CLASS-CONDITIONED LATENT DIFFUSION MODEL FOR DCASE 2023 FOLEY SOUND SYNTHESIS CHALLENGE

Technical Report

Robin Scheibler, Takuya Hasumi, Yusuke Fujita, Tatsuya Komatsu, Ryuichi Yamamoto, Kentaro Tachibana

LINE Corporation, Tokyo, Japan

ABSTRACT

This report describes our submission to DCASE2023 Task7: Foley sound synthesis challenge. Our system uses a latent diffusion model (LDM) that generates a latent representation of audio conditioned on a specified sound class, a variational autoencoder that converts the latent representation to a mel-spectrogram, and a neural vocoder based on HiFi-GAN that reconstructs a natural waveform from the mel-spectrogram. We train the LDM using the DCASE2023 Task7 development set with its sound class indices as conditioners for generating class-specific latent representations. To enhance the diversity of generated sounds, we finetune a pretrained text-to-audio LDM that is trained with the AudioCaps dataset and an instruction-tuned large language model. Furthermore, we utilize a postprocessing filter that selects a subset of generated sounds to match a distribution of target class sounds. Our system achieved an average Fréchet audio distance of 4.744, which is significantly better than 9.702 produced by the baseline system.

Index Terms— foley sound synthesis, conditional sound generation, latent diffusion, Fréchet audio distance

1. INTRODUCTION

Foley sound synthesis is the task of generating sound effects added to multimedia content to enhance the perceptual audio experience. The DCASE2023 challenge Task7 [1] is organized to stimulate research about this challenging problem. In this task, participants build a foley sound generation model that produces 100 diverse audio samples for each of the predefined sound classes: dog bark, footstep, gunshot, keyboard, moving motor vehicle, rain, and sneeze/cough.

For attempting the sound generation task, it is worth investigating recently proposed sound generation models. AudioLDM [2] has demonstrated conditional sound generation with text prompting. AudioLDM is composed of a latent diffusion model (LDM), a variational autoencoder (VAE), and a neural vocoder. The LDM is conditioned on a text prompt through the Contrastive Language-Audio Pretraining (CLAP) embedding. The latent representation is provided by the VAE learned to autoencode a mel-spectrogram into a compressed latent space. The neural vocoder is based on HiFi-GAN [3], which decodes a waveform from the mel-spectrogram. Tango [4] has been proposed to enhance the text prompting functionality of AudioLDM using an instruction-tuned large language model (LLM) instead of the CLAP embedding. Although such text prompting models have shown to be effective in fine-grained guidance for audio generation, existing models cannot utilize given class-specific audio examples to resemble, which should be incorporated for the "category-to-sound" task [1].

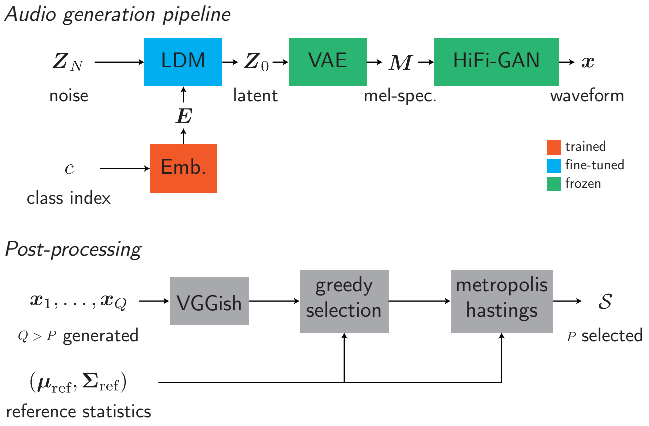

Figure 1: System overview. The audio generation pipline (top) has three elements. The core is a latent diffusion model (LDM) with class-conditional embeddings (Emb.). We use pre-trained variational-autoencoder and HiFi-GAN vocoder for the reconstruction. The samples produced are then filtered during post-processing (bottom) by greedy and Metropolis-Hastings optimization.

Therefore, we modify an existing implementation of Tango [1] to enable sound-class-based guidance instead of text prompting. The class-conditioned LDM is trained using the development set of audio with corresponding sound class labels. The model is initialized with a pretrained model of Tango, which was trained with AudioCaps [5] dataset and Flan-T5 [6] LLM. The conditioning part based on Flan-T5 is replaced with a simple linear embedding layer to realize sound-class-based conditioning. Moreover, we propose a postprocessing filter that selects a subset of generated samples to match a distribution of the target sound class. The postprocessing filter adopts a greedy backward selection strategy that iteratively drops a sample to achieve the minimum Fréchet audio distance (FAD). Our experiments show that our system significantly outperforms the baseline system provided by the task organizers in terms of FAD.

2. SYSTEM OVERVIEW

An overview of our submitted system is depicted in Figure 1. Our system adopts a similar pipeline with Tango [4], where a latent generator based on LDM, a latent-to-mel decoder using VAE, and a mel-to-wav vocoder are cascaded. Our LDM accepts a sound class index c as a conditioner instead of a text prompt. We use pretrained

[1] https://github.com/declare-lab/tango

VAE and HiFi-GAN models used in AudioLDM [2] to reconstruct a waveform from the latent representation. After the audio generation pipeline, a postprocessing filter is employed to drop irrelevant samples to match the distribution of a target sound class. In the following subsections, we describe our implementation of the modules.

2.1. Latent diffusion with sound-class-based conditioning

Our LDM transforms a sampled Gaussian noise $\boldsymbol{Z}_N \in \mathbb{R}^{C\times\frac{T}{r}\times\frac{F}{r}}$ into a latent representation $\boldsymbol{Z}_0$ through N reverse diffusion steps with a UNet-based neural network. T is the number of mel-spectrogram frames, F is the number of mel-filter bins, C is the number of channels in latent space, and r is the compression level of VAE. The neural network receives a L-length sequence of d-dimensional embedding vectors $\boldsymbol{E} \in \mathbb{R}^{L\times d}$ transformed from the sound class indices through a linear embedding layer. The conditioner $\boldsymbol{E}$ is fed into the network through the cross-attention mechanism.

When training, our model is initialized with a pretrained checkpoint of Tango. The checkpoint is designed to receive a sequence of embedding vectors $\boldsymbol{E}$ from the Flan-T5 text encoder. We replace the text encoder with a linear embedding layer that projects a sound class index c into a d-dimentional vector. Unlike Tango, we jointly train the conditioner with the main network of LDM. Although the cross-attention mechanism for conditioning accepts a sequence of embedding vectors, which is designed to accept a text sequence, we use a single target class embedding vector as $\boldsymbol{E} \in \mathbb{R}^{1\times d}$ in this work. Given the noisy latent feature $\boldsymbol{Z}_n$, the corresponding class embedding vector $\boldsymbol{E}$, and random isotropic Gaussian noise $\bar{\boldsymbol{\varepsilon}}_m \in \mathbb{R}^{C\times\frac{T}{r}\times\frac{F}{r}}$, the neural network is trained to minimize the following loss function $\mathcal{L}$ at time step n on the basis of the theory of denoising diffusion probabilistic models [7]:

$$\mathcal{L} = \|\bar{\boldsymbol{\varepsilon}}_n - \boldsymbol{\varepsilon}(\boldsymbol{Z}_n, \boldsymbol{E}, n; \theta)\|_2^2, \tag{1}$$

$\boldsymbol{\varepsilon}(\cdot,\cdot,\cdot;\theta)$ is the neural network that outputs the estimated noise of the same shape as $\boldsymbol{Z}_n$.

During the inference, we used the procedure of denoising diffusion implicit models (DDIM) [8] to accelerate the sampling speed. In addition, we used the classifier-free guidance [9] to boost the fidelity of the sound class. Using these techniques, the deterministic backward process to obtain $\boldsymbol{Z}_{n-1}$ from $\boldsymbol{Z}_n$ can be written by

$$\begin{align}\boldsymbol{Z}_{n-1} =& \sqrt{\alpha}_{n-1}\left(\frac{\boldsymbol{Z}_n - \sqrt{1-\alpha_n}\tilde{\boldsymbol{\varepsilon}}_n}{\sqrt{\alpha_n}}\right) + \sqrt{1-\alpha_n-\sigma_n^2}\tilde{\boldsymbol{\varepsilon}}_n, \tag{2}\\ \tilde{\boldsymbol{\varepsilon}}_n =& (1+w)\boldsymbol{\varepsilon}(\boldsymbol{Z}_n, \boldsymbol{E}, n; \theta) - w\boldsymbol{\varepsilon}(\boldsymbol{Z}_n, \boldsymbol{O}, n; \theta), \tag{3}\\ \alpha_n =& 1 - \beta_n, \tag{4}\end{align}$$

where β_n and σ_n^2 are variances of Gaussian distributions in the forward and reverse process, respectively. w is a parameter of the guidance scale.

2.2. Variational autoencoder and neural vocoder

We use VAE to compress a mel-spectrogram $\boldsymbol{M} \in \mathbb{R}^{T\times F}$ into the latent space parametrized by mean and variance $\mu, \sigma \in \mathbb{R}^{C\times\frac{T}{r}\times\frac{F}{r}}$. The VAE is composed of a stack of CNN-based encoders. In the submitted system pipeline, the latent $\tilde{\boldsymbol{Z}}_0$ produced by the LDM is fed into the decoder of VAE to reconstruct a mel-spectrogram $\boldsymbol{M}$.

To reconstruct a waveform $\boldsymbol{x} \in \mathbb{R}^{T'}$ from a mel-spectrogram $\boldsymbol{M}$ given by VAE, we use the generator of HiFi-GAN [3], where T' is a length of the waveform. The module repeatedly upsamples the mel-spectrograms by a transposed convolution followed by multi-receptive field fusion (MRF). The MRF is composed of residual blocks, where each block processes the inputs by convolutions of multiple kernel sizes and dilations to capture the temporal feature by various receptive fields.

2.3. FAD-oriented postprocessing filter

The quality of the samples produced by the system, while acceptable, can be improved by over-generating and filtering. For this task, a target sample quality metric is necessary. The FAD metric used in the challenge is an obvious choice. The FAD is computed as follows. First, VGGish [10] embeddings of both the reference and generated samples are computed. The embeddings are computed for segments of 16,000 samples with half-overlap. This produces 10 embedding vectors per 4 s of generated audio. We note that the challenge abuses the metric a little bit since the VGGish model was trained on 16 kHz data, while the challenge uses 22.05 kHz. The mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ of the embedding vectors of both reference and generated audio are computed and their Fréchet distance [11] is

$$\begin{aligned}&\mathrm{FAD}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) = \\ &\qquad \|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\|^2 + \mathrm{tr}\left(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{1/2}\right).\end{aligned} \tag{5}$$

To obtain P samples, we first generate Q samples, with $Q > P$. Then, we first reduce the number of samples by greedy selection. We start with the set of all Q samples, denoted $\mathcal{S} = \{1, \dots, Q\}$. At each iteration, we remove sample k whose absence decreases the FAD most, i.e.,

$$k = \underset{\ell\in\mathcal{S}}{\arg\min}\,\mathrm{FAD}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r, \bar{\boldsymbol{\mu}}_g^{(\mathcal{S}-\ell)}, \bar{\boldsymbol{\Sigma}}_g^{(\mathcal{S}-\ell)}), \tag{6}$$

where $\bar{\boldsymbol{\mu}}_g^{(\mathcal{S}-\ell)}$ and $\bar{\boldsymbol{\Sigma}}_g^{(\mathcal{S}-\ell)}$ are the mean and covariance matrix, respectively, after removing the ℓth sample. Then, we update $\mathcal{S} \leftarrow \mathcal{S} - \{k\}$, where "$-$" here is the set difference operator. We repeat until the size of $\mathcal{S}$ is P, or no sample can be removed without the FAD increasing.

If we still have more than P samples, we apply the Metropolis-Hastings algorithm [12] to find a good sub-set of P elements. We initialize the algorithm by evaluating the FAD for 100 subsets of P samples and picking the lowest one. At each iteration of the algorithm, we randomly swap two samples. We first pick at random one of the current P samples. Then, we pick one of the discarded samples with probability inversely proportional to the embedding distance to the first sample. We swap the two samples and evaluate the FAD. If it decreases, we accept the change. If it increases, we only accept the change with a small probability that decreases over time with a linear schedule. Otherwise, we reject the change. The subset with the lowest FAD over all iterations is returned by the algorithm.

3. EXPERIMENTS

3.1. Models and hyperparameters

HiFi-GAN and VAE: We used pretrained checkpoints of HiFi-GAN and VAE used in [2]. The HiFi-GAN model was trained with

Table 1: Fréchet audio distance (FAD) with baseline and our systems. 'raw' indicates the system without the FAD filter, i.e., the first 100 samples from the audio generation pipeline were used. 'filtered' indicates our submitted system with the FAD filter.

Sound class	Baseline [1]	Ours	
		raw	filtered
dog bark	13.411	5.835	3.816
footstep	8.109	11.209	8.227
gunshot	7.951	5.790	3.427
keyboard	5.230	3.698	2.758
moving motor vehicle	16.108	11.440	6.837
rain	13.337	7.031	5.399
sneeze cough	3.770	3.658	2.741
Average	9.702	6.952	4.744

AudioSet [13]. All the training data were segmented or padded into 10 seconds and resampled to 16 kHz, i.e., $T' = 160,000$. Each audio sample was transformed into a 64-dim Mel-spectrogram ($F = 64$) with a window length of 1024, and a hop length of 160. The number of frames T was 1024 by padding 24 frames to avoid further padding with downsampling operations in VAE and LDM. The VAE model was trained with AudioSet [13], AudioCaps [5], Freesound [2], and BBCSFX [3]. The compression level r was 4, and the number of channels C was 8.

LDM: We initialized our LDM using a checkpoint of Tango [4]. The model used the conditioning vector dimension $d = 1024$. The initial checkpoint was trained with AudioCaps [5]. For finetuning, we used the DCASE2023 Task7 development set. Since the audio data were sampled at 22.05 kHz and segmented in four seconds, we resampled them to 16 kHz and padded them into 10 seconds. We set $N = 1000$ forward diffusion steps for finetuning. Our LDM was finetuned with the AdamW optimizer with an initial learning rate of 3e-5 and a linear decay learning rate scheduler. We finetuned the model for 100k training iterations, with an effective batch size of 42 using seven A100 GPUs. In the inference phase, we used DDIM [8] for 100 sampling steps and a classifier-free guidance scale of $w = 3$. As our model produces a 10-second audio segment at a 16 kHz sampling rate, we extracted the first four-second segment and resampled it to 22.05 kHz to fit the challenge rule.

Postprocessing: For each sound class, we first generated $Q = 200$ samples with the aforementioned audio generation pipeline. Then the FAD filter is applied to reduce the number of samples to $P = 100$.

3.2. Results

Table 1 shows FAD from the evaluation set with baseline and our systems. Without the proposed FAD filter, our system produced better FADs in all sound classes except for the footstep class. With the FAD filter, the FADs were significantly reduced regardless of the sound classes. The results demonstrate that the proposed audio generation pipeline can generate class-specific audio samples with sufficient diversity, and that the proposed FAD filter can select a subset of generated samples with the statistics of the target sound class.

4. CONCLUSION

We submitted a system based on class-conditioned latent diffusion model to DCASE2023 Task7: Foley sound synthesis challenge. Based on an existing model for text-to-audio generation, we finetuned our model that realizes sound-class-based conditioning. Furthermore, we successfully utilized the embedding-based statistics of target classes for filtering the generated sounds. Our submission system achieved significantly better FAD than the baseline system.

5. REFERENCES

[1] K. Choi, J. Im, *et al.*, "Foley sound synthesis at the DCASE 2023 challenge," *arXiv preprint arXiv:2304.12521*, 2023.

[2] H. Liu, Z. Chen, *et al.*, "AudioLDM: Text-to-audio generation with latent diffusion models," *arXiv preprint arXiv:2301.12503*, 2023.

[3] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 17 022–17 033.

[4] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria, "Text-to-audio generation using instruction tuned LLM and latent diffusion model," *arXiv preprint arXiv:2304.13731*, 2023.

[5] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, June 2019, pp. 119–132.

[6] H. W. Chung, L. Hou, *et al.*, "Scaling instruction-finetuned language models," *arXiv preprint arXiv:2210.11416*, 2022.

[7] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[8] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. ICML*, 2021. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[9] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[10] S. Hershey, S. Chaudhuri, *et al.*, "CNN architectures for large-scale audio classification," in *Proc. ICASSP*, New Orleans, LA, USA, Mar. 2017, pp. 131–135.

[11] D. C. Dowson and B. V. Landau, "The Fréchet distance between multivariate normal distributions," *Journal of Multivariate Analysis*, vol. 12, no. 3, pp. 450–455, Sept. 1982.

[12] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, June 1953.

[13] J. F. Gemmeke, D. P. Ellis, *et al.*, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*. New Orleans, LA, USA: IEEE, 2017, pp. 776–780.

[2] https://freesound.org/

[3] https://sound-effects.bbcrewind.co.uk

[4] https://huggingface.co/declare-lab/tango