# DCASE TASK-7: STYLEGAN2-BASED FOLEY SOUND SYNTHESIS

Technical Report

*Purnima Kamath*[1], *Tasnim Nishat Islam*[2],*Chitralekha Gupta*[1], *Lonce Wyse*[3], *Suranga Nanayakkara*[1],

[1] National University of Singapore, Singapore, purnima.kamath@u.nus.edu,
{chitralekha, scn}@nus.edu.sg
[2] Bangladesh University of Engineering and Technology, Bangladesh, 1706092@eee.buet.ac.bd
[3] Universitat Pompeu Fabra, Barcelona, Spain, lonce.acad@zwhome.org

## ABSTRACT

For the DCASE Challenge 2023 Task 7 (Track B), Foley Sound Synthesis, we submit two systems, (1) a StyleGAN conditioned on the class ID, and (2) an ensemble of StyleGANs each trained unconditionally on each class separately. We quantitatively find that both systems out-perform the task 7 baseline models in terms of FAD Scores. Given the high inter-class and intra-class variance in the development datasets, the system conditioned on class ID is able to generate a smooth and a homogeneous latent space indicated by the subjective quality of its generated samples. The unconditionally trained ensemble generates more categorically recognizable samples than system 1, but tends to generate more instances of out-of-distribution or noisy samples.

***Index Terms—*** stylegan2, pghi, gabor transform

## 1. INTRODUCTION

Generative audio algorithms using deep neural networks aim to generate novel audio that matches naturally occurring sounds in their qualities such as realism or plausibility of the sound. Recently, there has been a focus on developing such models for inharmonic sounds such as those of environmental audio. Such synthesis models are useful for generating background environmental sound scores for movies, games, and automated Foley sound synthesis. The task in DCASE Foley Sound Synthesis challenge [1, 2] this year is to generate sounds of seven sound classes with high fidelity and diversity. There are two tracks - tracks A and B - in this challenge, each using a curated dataset and with or without external resources outlined on the challenge webpage[1]. Our submission is for **track B**, i.e., using only the development dataset and without the use of any external resources (audio data or pre-trained models).

For our submission, we use a type of Generative Adversarial Network (GAN) [3] called StyleGAN2 [4, 5] trained from scratch on log-magnitude spectrogram representations of the environmental sounds in the dataset. Generally, GANs learn a distribution of the sounds in the dataset, such that random sampling within the learned latent space generates novel audio samples matching the fidelity of the real-world training data. StyleGANs are designed to further improve the quality of the generated sounds by better disentangling the factors of variations observed in the dataset using an intermediate latent space. Such architectures are inspired by the style transfer tasks and learn the intermediate latent space using a set of affine transforms called the mapping network.

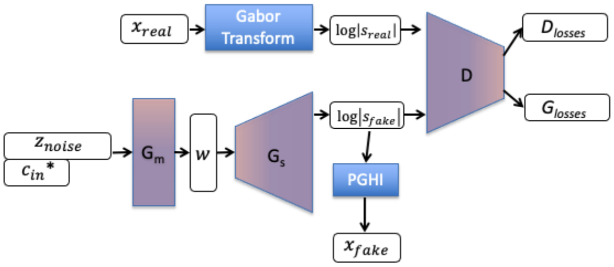

Figure 1: A schematic outlining the main components in our submission for both System 1 and System 2. Conditioning vector $c_{in}^*$ is applied only to System 1.

We submit two systems for this challenge - (1) System 1: A conditional StyleGAN2 trained on the entire development dataset and conditioned on the class-IDs using one-hot encoding, and (2) System 2: An ensemble of unconditionally trained StyleGAN2 networks, one for each class of sounds in the dataset. We empirically decide the values for certain hyperparameters of the StyleGAN2 architecture (e.g., the dimensionality of the latent space and the number of layers in the network) depending on the number of classes being modeled in the system. We report the Fréchet Audio Distance [6] scores for each class per system on the training set. We describe each system in detail in the following sections.

## 2. SYSTEM OVERVIEW

Figure 1 illustrates the main components within our submission. We use StyleGAN2 in conjunction with log-magnitude spectrogram representations generated using Gabor transforms [7]. Previously, Gupta et al [8] showed that using the phase gradient heap integration method (PGHI) [7, 9] for phase reconstruction during spectrogram inversion is an effective way to reconstruct sharp and clear transients in the resulting sounds. As most of the environmental sounds in the development dataset in this challenge include sound events with sharp attacks and transients (such as Dog Barks or Footsteps), we use the Gaussian windowed log-magnitude spectrogram representations during training and PGHI for high-fidelity spectrogram inversion in conjunction with StyleGAN2.

We use StyleGAN2 from Nvidia's official codebase[2] and adapt

[1] https://dcase.community/challenge2023/task-foley-sound-synthesis

[2] https://github.com/NVlabs/stylegan2-ada-pytorch

Table 1: System Details

| | **Class** | $w/z$**-dim** | **No. Mapping Layers** | **Training Iters (kimgs)** | **Training time (∼days)** |
|---|---|---|---|---|---|
| System 1<br>(Conditional. One model, all classes.) | All Classes | 512 | 8 | 1200 | 2.125 |
| System 2<br>(Unconditional. Individual models for each class.) | Dog Bark | 128 | 4 | 1600 | 2.5 |
| | Footstep | 128 | 4 | 2600 | 4.7 |
| | Gunshot | 128 | 4 | 3200 | 5 |
| | Keyboard | 128 | 4 | 2200 | 3.95 |
| | Moving Motor Vehicle | 128 | 4 | 800 | 1.29 |
| | Rain | 128 | 4 | 1600 | 2.4 |
| | Sneeze/Cough | 128 | 4 | 1800 | 3.79 |

it to train using audio spectrograms. In this report, we elaborate mostly on the Generator of StyleGAN2 as most of our changes to the official repository focus on that component. As shown in Figure 1, we use the Gaussian windowed log-magnitude Short-time Fourier Transform (STFT) of an audio sample $x_{real}$ to train the GAN. The aim of the generator is to synthesize an audio sample $x_{fake}$ which resembles $x_{real}$. The generator samples from the $Z$ latent space to synthesize $x_{fake}$. Specifically, a StyleGAN2's generator can be modeled as a two functions - a mapping network or a set of fully connected layers $G_m(.)$ that maps a d-dimensional latent space $z_{noise} \in R^{d_z}$ to an intermediate $w \in R^{d_w}$ space and a synthesis network $G_s(.)$ that maps the resulting $w$ vector to the spectrogram space $s \in R^{f \times t}$. Here $d_z$, $d_w$ is the dimensionality of the $Z$ and $\mathcal{W}$ space respectively. And $f$, $t$ are the number of frequency channels and time frames of the generated spectrogram.

## 3. EXPERIMENTAL SETUP

For System 1 (conditional), we set both $d_z$ and $d_w$ to 512. The number of fully connected layers in the mapping network $G_m$ (or the number of affine transforms before $w$ vectors are generated) is set to 8. For System 2 (unconditional ensemble), we set $d_z$ and $d_w$ both to 128 and number of layers in the mapping network to 4. Further, we use a batch size of 4 or 8 (depending upon the resource availability on our shared compute infrastructure during training) to train the networks.

All our models were trained either on a single RTX 3090 24GB GPU or the National University of Singapore's high-performance compute infrastructure (shared single Nvidia Tesla V100 32 GB GPU). The training details with respect to the number of epochs or iterations and the time taken are outlined in table 1.

### 3.1. Dataset

For this task, we used only the development dataset outlined in the challenge description [1, 2]. The dataset consists of environmental sounds from 7 classes. Classes such as Dog Bark, Footstep, and Gunshots contained multi-event sounds with sharp transients, whereas classes such as Rain or Motor Vehicle contained more noisy sounds. Each sound sample was 4 seconds long and sampled at 22,050 Hz. We generate the Gaussian windowed log-magnitude spectrogram with stft_channels $=$ 2048, n_frames $=$ 1024 and hop_size $=$ 128.

### 3.2. Data Augmentation

GANs are powerful generative architectures but need large datasets to model the distributions effectively. The number of samples per class in the development dataset was very small, with an average of ∼46 minutes per class. We thus augmented our development dataset using one of two simple strategies - zero-pad, and wrap-around, before training our unconditional System 2. Note that no data augmentation was done for the conditionally trained System 1.

For all audio samples in training that contained events lasting less than 2.5 seconds (detected by simply thresholding), we applied the zero-pad augmentation strategy. On closer observation of the nature of the audio samples under each class, multiple samples had sound events lasting only a few seconds with zero-padding for the remainder of the sample (e.g., some Dog Barks and Gunshot samples). To augment such samples, we shifted the sound events along the right of the time axis, while padding the beginning of the sample with zeros. For sounds that had events lasting more than 2.5 seconds (e.g., Moving Motor Vehicle), we used the wrap-around strategy where we simply wrapped around and shifted the samples along the time axis after removing the padded silences during augmentation. We applied these augmentations to each audio file 10 times, which augmented our training data by a factor of 10 for each class.

### 3.3. Evaluation Methodology

We use the Fréchet Audio Distance(FAD) [6] to evaluate the quality of our synthesized audio for both systems. This metric measures the distance between the distributions of training data and the synthesized audio based on their VGGish embeddings. We synthesized 100 samples for each class and computed the FAD score against the entire training set for that class. Further, this score was computed for multiple checkpoints during training. We selected 2-3 checkpoints based on best FAD scores and then subjectively evaluated by listening (internally within the research team) to the synthesized audio for artefacts such as smearing of the attack transients in the samples and recognizability of the sounds. We eventually selected the model which generated more recognizable sounds than others and qualitatively preserved the transients for submission for this task irrespective of their FAD scores.

## 4. RESULTS & DISCUSSION

Table 2 shows the FAD scores for both System 1 and 2. Standard error of means computed by bootstrapping 10 times. Scores marked with $^*$ are higher than the baseline in the task. Mean FAD scores for System 1 and 2 were 6.50 and 4.02 respectively.

Table 2: FAD Scores

| | **Class** | **FAD Scores(↓)** |
|---|---|---|
| System 1<br>(Conditional) | Dog Bark | 5.34 ± 0.76 |
| | Footstep | 5.06 ± 0.34 |
| | Gunshot | $9.98 \pm 0.66^*$ |
| | Keyboard | 3.94 ± 0.26 |
| | Moving Motor Vehicle | 14.26 ± 0.82 |
| | Rain | 5.30 ± 0.52 |
| | Sneeze/Cough | 1.65 ± 0.08 |
| System 2<br>(UnConditional or per-class) | Dog Bark | 3.80 ± 1.09 |
| | Footstep | 3.30 ± 0.21 |
| | Gunshot | 4.40 ± 0.36 |
| | Keyboard | 3.38 ± 0.18 |
| | Moving Motor Vehicle | 7.05 ± 1.27 |
| | Rain | 4.21 ± 0.38 |
| | Sneeze/Cough | 2.02 ± 0.11 |

While System 2 (unconditional ensemble) organizes its latent space according to the variances in each individual class (intra-class variance), System 1 (conditional) has an additional task of organizing its latent space according to both inter-class as well as intra-class variances in the dataset. The implications from this on the quality of generated sounds is two-fold - (1) though System 2 shows lower FAD scores than System 1, the latent space generated by System 2 has 'holes' [10] in the latent space which generate out-of-distribution (OOD) or noisy sounds. This nature of the latent space can be attributed to the high intra-class variance in the sound samples in the training set. (2) Although System 1 does not generate many OOD sounds and has a homogenous or smooth latent space as compared to System 2, it generates more sounds which can be subjectively mis-categorized (i.e., the 'holes' in the latent space are filled with sounds from another class or category). For instance, some System 1 synthesized Gun Shot sounds, such as machine gun sounds, sound like Keyboard clicks. In this regard, System 2 generates more categorically recognizable sounds.

Further, while training System 1 (conditional), we observe that all classes do not train equally through the training iterations. While training for longer epochs, some classes, such as Dog Barks, tend to overfit while other classes such as Gun Shots are still generalizing to the distribution.

## 5. LIMITATIONS

The StyleGAN2 architecture was originally developed to a learn latent distributions for images. As such, this architecture trains using square (same height and width)images. To adapt this architecture to audio, we design square spectrograms by zero padding the raw audio and selecting a specific number of frequency channels and time bins. Our future work will involve modifying this architecture to use spectrograms of any number of frames and frequency channels.

## 6. REFERENCES

[1] K. Choi, S. Oh, M. Kang, and B. McFee, "A proposal for foley sound synthesis challenge," 2022.

[2] K. Choi, J. Im, L. Heller, B. McFee, K. Imoto, Y. Okamoto, M. Lagrange, and S. Takamichi, "Foley sound synthesis at the dcase 2023 challenge," *In arXiv e-prints: 2304.12521*, 2023.

[3] I. J. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," *CoRR*, vol. abs/1701.00160, 2017. [Online]. Available: http://arxiv.org/abs/1701.00160

[4] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[5] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

[6] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms." in *INTERSPEECH*, 2019, pp. 2350–2354.

[7] A. Marafioti, N. Perraudin, N. Holighaus, and P. Majdak, "Adversarial generation of time-frequency features with application in audio synthesis," in *International conference on machine learning*. PMLR, 2019, pp. 4352–4362.

[8] C. Gupta, P. Kamath, and L. Wyse, "Signal representations for synthesizing audio textures with generative adversarial networks," *arXiv preprint arXiv:2103.07390*, 2021.

[9] Z. Prusa, P. Balazs, and P. L. Sondergaard, "A noniterative method for reconstruction of phase from stft magnitude," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1154–1164, 2017.

[10] A. Pati and A. Lerch, "Is disentanglement enough? on latent representations for controllable music generation," *CoRR*, vol. abs/2108.01450, 2021. [Online]. Available: https://arxiv.org/abs/2108.01450