HYU SUBMISSION FOR THE DCASE 2023 TASK 7: DIFFUSION PROBABILISTIC MODEL WITH ADVERSARIAL TRAINING FOR FOLEY SOUND SYNTHESIS

Technical Report

Won-Gook Choi

Department of Electronic Engineering
Hanyang University, Seoul, Republic of Korea
onlyworld94@hanyang.ac.kr

*Joon-Hyuk Chang**

Department of Electronic Engineering
Hanyang University, Seoul, Republic of Korea
jchang@hanyang.ac.kr

ABSTRACT

This paper is a technical report of the Hanyang University team submission for the DCASE 2023 challenge task 7, Foley Sound Synthesis. The goal of the task is to build a generative model that can synthesize high-quality and various foley sounds: the sounds of dog barking, footsteps, gunshots, keyboards, moving motor vehicles, rainy scenes, and sneezing. The core strategy of the submissions is a diffusion probabilistic model-based acoustic model. Also, we adopted adversarial training on the evidence lower bound (ELBO) of the diffusion model for the higher quality. The submissions did not use any external dataset and achieved lower Frechet audio distance (FAD) scores than the DCASE baseline, except for the sounds of moving motor vehicles.

***Index Terms*—** Diffusion probabilistic model, adversarial training, sound synthesis

1. INTRODUCTION

Foley sound synthesis (FSS) is the task of generating realistic audio effects from the given class indicators. The foley sounds comprise some environmental sounds and event sounds such as rainy acoustic scenes, sounds of gunshots, etc.

This report describes the technical approaches for the detection and classification of acoustic scenes and events (DCASE) challenge task 7 track B[1]. The goal of the task is to generate high-quality and various sounds for the seven predefined sounds only using the given dataset. The dataset contain sounds of dog barks, footsteps, gunshots, keyboards, moving motor vehicles, rainy scenes, and sneezing.

One of the well-known issues of the generative task is the trade-off between sample quality and diversity. Our solution to the problem is designing the FSS model with a diffusion probabilistic model [1] allowing the synthesis of both high-quality and diverse sounds. Also, we adopted adversarial training to the evidence lower boundary (ELBO) of the diffusion model to improve the quality of samples [2, 3]. In the following sections, we will report the data description, audio processing methods, strategies for the problem, and results in detail.

*corresponding author.

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

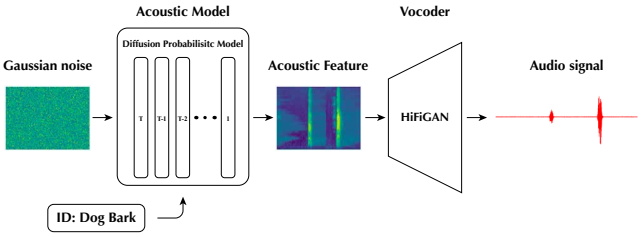

Figure 1: Overall process of the proposed sound synthesis system.

2. AUDIO DATA

The development dataset[2] consists of 4,850 audio clips with 4 s lengths. Each audio clip was sampled from one of three datasets: UrbanSound8K [4], FSD50K [5], and BBC Sound Effects, and converted into a mono channel, 16-bit, and sampling rates of 22,050 Hz. Some sounds were zero-padded if the duration was shorter than 4 s. The dataset is divided into seven classes: sounds of dog barks, footsteps, gunshots, keyboards, moving motor vehicles, rainy scenes, and sneezing (or cough), with more than 500 and less than 800 clips for each class. Within the class, the sounds consist of various patterns and characteristics. For example, the sounds of gunshots are composed of the sounds of different machine guns, pistols, etc.

3. PROPOSED SOULTIONS

3.1. Overall process

Our systems consist of two main parts: an acoustic model and a vocoder (Fig. 1). The acoustic model extracts an acoustic feature corresponding to a given target ID; subsequently, the vocoder decodes the acoustic feature into the audible signal by phase reconstruction. In this work, we design the acoustic model using diffusion probabilistic model [1] and use pre-trained HiFiGAN [7] as the vocoder. Also, the target IDs are implemented with one hot vectors.

3.2. Acoustic model

In many studies, diffusion probabilistic models have been shown to generate high-quality and diverse samples [8, 9]. The main idea

[2] Dataset is available in https://drive.google.com/drive/folders/1GzfZvYVdbgDXnykOR93C3LCchPYBPh5I

of the diffusion model is to estimate the diffusion noise from the diffused data. The diffusion process $q(X_t|X_{t-1})$ follows:

$$\begin{aligned} q(X_t|X_{t-1}) &\triangleq \mathcal{N}(\sqrt{1-\beta_t}X_{t-1}, \beta_t I) \\ X_t &= \sqrt{1-\beta_t}X_{t-1} + \sqrt{\beta_t}\epsilon_t \\ &= \sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, \end{aligned} \tag{1}$$

where $X_0, \{X_t\}_{t=1}^T, \{\beta_t\}_{t=1}^T$, and T denote a log-mel spectrogram of an original data, set of diffused data on t step, variance schedule, and the total diffusion timesteps. The diffusion noise ϵ_t is sampled from the normal distribution $\mathcal{N}(0, I)$, and the distribution of the X_t converges to $\mathcal{N}(0, I)$ when t becomes closer to the T. If the diffusion process is assumed as a Markov chain, X_t can be sampled from X_0 with the definition of $\bar{\alpha}_t := \prod_{t=1}^T (1-\beta_t)$. The diffusion probabilistic model is defined as $p_\theta(X_{t-1}|X_t, y)$, where the output of denoising network $\epsilon_\theta(X_t, t, y)$ estimates the observed diffusion noise ϵ_t. We optimized the denoising network using the following objective function:

$$L_{\text{DDPM}} = \mathbb{E}_{X_0,\epsilon_t,t}[||\epsilon_\theta(X_t, t, y) - \epsilon_t||_2^2], \tag{2}$$

which is simplified version of the ELBO of the log-likelihood. In this work, we use the same variance schedule and the denoising network (UNet) as used by Ho et al [1].

The token IDs are expressed as one-hot vectors and the vectors indicate embedding vectors stored in the lookup tables implemented by the PyTorch *Embedding* module. Subsequently, each embedding is projected by two linear layers with a Gaussian error linear units activation, and summed up with time embeddings. We also applied classifier-free guidance [10] with null ID, and the ratio of null ID was 0.2.

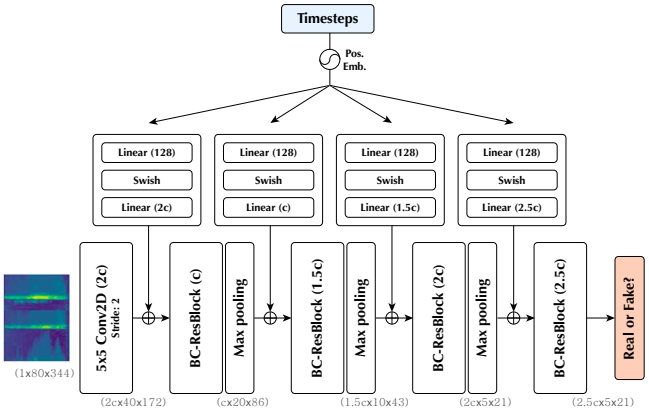

Figure 2: Detailed architecture of the discriminator. In this work, $c = 40$

3.3. Discriminator

Wang et al. [3] showed that the adversarial training to the diffusion model can make sample quality higher. In this work, we added a min-max objective following:

$$\begin{aligned} L_{\text{D}} = \mathbb{E}_{q(X_{t-1}|X_t)p_\theta(\hat{X}_{t-1}|X_t)}[&(D_\phi(X_{t-1}, t) - 1)^2 \\ &+ (D_\phi(\hat{X}_{t-1}, t))^2] \end{aligned} \tag{3}$$

Algorithm 1 Pseudo code of sampling process of the acoustic model.

 # prepare the optimized denoising netowork, net()
choose y **for** y **in** $[0, 1, 2, 3, 4, 5, 6]$
$X_T \sim \mathcal{N}(0, I)$
 # the shape of X_T is same to the log-mel spectrogram of the 4 sec. length audio
 # the elements of X_T are i.i.d. samples
$X_t = X_T$
for t **in** $(T, ..., 1)$**:**
  $z \sim \mathcal{N}(0, I)$
  if γ == 1:
    ϵ_θ = net(X_t, t, y)
  else:
    ϵ_θ = (1-γ) net$(X_t, t, \emptyset)$ + γ net(X_t, t, y)
  $X_{t-1} = \frac{1}{\sqrt{\alpha_t}}(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(X_t, t, y)) + \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}z$
  $X_t = X_{t-1}$
return X_t

$$L_{\text{Adv}} = \mathbb{E}_{p_\theta(\hat{X}_{t-1}|X_t)}[(D_\phi(\hat{X}_{t-1}, t) - 1)^2], \tag{4}$$

where D_ϕ, and $\hat{X}_{t-1}$ denote the discriminator network, and the estimated diffused posterior on the step t by p_θ. Finally, the diffusion model is optimized by minimizing the the total loss:

$$L_{\text{G}} = L_{\text{DDPM}} + L_{\text{Adv}}. \tag{5}$$

In this work, we used a little modified BC-ResNet-mod-4 [11] (Fig. 2) as the descriminator. The discriminator does not aggregate the feature map into scalar for determination whether the input data is real or fake; in other words, the output of the discriminator is the feature map of the input data, and Equation 3 is calculated pixel by pixel.

3.4. Sampling

Sampling process follows the probability distribution of reverse process $p_\theta(X_{t-1}|X_t, y)$.

$$\begin{aligned} p_\theta(X_{t-1}|X_t, y) &\triangleq \mathcal{N}(\tilde{\mu}_\theta(X_t, t, y), \tilde{\beta}_t I) \\ \tilde{\mu}_\theta(X_t, t, y) &= \frac{1}{\sqrt{\alpha_t}}(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(X_t, t, y)) \\ \tilde{\beta}_t &= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \end{aligned} \tag{6}$$

If the classifier-free guidance is applied, $\epsilon_\theta(X_t, t, y)$ is replaced to $(1-\gamma)\epsilon_\theta(X_t, t, y = \emptyset) + \gamma\epsilon_\theta(X_t, t, y)$, where γ and $\emptyset$ denote a guidance scale and the null ID. The details of sampling procedure is shown in **Algorithm 1**.

3.5. Vocoder

After sampling the acoustic feature X_0, we decode the feature into the audible signal using vocoder. The specification of the decoded signal is 22,050 Hz, and mono channel. We used the pre-trained HiFiGAN given from the task 7[3].

[3] https://github.com/DCASE2023-Task7-Foley-Sound-Synthesis/dcase2023_task7_baseline

Table 1: Comparison of the FAD scores to the baseline and the submissions.

System		Dog Bark	Footstep	Gunshot	Keyboard	Vehicle	Rain	Sneeze
FAD score	DCASE Baseline[4] [6]	13.411	8.109	7.951	5.230	**16.108**	13.337	3.770
	Submission 1	5.056	5.753	**5.886**	**4.508**	20.729	**6.399**	1.706
	Submission 2	**4.518**	**5.745**	6.992	4.696	18.623	6.912	**1.600**

4. EXPERIMENTS

4.1. Audio processing

We used log mel spectrogram for the audio feature. The sampling rates, window size, hop size, and the number of mel bands were 22,050 kHz, 1024, 256, and 80. The number of FFT points were same to the window size. Before STFT, we eliminated the zero paddings, and after extracting the log-mel spectrogram, we randomly cropped the spectrogram into 140 frames for the efficient training. If the number of frames were less than 140, spectrograms were zero-padded.

4.2. Training details

We used UNet[5] of 32 initial dimensions for the denoising network, in which the feature map resolutions were reduced to $80 \times F$, $40 \times F/2$, $20 \times F/4$ and $10 \times F/8$ for each down-block. The diffusion process followed the linear variance schedule of from 2.5×10^{-4} to 0.05 with the total timestep $T = 400$. Also, the dimensions of the temporal positional embeddings, token embeddings, and linear projections for those embeddings were 128.

The BC-ResNet-Mod-4 [11] serves as the discriminator (Fig. 2). In details, pooling layers were inserted after all BC-ResBlocks [12] except the last block, and the sinusoidal time embeddings [13] were summed up with the hidden features before the operation of the BC-ResBlocks. Also, we eliminated the classifier layer; in other words, the outputs of the last Res-Block is the final outputs of the discriminator.

We optimized the networks using AdamW [14] optimizers with $\beta_1 = 0.8, \beta_2 = 0.99$, learning rates of 0.0001, and weight decays of 0.0001. For the discriminator, we used a learning rate scheduler that decayed the learning rate with a 0.999975 factor in every training step. Also we used mixed precision training. *Submission 1* and *Submission 2* were trained 600 k and 650 k steps, respectively, with 64 batch size.

5. RESULTS

To evaluate our systems, we randomly sampled 100 sounds for each class, and measured Frechet audio distance (FAD) score[6] [15]. All sounds were sampled in 22,050 Hz, mono channel and 4 sec length. Sounds of dog bark, gunshot, and sneeze were sampled with a guidance scale of 3, and the others were sampled without guidance. As shown in Table 1, our submissions showed the lower FAD scores than the DCASE baseline except the moving vehicle sounds. When the guidances were applied, the FAD scores of dog bark, gunshot, and sneeze samples were enhanced but the others were lower. In the case of the moving motor vehicle sounds, the FAD scores were higher than the baseline. In our analysis, this is because the sounds of moving motor vehicles have noisy pattern, and the denoising process could not recognize the diffusion noise to the noise for the reconstruction. In other words, the network might confuse the noisy pattern of the sound to the diffusion noise.

[5] https://github.com/lucidrains/denoising-diffusion-pytorch

[6] https://github.com/DCASE2023-Task7-Foley-Sound-Synthesis/dcase2023_task7_eval_fad

6. REFERENCES

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," vol. 33, 2020, pp. 6840–6851.

[2] S. Liu, D. Su, and D. Yu, "Diffgan-tts: High-fidelity and efficient text-to-speech with denoising diffusion gans," *arXiv preprint arXiv:2201.11972*, 2022.

[3] Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou, "Diffusion-gan: Training gans with diffusion," *arXiv preprint arXiv:2206.02262*, 2022.

[4] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. the 22nd ACM international conference on Multimedia*, 2014, pp. 1041–1044.

[5] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2021.

[6] X. Liu, T. Iqbal, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, "Conditional sound generation using neural discrete time-frequency representation learning," 2021, pp. 1–6.

[7] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," vol. 33, 2020, pp. 17 022–17 033.

[8] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-tts: A diffusion probabilistic model for text-to-speech," in *Proc. International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 8599–8608.

[9] M. Jeong, H. Kim, S. J. Cheon, B. J. Choi, and N. S. Kim, "Diff-tts: A denoising diffusion model for text-to-speech," 2021.

[10] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *Proc. Advances in neural information processing systems (NeurIPS) Workshop on Deep Generative Models and Downstream Applications*, 2021.

[11] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE: Residual normalization for device-imbalanced acoustic scene classification with efficient design," DCASE2021 Challenge, Tech. Rep., June 2021.

[12] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," 2021.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," vol. 30, 2017.

[14] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. International Conference on Learning Representations (ICLR)*, 2019.

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