FOLEY SOUND SYNTHESIS BASED ON GENERATIVE ADVERSARIAL NETWORKS USING ONESELF-CONDITIONED CONTRASTIVE LEARNING

HaeChun Chung, Yuna Lee, JaeHoon Jung

AI2XL Lab.,
Institute of Convergence Technology,
KT Corporation

ABSTRACT

The creation of sound effects, such as foley sounds, for radio or film has traditionally relied on the expertise of skilled professionals. However, synthesizing these sounds automatically without expert intervention presents significant challenge. Particularly, when the available data is limited, this challenge becomes even more compounded. This often leads to a lack of diversity in the generated data. In this paper, we propose effective GAN frameworks, O2C-GAN and OC-SupConGAN for foley sound synthesis in this situation. The proposed frameworks use a new learning method, oneself-conditioned contrastive learning (OCC learning), to solve problems encountered in small dataset. The OCC learning is a method that aims to expand the diversity of data while preserving the inherent attributes of each class within the data. Experiments show that the proposed framework outperforms baseline schemes, ranking 2nd in DCASE2023-T7 Track B with a FAD score of 5.023 on the evaluation set.

Index Terms— Foley sound synthesis, Generative Adversarial Network, Contrastive Learning

1. INTRODUCTION

In recent years, there have been significant advancements in the field of generative models, leading to a growing interest in generating images or sounds that fulfill specific user-defined conditions across various domains. While the audio domain has seen substantial advancements in voice synthesis for singing, text-to-speech (TTS), and music generation, the focus on generating in other acoustic domains, such as sound effects or background noises, has been relatively limited [1, 2, 3]. Notably, foley sound synthesis [4], crucial for enriching auditory experiences in narratives like radio or movies, has been received relatively little attention. Foley sounds are meticulously crafted to synchronize with on-screen events and actions, adding realism and depth to the overall sound design. However, the creation of foley sounds traditionally relies on skilled professionals manually performing and recording the necessary sounds. This expert-driven approach restricts the scalability, flexibility, and creative exploration in sound production. As a result, there is a clear need to explore automated approaches for generating user-desired foley sounds. However, tackling this challenge is accompanied by various difficulties due to the complex nature of foley sounds. Specifically, the problem is further exacerbated when the available data for training models is scarce.

To promote research in the aforementioned field, task 7: Foley sound synthesis was introduced in the DCASE challenge. This aimed to pioneer a new ground of audio synthesis and generate user-desired sounds tailored to custom environments [5]. The following task was divided into two sub-tasks: A and B. Participants were challenged to generate 4-second audio clips with a dataset consisting of about 800 data per class given from the challenge. Task B allows only a dataset given from the challenge, while Task A allows the use of external datasets. We participated in Task B. The requirement to train models with such a limited amount of data imposes a critical flaw for the generative models. The scarcity of data is likely to lead to problem with a lack of diversity in the generated data.

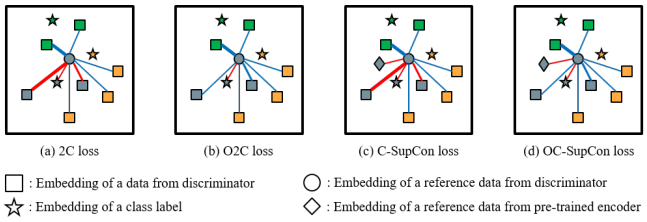

Figure 1: Schematic figure of the application of oneself-conditioned contrastive learning to different conditional contrastive losses. The color of each shape represents a class. The color of line implies the push-and-pull between the embeddings. The red line represents pulling each embedding while the blue line represents pushing each other. The thickness of the line expresses the strength of the pushing and pulling force. The thicker the line, the stronger the pull or push.

In this paper, we propose oneself-conditioned contrastive learning (OCC learning) that selectively applies label information in conditional contrastive learning methods. The OCC learning uses label information of the data itself but does not use label information between data. This extends the diversity between data while maintaining the class-specific characteristics of the data. In small dataset situations, OCC learning intentionally makes training of GAN difficult, increasing the stability of learning and solving the mode collapse problem. This can be applied to models using conditional contrastive learning method, among which we applied it to ContraGAN [6] and C-SupConGAN [7], thereby proposing O2C-GAN and OC-SupConGAN respectively. The schematic difference between applying and not applying OCC learning to the contrastive loss of each model is depicted in Figure 1.

The rest of this paper is structured as follows: In Section 2, we provide a detailed description of our proposed methods, O2C-GAN and OC-SupConGAN. Section 3 outlines the dataset used and presents the experimental setup to compare with the baseline and other variants of our approach. In Section4, we discuss the results of our experiments, and finally, the last section concludes this paper.

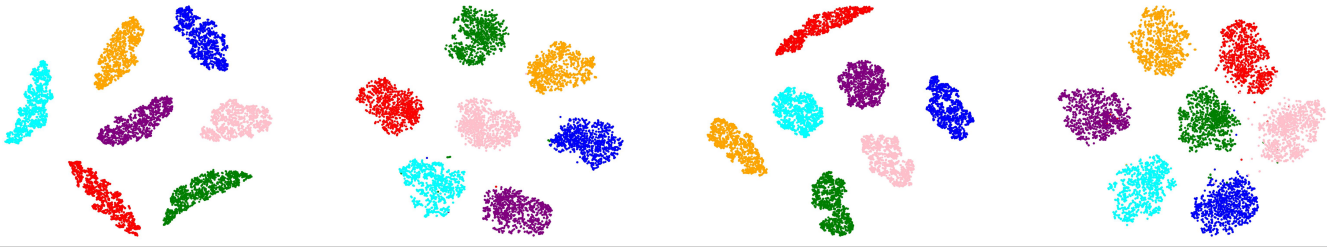

Figure 2: Illustration of a visual comparison of clusters of data embeddings generated by different trained GANs using t-SNE. The points of each t-SNE are the embeddings of the data generated by the generator in the latent space of the discriminator.

2. METHODS

We build the 2-stage system to obtain high performance of FAD score in the DCASE2023-T7 Track B. We denote the first stage as the 'Category-to-Mel spectrogram' section and the second as the 'Mel spectrogram-to-Sound' section for straightforward explanation. In the first step, we explain adversarial loss and introduce two contrastive loss functions which proposed oneself-conditioned contrastive learning (OCC learning) was fed into. We apply the following OCC learning to ContraGAN and C-SupConGAN and demonstrate effectiveness.

2.1. Category-to-Mel spectrogram

Adversarial Loss GAN [8] is composed of a generator and a discriminator. Generator G intends to deceive the discriminator D with a synthetic Mel spectrogram generated from the given label information. On the other hand, the discriminator D must establish the validity of the generated Mel spectrogram and the real Mel spectrogram using label information. G takes noise z_i with label information of class i, c_i, and D takes real Mel spectrogram x_i or fake Mel spectrogram $G(z_i, c_i)$ based on the same label information c_i. We use the hinge loss function as the adversarial loss function, and each objective function for D and G are shown in the equation below.

$$l_D = -\min\left(0, -1 + D\left(x_i, c_i\right)\right) - \min\left(0, -1 - D\left(G\left(z_i, c_i\right), c_i\right)\right) \tag{1}$$
$$l_G = -D\left(G\left(z_i, c_i\right), c_i\right)$$

Oneself-Conditioned Contrastive Loss (O2C loss) We first use ContraGAN, which introduced conditional contrastive loss (2C loss) to GAN. 2C loss is a supervised method that minimizes data-to-data distances belonging to the same class and data-to-class distance and maximizes data-to-data distances belonging to the different classes using data embeddings and class embeddings. To extract embeddings for contrastive learning, we divided the discriminator D into two separate networks: D_1 and D_2. Firstly, $D(\cdot) = D_2(D_1(\cdot))$ is used for calculating adversarial loss. To extract data embeddings d_i, features of real or fake data extracted from $D_1(\cdot)$ are additionally feedforward to the projection head $h(\cdot)$. Thus, we can term $d_i = h(D_1(x_i, c_i))$ for simplicity. The class embedding is extracted by the embedding function $e(\cdot)$ and can be denoted as $e(c_i)$. Further, these features are mapped to the unit hypersphere for cosine similarity computation.

Although the 2C loss function itself produces decent performance, the small number of data per class leads to an unexpected situation. We discovered that the adversarial loss of the discriminator D falls too quickly when we implement the 2C loss function as it is in the current task. This occurrence leads to poor GAN training, further to mode collapse problem [9] that produces similar outputs within the class. To resolve this tragic event, we introduce oneself-conditioned contrastive learning (OCC learning) to the original 2C loss function, and term this oneself-conditioned contrastive loss (O2C loss). As aforementioned above, 2C loss uses label information for both data-to-data and data-to-class relations. O2C loss ignores label information for data-to-data relations and uses label information only for data-to-class relations. The training guidelines for 2C loss and the O2C loss are outlined in (a) and (b) of Figure 1. As shown in the figure 1, the O2C loss maximizes distances between all data embeddings, regardless of whether the data belong to the same class or different classes, and only minimizes data-to-class distance. This optional use of label information distributes data within a class while maintaining the class's distinctiveness. The effect of O2C loss is shown in Figure 2. This solves the mode collapse problem by securing the diversity of data while generating well-classified data according to class and shows tremendous performance improvement. The following data-to-data distance $d2d_{i,j}$ and data-to-class distance $d2c_{i,i}$ can be denoted as the equation 3.

$$d2d_{i,j} = \exp\left(d_i \cdot d_j / \tau_d\right), \; d2c_{i,i} = \exp\left(d_i \cdot e(c_i) / \tau_c\right) \tag{2}$$

With the aforementioned notation, the O2C loss function is defined as follows:

$$l_{O2C}\left(d_i, c_i\right) = -\log\left(\frac{d2c_{i,i}}{d2c_{i,i} + \sum_{k=1}^{N} 1_{i \neq k} \cdot d2d_{i,j}}\right) \tag{3}$$

The $\cdot$ symbol denotes the inner (dot) product, and N is batch size. The hyperparameter τ is applied to control the pushing and pulling forces for distance between embeddings; the larger τ, the weaker the force, and the smaller τ, the stronger the force. C-SupConGAN differentiates the temperature hyperparameter for data-to-data distance τ_d and data-to-class distance τ_c to boost performance. We set $\tau_d = 0.1, \tau_c = 0.1$ by default, but we also conducted the experiment with different values of the two variables, which leads to better results.

Oneself-Conditioned Supervised Contrastive loss (OC-SupCon loss) C-SupConGAN, an advanced version of ContraGAN, uses

pre-trained data features to support the feature learning of the discriminator. The conditional supervised contrastive loss (C-SupCon loss) appends data-to-source relation to prior 2C loss. For data-to-source relation, C-SupCon loss uses reference data embedding extracted from the pre-trained encoder $f(\cdot)$. This aids GAN's feature learning, thereby reduces the instability of the training process and enable long-term training, and ultimately improved performance. Nonetheless, mode collapse still occurs when C-SupConGAN is applied to the current task as it is. Therefore, we also apply the OCC learning to C-SupCon loss and call it OC-SupCon loss.

$$d2s_{i,i} = \exp\left(d_i \cdot f(x_i)/\tau_c\right) \tag{4}$$

In the same way, the OC-SupCon loss can be described as follows:

$$l_{OC-SupCon}\left(d_i, c_i\right) = -\log\left(\frac{d2s_{i,i} + d2c_{i,i}}{d2s_{i,i} + d2c_{i,i} + \sum_{k=1}^{N} 1_{i\neq k} \cdot d2d_{i,k}}\right) \tag{5}$$

The conceptual difference between the C-SupCon loss and the OC-SupCon loss can be schematically confirmed in Figure 1.

We used ResNet18 [10] as the encoder network $f(\cdot)$, and it was pretrained with Supervised Contrastive Learning (SupCon) [11] loss function. For audio augmentation, we used fade in/out and time masking during the pretraining process. After the pretraining process is completed, we proceed with classification finetuning and classification evaluation. Since additional dataset such as the evaluation dataset was not open to the public, we could only evaluate the performance of classification on the training set. The classification accuracy achieved 100%, which may appear as overfitting, but we can infer that the pretrained encoder network $f(\cdot)$ is capable of extracting high-quality audio embeddings from the training set. Thus, we use the data embedding $f(x_i)$ extracted from the pretrained encoder $f(\cdot)$ as a reference to the data embedding d_i extracted from the discriminator.

Our total system is optimized through two types of loss function, which is the combination of adversarial loss and O2C loss function and the combination of adversarial loss and OC-SupCon loss function. O2C loss or OC-SupCon loss is expressed as l_C. In this way, total loss function $\mathcal{L}$ can be described:

$$\mathcal{L}_D = \frac{1}{N}\sum_{k=1}^{N} l_D + \frac{1}{N}\sum_{k=1}^{N} l_C,\ \mathcal{L}_G = \frac{1}{N}\sum_{k=1}^{N} l_G + \frac{1}{N}\sum_{k=1}^{N} l_C \tag{6}$$

$$\mathcal{L} = \mathcal{L}_D + \mathcal{L}_G \tag{7}$$

We term GAN using O2C loss as O2C-GAN, and OC-SupCon loss as OC-SupConGAN.

2.2. Mel spectrogram-to-sound

After the training on the first stage, the trained generator network G have the ability to generate Mel-spctrograms from class categories. During the second stage, a pre-trained vocoder network transforms the generated Mel-spectrogram into a time-domain digital audio signal. Instead of proposing a new vocoder network, we leverage the pre-trained vocoder network, HiFi-GAN [12], provided by the DCASE challenge.

3. EXPERIMENT

We design our experiments for three purposes. First, we demonstrate the effectiveness of oneself-conditioned contrastive learning (OCC learning). The performances of models with or without OCC learning are compared. Second, we examine the effect of the temperature hyperparameter τ on the frameworks by adjusting τ_c. Third, we verify the two proposed models exceed the performance of the baseline system.

3.1. Experiment metrics

We use Frechet Audio Distance (FAD) [13]. FAD is a standard metric for music enhancement and is very useful in that it is a reference-free evaluation metric. FAD can be employed even in the absence of a ground truth reference audio because it is calculated from collections of hidden representations of created and real samples. The FAD score can be computed by multivariate Gaussians between the generated data set and the actual audio data set, which can be referred to as the reference embeddings.

3.2. Implementation Details

We use the log mel-band energies of input audio as an audio feature. We set the frame length to 1024, and hop size to 256. All the models we train are devised to generate 80×344 mel spectrogram. Initially, we employed the learning rates used in C-SupConGAN to train our proposed models. The generator was trained with a learning rate of 0.0001, while the discriminator was trained a learning rate of 0.0004. However, the small amount of dataset led to the circumstance of discriminator D learning too quickly. Thus, we set both learning rates equally to 0.0001. For all models, we use Adam optimizer [14] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ for training. For contrastive learning, we build a 2-layer projection layer $h(\cdot)$ which embeds the output of the portion of discriminator network D_1 to 128-dimension. During training, we freeze the weight of pretrained encoder network $E(\cdot)$.

3.3. Dataset

The DCASE2023-T7 Track B development set contains 4,850 labeled sound fragments, classified into 7 categories: dog bark, footstep, gunshot, keyboard, moving motor vehicle, rain, and sneeze/cough. Each sound was fitted to a length of 4 seconds, and zero-padded or segmented if necessary. All audio was transferred to mono 16-bit 22,050 Hz sampling rate [5]. As we are participating in subtask B, we do not use any external sources.

4. RESULTS

	w/o OCC learning		w. OCC learning	
	ContraGAN	C-SupConGAN	O2C-GAN	OC-SupConGAN
FAD	12.667	12.552	5.480	5.230

Table 1: The comparison of FAD score on two baselines with and without OCC learning.

Method	DT	CT	DogBark	Footstep	GunShot	Keyboard	Vehicle	Rain	Sneeze	Average
O2C-GAN	0.1	0.1	2.784	4.370	4.667	3.555	17.511	3.899	1.577	5.480
O2C-GAN	0.1	1.0	3.348	3.990	3.495	4.074	14.861	3.529	1.865	5.023
OC-SupConGAN	0.1	0.1	2.616	3.739	6.322	4.089	14.172	4.304	1.371	5.230
OC-SupConGAN	0.1	1.0	4.854	3.103	4.790	3.665	13.604	3.727	1.435	5.026

Table 2: The comparison of FAD score on our proposed methods submitted to the DCASE2023-T7 Track B.

4.1. Effectiveness of OCC learning

We demonstrate the effectiveness of oneself-conditioned contrastive learning (OCC learning) by comparing the cases with and without OCC learning for two different GAN. We use t-SNE [15] to visually compare clusters of embeddings of data generated by different trained GANs. In Figure 2, the points of each t-SNE are the embeddings of the data generated by the generator in the latent space of the discriminator. Generated data are compared quantitatively using FAD and it is shown in Table 1.

The GANs which do not use OCC learning train the model in a way that the distance between the data embedding and the data's own condition embeddings as well as the distance between data embeddings belonging to the same class becomes close. In ContraGAN, class embedding is used as condition, and in C-SupConGAN, class embedding and pretrained data embedding are used as condition. When the dataset with a large amount of data per class is used, this helps the data to cluster for class distinction. When a small dataset with fewer data per class is used as in the current task, this causes the data belonging to the class to clump too much, resulting in a decrease in the diversity of data. As a result, the loss of discriminator D drops rapidly, resulting in poor training of GAN. The GANs using OCC learning, O2C-GAN, and OC-SupConGAN, optimize the model so that the distance between the data embedding and the data's own condition embeddings becomes close as in the previous loss function, but the distance between data embeddings belonging to the same class becomes far. This expands data clustering, amplifying the diversity among data belonging to the same class while maintaining class distinctiveness by retaining class attributes in data. As a result, by making learning task difficult, GAN training becomes stable, and various and higher-quality data are generated. Moreover, OC-SupConGAN leverages additionally pre-trained data embeddings as the condition to enhance the subjectivity of the data. Consequently, it leads to a broader dispersion of data and improves performance compared to O2C-GAN. These effects are visually illustrated in Figure 2 and shows a significant performance improvement in Table 1.

4.2. Performance Comparison

Unlike ContraGAN, C-SupConGAN uses the different temperature hyperparameters τ, which controls the strength of pulling or pushing between embeddings, for data-to-data distance τ_d and data-to-condition distance τ_c. The higher the τ value, the weaker the strength, the lower the τ value, the stronger. The temperature hyperparameter τ_d, which controls the strength of the data-to-data distance, is called DT, and the temperature hyperparameter τ_c, which controls the strength of the data-to-condition distance, is called CT. In C-SupConGAN, experiments using various values of τ_d and τ_c were conducted, and the best performance was achieved at $\tau_d = 0.1$ and $\tau_c = 1.0$. We also performed experiments not only with $\tau_d = 0.1$ and $\tau_c = 0.1$, which were used by default, but also with $\tau_d = 0.1$ and $\tau_c = 0.1$. In OCC learning, this leads the distance between all data embeddings to be strongly far, and the distance between the data embeddings and the data's own condition embeddings to be weakly close. This encourages data to maintain the unique characteristics of the class, but weaken the binding force of the class, and secure more diversity by widening the distance from other data. As a result, as shown in Table 2, the generation performance is further improved.

Class	Baseline	**Ours**	
		O2C-GAN	**OC-SupConGAN**
DogBark	13.412	3.348	4.854
Footstep	8.108	3.990	3.103
GunShot	7.952	3.495	4.790
Keyboard	5.230	4.074	3.665
Vehicle	16.107	14.861	13.604
Rain	13.338	3.529	3.727
Sneeze	3.771	1.865	1.435
Average	9.702	5.023	5.026

Table 3: The FAD score on each class of baseline scheme, O2C-GAN, and OC-SupConGAN.

Table 3 refers to the performance comparison between baseline method with our proposed methods: O2C-GAN and OC-SupConGAN. Our two techniques outperform baseline methods in every way. In particular, in 'DogBark' and 'Rain' classses, our baseline frameworks performed 4 to 5 times better than the existing baseline. We speculate that this remarkable performance is due to the proposed frameworks' ability to enhance variance of data features within the class while keeping distinct characteristic of class using our proposed OCC learning. In Table 3, we can see that improvement of FAD performance of class 'Moving Motor Vehicle' is rather low. We infer this outcome is based on insufficient variance of audio data within the class. This trait induce generation of similar data in the class regardless of the methods. To sum up, our proposed frameworks achieve the average FAD score of 5.023 and 5.026, which is the half of the baseline.

5. CONCLUSION

In this paper, we propose new GAN frameworks, O2C-GAN and OC-SupConGAN, for foley sound synthesis introduced by DCASE challenge. The proposed frameworks use a new learning method, oneself-conditioned contrastive learning (OCC learning), to solve problems encountered in small dataset. The OCC learning is a method that aims to expand the diversity of data while maintaining the class properties in the data. Our proposed frameworks achieved FAD scores of 5.023 and 5.026, outperformed the baseline framework, and ranked 2nd in the DCASE2023-T7 Track B.

6. REFERENCES

[1] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on generative adversarial networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6955–6959.

[2] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[3] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1068–1077.

[4] http://dcase.community/challenge2023/.

[5] 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.

[6] M. Kang and J. Park, "Contragan: Contrastive learning for conditional image generation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 357–21 369, 2020.

[7] H. Chung and J.-K. Kim, "C-supcongan: Using contrastive learning and trained data features for audio-to-image generation," in *Proceedings of the 2022 5th Artificial Intelligence and Cloud Computing Conference*, 2022, pp. 135–142.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, "Generative adversarial networks, 1–9," *arXiv preprint arXiv:1406.2661*, 2014.

[9] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in gans," in *2020 international joint conference on neural networks (ijcnn)*. IEEE, 2020, pp. 1–10.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer, 2016, pp. 630–645.

[11] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[12] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[13] 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.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.