations namely, SA-GAN, BigGAN, and our slightly modified BigGAN architecture (see Appendix C) without investigation for the amenability to truncation. Results are summarized in Table 3.

As shown in this table, the DDFN metric not only improves both IS and FID scores of the GAN architectures, but also it considerably delays collapse onsets for the models. The rest of the settings for these models is as explained in Appendix C.

Since images in this dataset have low resolution, it is very difficult to visually compare quality of the generated images with the ground-truth, however we have included some images in the supplementary material.

Model	IS	FID	Iteration $\times 1000$
SA-GAN	06.32 ± 0.41	27.36 ± 1.49	58.36
SA-GAN+DDFN	07.53 ± 0.08	23.19 ± 0.12	64.73
BigGAN	08.87 ± 0.03	18.91 ± 1.14	66.23
BigGAN+DDFN	09.14 ± 0.16	15.05 ± 0.51	71.04
Proposed	08.98 ± 0.04	14.98 ± 0.71	71.07
Proposed + DDFN	$\mathbf{09.31 \pm 0.17}$	$\mathbf{14.79 \pm 0.06}$	**79.38**

Table 3: Comparison of the inception score and Fréchet inception distance for three state-of-the-art GAN configurations on CIFAR-10 dataset. In all these experiments the batch size is set to 512. Moreover, the spectral normalization has been applied on all the discriminator networks. Score are averaged over 5k generated sample in 10 different runs. Better values are shown in boldface.

B APPENDIX

Table 4 is an extension of Table 2, which includes results for various resolutions. Both IS and FID scores for the resolution 128×128 are very competitive to 256×256. Since Table 1 reports the results of our proposed GAN architecture with the DDFN metric on the resolution 128×128, we have excluded this record in Table 4.

Dataset	Model	Res.	IS	FID	SNR_{db}
MCV	SA-GAN (baseline)	128	–	59.14 ± 2.24 ($\downarrow 4.54, \downarrow 1.46, -$)	41.91
		256	–	60.68 ± 2.17 ($\downarrow 2.39, \uparrow 0.73, \uparrow 1.03$)	39.11
	BigGAN	128	–	54.95 ± 1.04 ($\uparrow 2.14, \downarrow 3.48, -$)	46.93
		256	–	55.57 ± 2.78 ($\downarrow 1.71, \downarrow 2.15, \uparrow 1.56$)	46.29
	BigGAN (+DDFN)	128	35.57 ± 2.92 ($\uparrow 1.02, \uparrow 3.46, -$)	51.01 ± 2.18 ($\downarrow 3.52, \downarrow 2.11, -$)	53.72
		256	34.43 ± 1.19 ($-, \uparrow 2.04, \downarrow 0.97$)	53.46 ± 1.21 ($\uparrow 0.86, \downarrow 2.55, -$)	50.18
	Proposed (+DDFN)	128	50.79 ± 0.14 ($\downarrow 3.30, \downarrow 2.48, -$)	43.37 ± 1.65 ($\downarrow 6.88, \downarrow 4.99, -$)	56.91
		256	$\mathbf{56.20 \pm 1.73}$ ($\downarrow 4.03, \downarrow 2.95, -$)	$\mathbf{36.31 \pm 3.63}$ ($\downarrow 7.59, \downarrow 4.36, \uparrow 23.48$)	**62.89**
US8K	SA-GAN (baseline)	128	32.44 ± 1.76 ($\downarrow 0.43, \uparrow 1.35, -$)	–	33.95
		256	34.72 ± 0.23 ($\uparrow 1.17, \uparrow 2.39, -$)	–	34.12
	BigGAN	128	36.58 ± 1.88 ($\downarrow 1.37, \uparrow 0.77, \downarrow 1.83$)	39.20 ± 2.76 ($-, \downarrow 1.59, \downarrow 10.63$)	41.02
		256	39.16 ± 5.07 ($\downarrow 4.56, \downarrow 3.94, -$)	37.34 ± 3.53 ($\downarrow 5.21, \downarrow 2.75, \uparrow 16.18$)	48.14
	BigGAN (+DDFN)	128	40.99 ± 3.56 ($\downarrow 2.16, \uparrow 0.71, -$)	31.74 ± 1.17 ($\uparrow 1.95, \downarrow 2.33, \downarrow 8.77$)	49.39
		256	48.62 ± 0.23 ($-, \uparrow 2.39, -$)	25.63 ± 2.42 ($\uparrow 2.17, \uparrow 0.94, -$)	55.17
	Proposed (+DDFN)	256	$\mathbf{57.91 \pm 2.19}$ ($\downarrow 7.48, \downarrow 4.88, \downarrow 15.34$)	$\mathbf{16.33 \pm 1.87}$ ($\downarrow 3.42, \downarrow 4.88, -$)	**63.24**

Table 4: Extended comparison of average IS, FID and SNR for different models trained on logarithmic DWT spectrograms. Outperforming values are in bold. Spectral normalization applied only on $D(\mathbf{x}; \theta_d)$ for all the BigGANs and its modified variant (the proposed architecture). Res. stands for resolution.

C APPENDIX

Figure 4 depicts the schematic of our slightly modified version of BigGAN (Brock et al., 2019). This architecture uses the ResNet (He et al., 2016) with different channel multipliers and shared class embeddings in the generator. Unlike the BigGAN architecture, we constantly use 3×3 padded convolution with stride of 2. For the skip-z connection, we use static chunks of 20-D in accordance with each residual block. Details of the networks are shown in Table 5 and Table 6.

Table 5: Our slightly modified BigGAN architecture for the resolution 128×128 spectrograms. Channel stands for the width multiplier in both networks. The rest of the settings such as standing statistics for batch normalization (at the test time) are the same as (Brock et al., 2019).

Generator
$z \in \mathbb{R}^{120} \sim \mathcal{N}(0, I)$
Linear $(20 + 128)$ $(\rightarrow 4 \times 4 \times 16$ channel$)$
Residual Block $(16$ channel $\rightarrow 4$ channel$)$
Residual Block $(4$ channel $\rightarrow 1$ channel$)$
Non-local Block (128×128)
Batch Normalization, ReLU, Convolution
$\tanh$

Discriminator
RGB Spectrogram $\mathbf{x}_g \in \mathbb{R}^{128\times128\times3}$
Residual Block (channel $\rightarrow 2$ channel)
Non-local Block (64×64)
Residual Block (channel $2 \rightarrow 8$ channel)
Residual Block (channel $8 \rightarrow 16$ channel)
ReLU, Global Sum Pooling
Linear $\rightarrow 1$

Table 6: The modified BigGAN architecture for 256×256 spectrograms. This architecture has one additional residual network compared to smaller resolution 128.

Generator
$z \in \mathbb{R}^{140} \sim \mathcal{N}(0, I)$
Linear $(20 + 128)$ $(\rightarrow 4 \times 4 \times 16$ channel$)$
Residual Block $(16$ channel $\rightarrow 4$ channel$)$
Residual Block $(4$ channel $\rightarrow 4$ channel$)$
Residual Block $(4$ channel $\rightarrow 1$ channel$)$
Non-local Block (128×128)
Batch Normalization, ReLU, Convolution
$\tanh$

Discriminator
RGB Spectrogram $\mathbf{x}_g \in \mathbb{R}^{256\times256\times3}$
Residual Block (channel $\rightarrow 2$ channel)
Non-local Block (64×64)
Residual Block (2 channel $\rightarrow 4$ channel)
Residual Block $(4$ channel $\rightarrow 8$ channel$)$
Residual Block $(8$ channel $\rightarrow 16$ channel$)$
ReLU, Global Sum Pooling
Linear $\rightarrow 1$

As suggested (Brock et al., 2019) for the CC conditioning in each residual block, the linear projection has been used where the bias and gain projections are centered at 0 and 1, respectively. We use Orthogonal initialization (Saxe et al., 2014) for both generator and discriminator networks. For the choice of the optimier, Adam (Kingma & Ba, 2014) is utilized with $\beta_1 = 0.0$ and $\beta_2 = 0.9$. The learning rate is set to $2 \cdot 10^{-4}$ and $3 \cdot 10^{-5}$ for discriminator and generator at both resolutions. The rest of settings for baselines are the same as (Zhang et al., 2019) implemented in TensorFlow (Abadi et al., 2016).

C.1 STEPS FOR TRAINING

The first step in training a GAN with DDFN is computing DFN for real and generated samples. However, we have explained how to fastly approximate this metric in section 3.1. For real samples once we compute the DFN, but this operation is needed for all the generated samples. The second step is initializing the differentiable interval $[\varpi_{inf}, \varpi_{sup}]$ for (6). We empirically set $\varpi_{inf} = -1$ and $\varpi_{sup} = 1$. However, this is tunable according to the performance of the model. The third step is running (1) with the constraint (8).

D APPENDIX

We highly recommend reviewing these sources (Agbinya, 1996; Bahoura & Rouat, 2001) about DWT representations for audio and speech signals. Then we draw attentions to the clarification of three visualizations for DWT spectrograms. Briefly, the following code snippet explains linear, logarithmic, and logarithmic real visualizations.

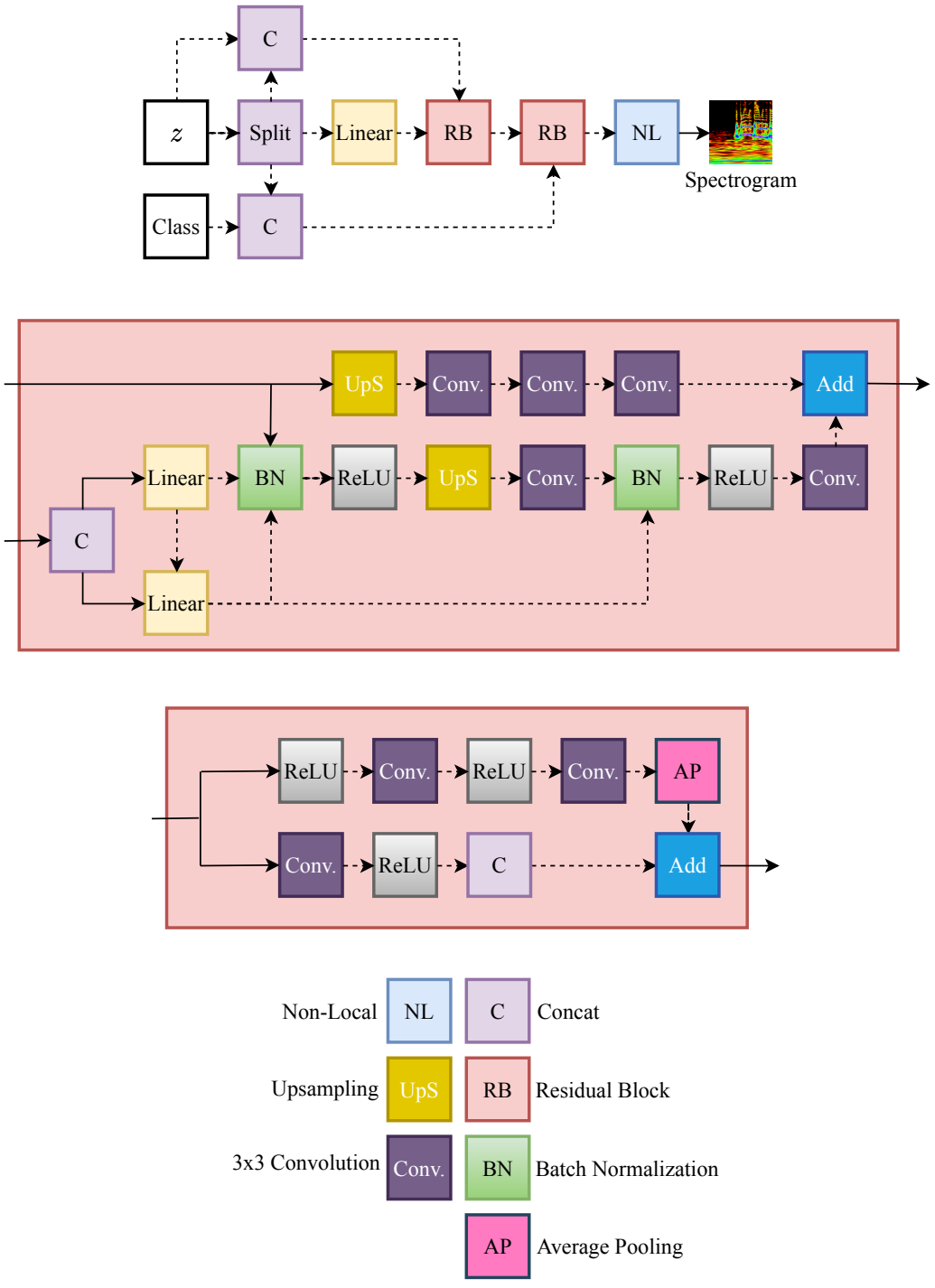


Figure 4: The proposed architecture, which is the slightly modified version of BigGAN (Brock et al., 2019). (Top): The generator architecture. (Middle): A residual block in generator. (Bottom): a residual block in discriminator.

```
static float linearDWTSpectrogram(float* data)
{
    /* Generating linear visualization for DWT spectrogram
    Real part: data[0], Imaginary part: data[1]
    */
    return sqrt(data[0]*data[0]+data[1]*data[1]);
}
static float logDWTSpectrogram(float* data)
{
    /* Generating logarithmic visualization
    for DWT spectrogram */
    return log(sqrt(data[0]*data[0]+data[1]*data[1]));
}
static float logRealDWTSpectrogram(float* data)
{
    /* Generating logarithmic-real visualization
    for DWT spectrogram */
    if (data[0] == 0.0) {
        return 0.0;
    }
    return log(abs(data[0]));
}
```

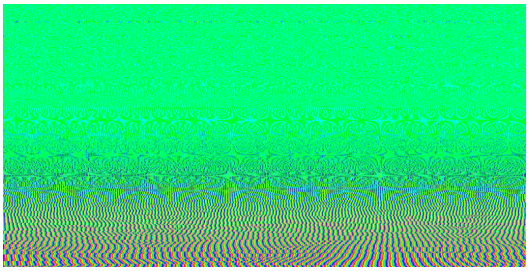
We have carefully trained (early stopped at checkpoints before overtraining or potential collapse) our proposed GAN architecture (the modified version of BigGAN with DFN) separately on the generated visualizations of DWT spectrograms. Afterwards, we illustrate some generated spectrograms. Additionally, we reconstruct each audio signals (one channel) from its spectrogram with a predefined original phase matrix to measure relative SNR (see Figures 5-10).

We encourage readers to listen to some reconstructed speech signals from the synthesized logarithmic DWT spectrograms using the proposed GAN architecture[3] which are included in folder "Synthesized-Speech" of the supplementary material. This folder includes two subfolders (set 1 and set 2) associated with two different resources. Each subfolder contains eight WAV signals (with their associated spectrograms under the same names as signals) which their details are listed herein:

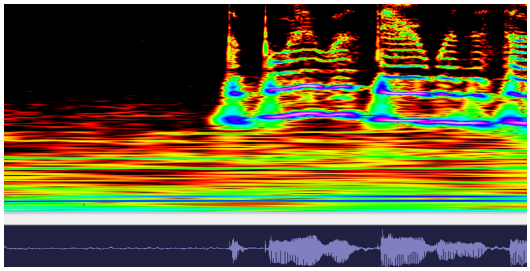
1. **Original speech:** This is a random original file from MCV dataset.
2. **Synthesized speech 1:** The reconstructed speech using the identical phase information from the aforementioned original speech. The synthesized spectrogram is generated by the model with $\epsilon = 34.53$ and $\alpha = 0.25$.
3. **Synthesized speech 2:** The reconstructed speech using the identical phase information from the aforementioned original speech. The synthesized spectrogram is generated by the model with $\epsilon = 47.05$ and $\alpha = 0.50$.
4. **Synthesized speech 3:** The reconstructed speech using the identical phase information from the aforementioned original speech. The synthesized spectrogram is generated by the model with $\epsilon = 56.37$ and $\alpha = 1$.
5. **Synthesized speech 4:** The reconstructed speech using the identical phase information from the aforementioned original speech. The synthesized spectrogram is generated by the model with $\epsilon = 78.14$ and $\alpha = 2$.
6. **Approximated approach 1:** The reconstructed speech using the approximated phase information as proposed in (Krawczyk & Gerkmann, 2014). The synthesized spectrogram is generated by the model with $\epsilon = 34.53$ and $\alpha = 0.25$.
7. **Approximated approach 2:** The reconstructed speech using the Griffin-Lim approach (Masuyama et al., 2019).
8. **Approximated approach 3:** The reconstructed speech using MelGAN approach (Kumar et al., 2019).

[3] $Res. = 256, batch = 512, ch. = 96$ with spectral normalization and the DDFN on discriminator and generator, respectively.

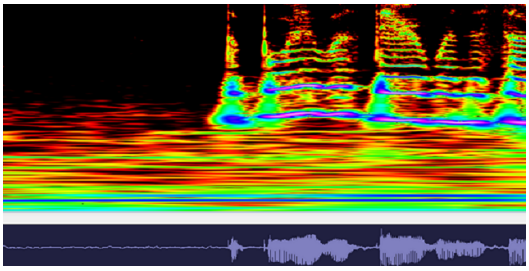
There is (relatively) no sensible background or ambient noise in the reconstructed speech signals with synthesized spectrograms and original phase information (synthesized speech 1-4). Although, reconstructing signals with approximated phase vectors (i.e., approximated approach 1-3) add noticeable noises to the signals, still the speech is understandable. This denotes the critical role of high quality spectrogram in reconstruction.



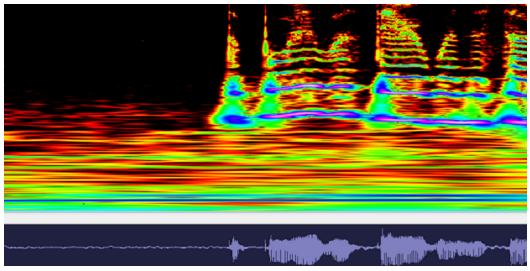
(a) Original phase



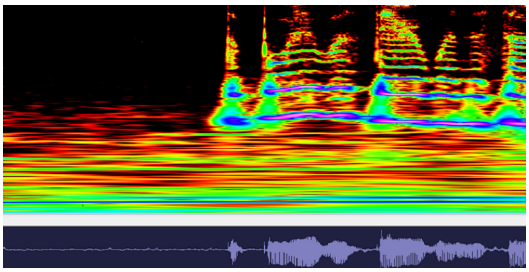
(b) Original Spectrogram



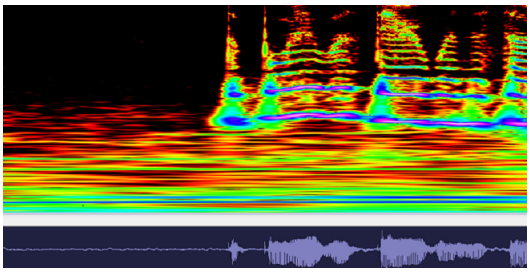
(c) Synthesized spectrogram



(d) Synthesized spectrogram

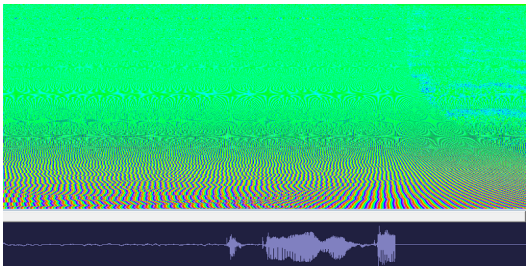


(e) Synthesized spectrogram

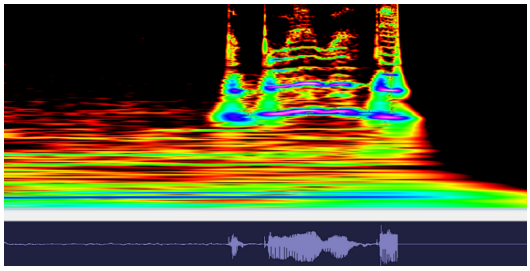


(f) Synthesized spectrogram

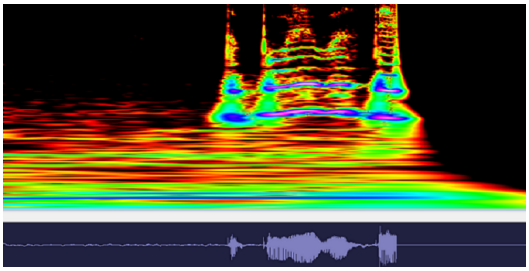
Figure 5: Generated logarithmic DWT spectrograms with our modified BigGAN and the DDFN for a random signal sig_1. Synthesized signals are reconstructed with the the original phase.



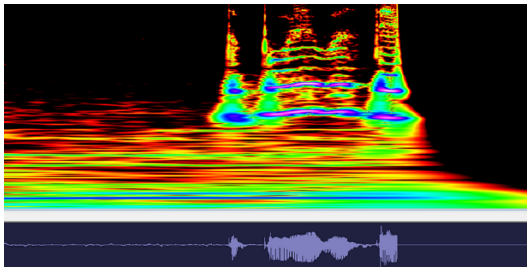
(a) Original phase



(b) Original spectrogram

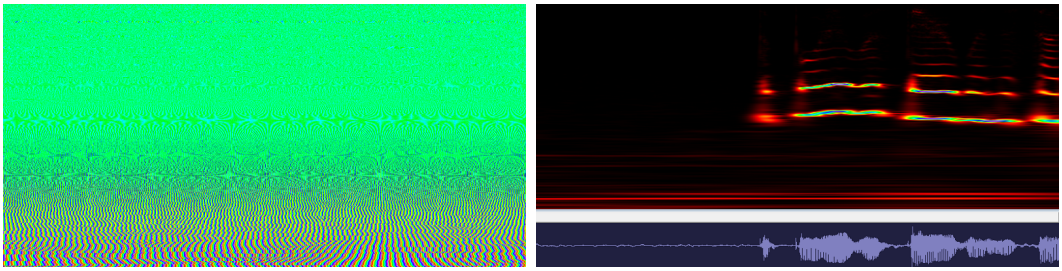


(c) Synthesized spectrogram

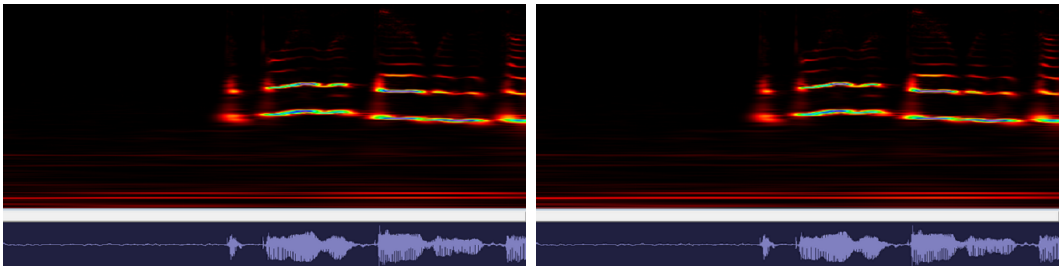


(d) Synthesized spectrogram

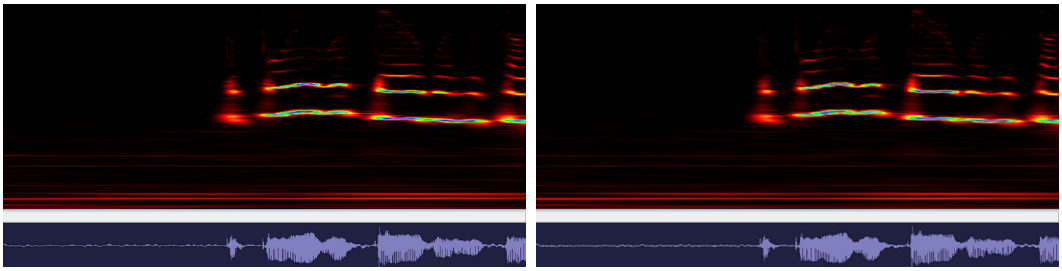
Figure 6: Generated logarithmic DWT spectrograms with our modified BigGAN and the DDFN for another random signal sig_2. Synthesized signals are reconstructed with the the original phase.



(a) Original Phase (b) Original spectrogram

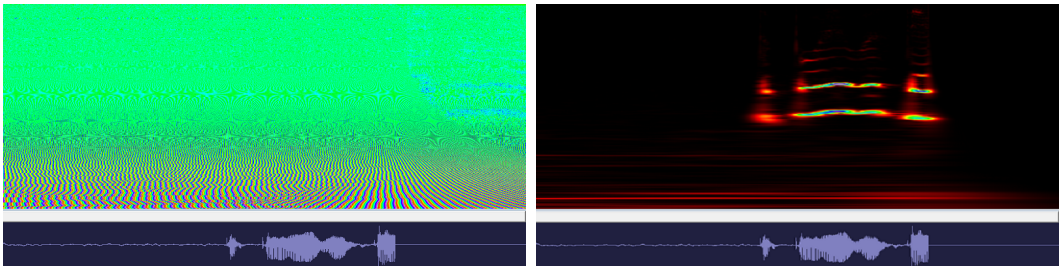


(c) Synthesized spectrogram (d) Synthesized spectrogram

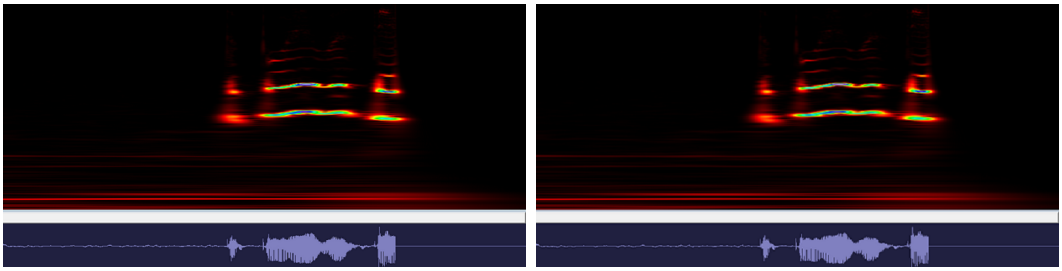


(e) Synthesized spectrogram (f) Synthesized spectrogram

Figure 7: Generated linear DWT spectrograms with our modified BigGAN and the DDFN for a random signal sig_1. Synthesized signals are reconstructed with the the original phase.

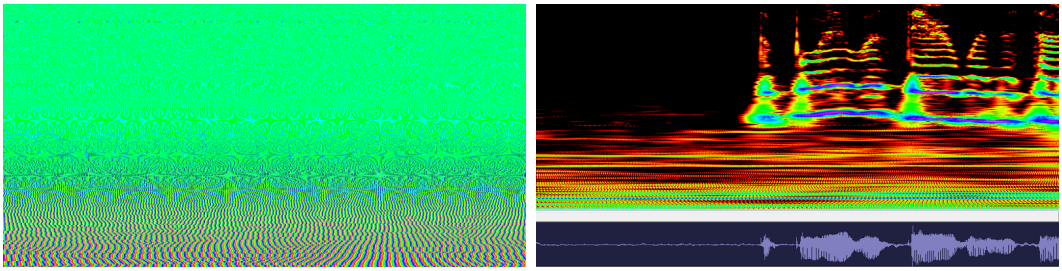


(a) Original phase (b) Original spectrogram

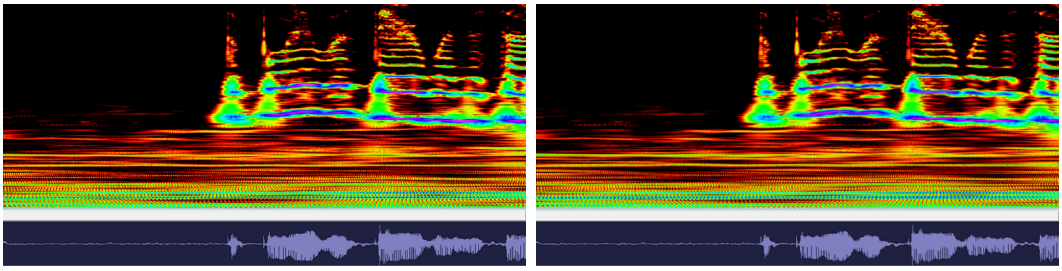


(c) Synthesized spectrogram (d) Synthesized spectrogram

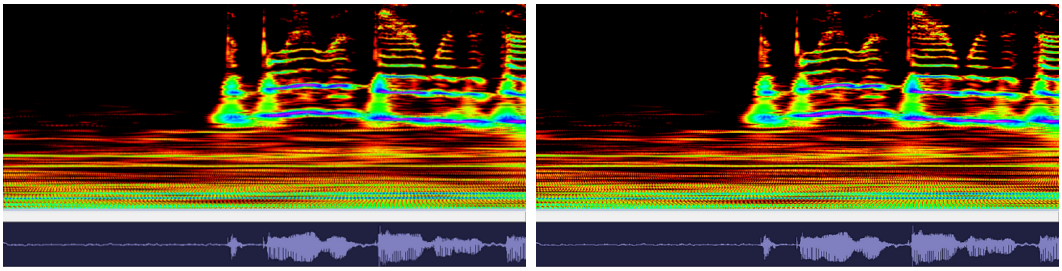
Figure 8: Generated linear DWT spectrograms with our modified BigGAN and the DDFN for another random signal sig_2. Synthesized signals are reconstructed with the the original phase.



(a) Original phase (b) Original spectrogram

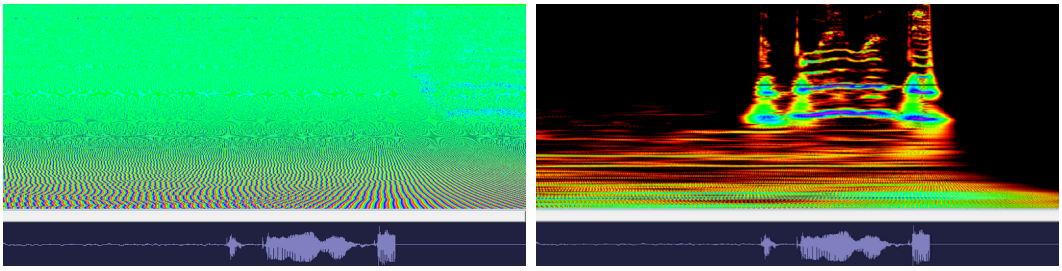


(c) Synthesized spectrogram (d) Synthesized spectrogram

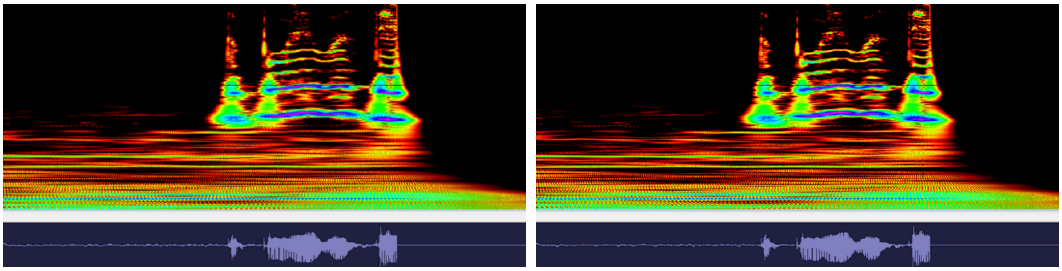


(e) Synthesized spectrogram (f) Synthesized spectrogram

Figure 9: Generated logarithmic-real DWT spectrograms with our modified BigGAN and the DDFN for a random signal sig_1. Synthesized signals are reconstructed with the the original phase.



(a) Original phase (b) Original spectrogram



(c) Synthesized spectrogram (d) Synthesized spectrogram

Figure 10: Generated logarithmic-real DWT spectrograms with our modified BigGAN and the DDFN for another random signal sig_2. Synthesized signals are reconstructed with the the original phase.