THE X-LANCE SYSTEM FOR DCASE2023 CHALLENGE TASK 7: FOLEY SOUND SYNTHESIS TRACK B

Technical Report

Zeyu Xie, Xuenan Xu, Baihan Li, Mengyue Wu, Kai Yu

MoE Key Lab of Artificial Intelligence
X-LANCE Lab, Department of Computer Science and Engineering
AI Institute, Shanghai Jiao Tong University, Shanghai, China
{*zeyu_xie, wsntxxn, lbh0612, mengyuewu, kai.yu*}@sjtu.edu.cn

ABSTRACT

This report describes the system submitted to the Detection and Classification of Acoustic Scenes and Events (DCASE) 2023 challenge Task 7: foley sound synthesis track B. We first train a VQ-VAE model to learn the discrete representation of the audio spectrogram. Then an auto-regressive model is trained to predict discrete tokens based on input conditions. Finally, a trained vocoder converts the generated spectrogram into a waveform, where the spectrogram is restored from predicted tokens by VQ-VAE decoder. To achieve higher accuracy, fidelity and diversity, we introduce some training schemes, including (1) a discriminator model to filter audio; (2) mixup method for data augmentation; (3) clustering methods for better training. Our best system achieved a FAD score of 6.99 averaging on all categories.

***Index Terms*—** Foley Sound Synthesis, conditional audio geneartion, VQ-VAE, auto-regressive model

1. INTRODUCTION

Recently, the generative model has witnessed great development, and audio generation also becomes one of the most popular topics. Audio generation based on free text conditions has a wide range of application scenarios, such as editing the background sound effects of videos for movies and games [1], and generating sound effects to make virtual reality scenes more realistic.

To address the difficulty of directly generating audio spectrogram, VAE (Variational AutoEncoder) or VQ-VAE (Vector Quantised Variational AutoEncoder) mechanisms are widely used for extracting latent representations of audio spectrogram [2, 3, 4, 5, 6]. During the next stage of training, the prediction models learn to predict the audio latent representation based on conditional inputs. Recent works have introduced different types of prediction models, such as CNN-based Pixelsnail model [6], diffusion model [2, 3, 5] and the Transformer [4].

In DCASE 2023 challenge task 7 track B [7], the input text conditions are limited to 7 sound tags, and the available audios for training are limited to the 4850 provided clips. It is a challenge to train a model efficiently under the limited data set and balance the accuracy, fidelity and diversity in sound generation. To address the above issues, we train a VQ-VAE model to extract discrete tokens as latent representations. Then we adopt a Transformer type auto-regressive model to accelerate model training and introduce some training schemes to balance model performance.

The remaining part of this report is ordered as follows. Section 2 describes our methodology. Section 3 and section 4 illustrate our experimental setup and result, respectively. Finally, section 5 summarises our work.

2. METHODOLOGY

In this section, we describe our training system which consists of a representation learning model and a generation model, as well as our training schemes.

2.1. Discrete representation learning

In audio generation tasks, it is difficult to directly process in continuous and complex spectral spaces. One solution is to represent the audio spectrogram by extracting discrete tokens to reduce the burden on the model. We employ a VQ-VAE-based model following Liu et al. [6], which consists of a codebook, an encoder and a decoder.

The encoder converts the input spectrogram $a \in \mathbb{R}^{M \times T}$ into a latent representation $\{p_{m,t}\} \in \mathbb{R}^{M/2^c \times T/2^c \times D}$, where M, T, D, c denote number of Mel bands, time dimension, codebook dimension and downsampling rate, respectively. In order to capture both local and global information, a Multi-scale convolutional Encoder is adopted, where 4 convolutional blocks with different kernel sizes extract latent features in parallel and add them together.

$$\{p_{m,t}\} = \text{Encoder}(a) = \sum_i \text{CNNBlock}_i(a) \tag{1}$$

The codebook contains $|V|$ discrete tokens $\{v_j\}$, each of which is a D-dimensional vector representing one of the attributes. The latent representation $\{p_{m,t}\}$ is quantized through L-2 distance:

$$\begin{aligned} \hat{p}_{m,t} = \text{Quantize}(p_{m,t}) &= \min_{v_j} ||v_j - p_{m,t}||_2 \\ \text{ind}_{m,t} &= \min_{j} ||v_j - p_{m,t}||_2 \end{aligned} \tag{2}$$

where $\{\text{ind}_{m,t}\}$ indicates the token index list that represents audio feature in discrete space.

The quantized $\{\hat{p}_{m,t}\}$ are restored to the spectrogram via the Decoder, which has a similar but reverse structure with the Encoder, except for using fixed transposed Convolutional layers rather than parallel convolutional layers with different kernel sizes.

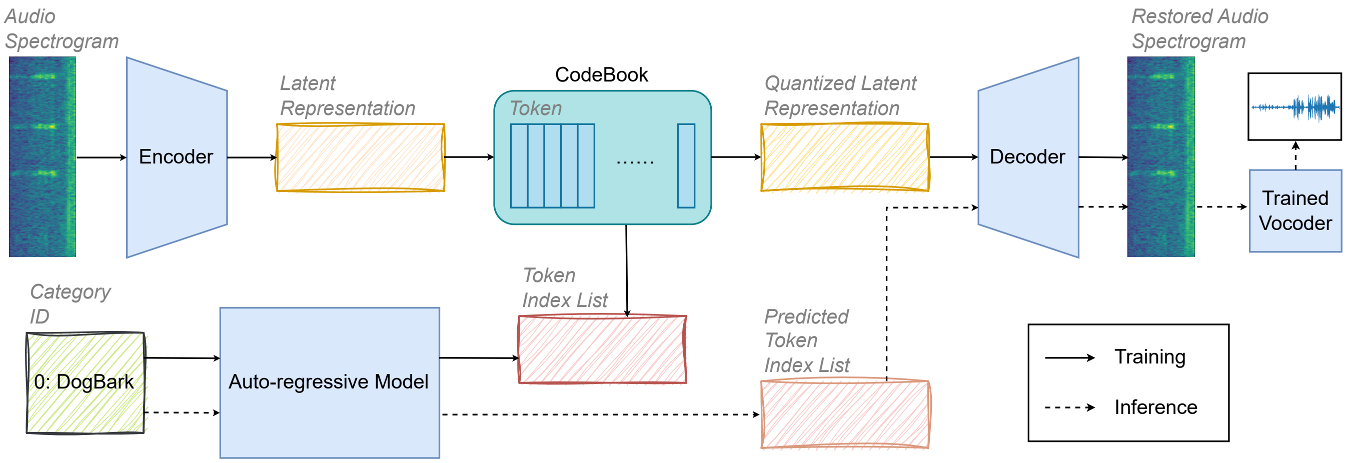

Figure 1: The diagram of conditional audio generation framework. The solid arrows represent the process of model training, including (1) VQ-VAE learning discrete audio representations, and (2) auto-regressive models learning and predicting token index lists based on input event labels. The dashed arrows represent the process of audio generation. Given the input event label, the auto-regressive model predicts the token index list, which is then restored to a spectrogram by the VQ-VAE decoder. Finally, the audio is generated using a pre-trained Vocoder.

$$\hat{a} = \text{Decoder}(\{\hat{p}_{m,t}\}) \tag{3}$$

The VQ-VAE is trained end-to-end:

$$\mathcal{L}_{\text{VQ-VAE}} = ||a - \hat{a}||_2^2 + ||\text{sg}\,[p] - \hat{p}||_2^2 + \beta||\text{sg}\,[\hat{p}] - p||_2^2 \tag{4}$$

where “sg” denotes stop gradient and β is a regularization parameter.

2.2. Auto-regressive generation

The generation model takes the category tag of sound events as the input to predict the codebook entries $\{\text{ind}_{m,t}\}$, which can be restored into the spectrogram by the VQ-VAE decoder. The entries are stretched into a sequence to facilitate auto-regressive generation. In order to accelerate training, we use Transformer [8] since it can predict the distribution of all tokens of a sequence in parallel.

Transformer predicts tokens based on previous ones and a feature sequence. Self-attention first encoded previous tokens into embeddings. Then cross-attention is used between the input feature sequence and embeddings. Depending on the ways to generate the feature sequence, we adopt two types of architectures. The first architecture, marked as $Transformer$, uses a Transformer encoder to obtain the feature sequence from the category tag, which has a full Transformer architecture. The second directly uses a look-up table to transform the tag into the sequence, marked as $TransformerDecoder$.

2.3. Training schemes

We introduce some training schemes to balance model performance with limited data.

Discriminator: In order to improve the fidelity of generated audio, we train a discriminator model as a scorer. The audios in training set and generated audios are regarded as positive and negative examples, respectively. The discriminator takes Mel spectrogram as input to score audios. During generating stage, the audios scored higher by the discriminator will be prioritized.

Table 1: Results of Auto-tuning Spectral Clustering

Category	Number of subcategories
DogBark	2
Footstep	2
GunShot	8
Keyboard	5
MovingMotorVehicle	8
SneezeCough	2

Mixup: One of the methods for data augmentation when dealing with small datasets is to do mixup. For two adjacent audios in the same category, we alternately select tokens from the VQ-VAE extracted token list to get a mixup sample. By doing this, the amount of training data increases by 1/2 for the generation model.

Clustering: The audios within each sound category may still have different distributions, so we further cluster them for better training and diversified generation. The first clustering method is to perform k-mean clustering on the token list extracted by VQ-VAE, marked as $cluster\_tokens$. Each class is subdivided into two subcategories.

Another clustering, marked as $cluster\_audios$, is conducted based on several audio features (e.g. spectrum centroid, spectrum bandwidth, zero crossing rate, root mean square, spectrum flatness, mfcc and mel spectrum). $cluster\_audios$ consists of the auto-tuning spectral clustering algorithm[9] followed by a k-mean algorithm, the number of subcategories are shown in table 1.

3. EXPERIMENTAL SETUP

The model setting of VQ-VAE followed Liu et al [6], so did the Mel extractor and the HiFi-GAN Vocoder. The Transformer model consists of 1 layer encoder and 2 layers decoder, while the Transformer Decoder model only has 1 layer of decoder. In both of them, hidden dimension, feed forward dimension and attention heads are set to

Table 2: FAD scores of system performance. “#” indicates an ensemble system that integrates the performance of the best categories of models TransformerDecoder, Transformer, and Transformer with mixup (e.g. “dog bark” for Transformer). $Discriminator100$ and $discriminator300$ are adopted to filter audios scored less than the threshold 0.5. “✓” indicates four systems that we submit.

System	dog bark	footstep	gunshot	keyboard	moving motor vehicle	rain	sneeze cough	average FAD
Baseline	13.411	8.109	7.951	5.23	16.108	13.337	3.77	9.702
TransformerDecoder	11.831	7.162	11.57	6.599	12.534	10.621	3.095	9.059
Transformer ✓	8.036	6.987	8.185	**3.494**	13.560	9.265	2.313	7.406
Transformer+Mixup	6.812	**6.894**	**7.813**	4.196	20.299	**9.263**	**2.164**	8.206
Ensemble (#) ✓	6.812	**6.894**	**7.814**	**3.494**	12.534	**9.263**	**2.164**	6.996
# + discriminator100 ✓	7.012	6.948	7.912	3.599	11.619	9.348	2.493	**6.990**
# + discriminator300 ✓	**6.657**	7.762	8.198	3.703	**11.440**	9.813	2.655	7.175

Table 3: FAD scores to illustrate the impact of clustering. “ +” indicates the following training schemes are implemented with Transformer.

System	dog bark	footstep	gunshot	keyboard	moving motor vehicle	rain	sneeze cough	average FAD
Transformer	**8.036**	6.987	**8.185**	**3.494**	13.560	9.265	**2.313**	**7.406**
+ Cluster_tokens	9.97	**5.771**	10.081	4.63	14.810	**8.897**	3.994	8.33
+ Cluster_audios	11.630	7.260	10.775	5.421	**11.334**	10.316	3.433	8.596

512, 2048 and 8 respectively.

The VQ-VAE model is trained for 800 epochs via loss function calculated by equation 4. Cross-entropy loss is adopted to train the auto-regressive model for 800 epochs. Optimization is performed using Adam optimizer with a learning rate 3×10^{-4} in both two training stages. The model with the minimum loss value will be adopted during the audio generation phase.

The discriminator is a 1-layer Transformer Encoder with 8 attention heads and a hidden size of 512. The audio clips in the training set are regarded as positive samples. 100 generated audios for each category generated by the Transformer are taken as negative samples, which are used to train the discriminator model $discriminator100$. For another $discriminator300$, there are 300 negative samples in each category. The audio with a score less than the threshold (=0.5) determined by the discriminator will be discarded during generation. The discriminators are trained for 800 through BCEloss and the one with minimum loss is selected.

4. RESULT

Frechet Audio Distance (FAD) [10] is adopted to measure performance, which is calculated between generated and reference audios. Results are illustrated in table 2 and table 3.

Results in table 2 shows that Transformer Decoder, Transformer, and Transformer with mixup achieve better performance than baseline model. To integrate their strengths, an ensemble model generate audio using the best model of each category (e.g. “dog bark” for Transformer). Discriminator that filters out audios with scores below the threshold further optimizes system performance. The ensemble system together with $discriminator100$ achieves the best FAD score 6.99 in our experiment. We submit four best models as indicated by the “✓” in table 2.

Table 3 illustrates that although there is no improvement in the overall system, clustering helps system to achieve good results in some specific categories. How to use clustering reasonably to assist model training may be one of the future directions. In addition, in the experiment, we notice that the performance of these models greatly affected by random seed due to sampling method.

5. CONCLUSION

In this report, we describe our conditional audio generation system, including a VQ-VAE module learning the latent representation of audio and an auto-regressive model predicting the index token list. In addition, in order to balance the performance of different aspects of the model, we also introduce some training schemes such as discriminator, mixing method and clustering. The FAD distance between reference audios is used to measure the quality of generated audios. The best average FAD 6.99 is obtained by an ensemble system for generation together with a discriminator to filter.

6. REFERENCES

[1] Q. Kong, Y. Xu, T. Iqbal, Y. Cao, W. Wang, and M. D. Plumbley, “Acoustic scene generation with conditional samplernn,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 925–929.

[2] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, “Diffsound: Discrete diffusion model for text-to-sound generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[3] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, “Audioldm: Text-to-audio generation with latent diffusion models,” *arXiv preprint arXiv:2301.12503*, 2023.

[4] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, “Audiogen: Textually guided audio generation,” *arXiv preprint arXiv:2209.15352*, 2022.

[5] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, “Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models,” *arXiv preprint arXiv:2301.12661*, 2023.

[6] X. Liu, T. Iqbal, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, “Conditional sound generation using neural dis-

crete time-frequency representation learning," in *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2021, pp. 1–6.

[7] 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," *arXiv preprint arXiv:2304.12521*, 2023.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[9] T. J. Park, K. J. Han, M. Kumar, and S. Narayanan, "Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap," *IEEE Signal Processing Letters*, vol. 27, pp. 381–385, 2019.

[10] 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.