VIFS: AN END-TO-END VARIATIONAL INFERENCE FOR FOLEY SOUND SYNTHESIS

Technical Report

Junhyeok Lee[1*], *Hyeonuk Nam*[2*], *Yong-Hwa Park*[2]

[1] maum.ai Inc., Republic of Korea,
[2] Korea Advanced Institute of Science and Technology, Republic of Korea,
jun3518@icloud.com, {frednam, yhpark}@kaist.ac.kr

ABSTRACT

The goal of DCASE 2023 Challenge Task 7 is to generate various sound clips for Foley sound synthesis (FSS) by "category-to-sound" approach. "Category" is expressed by a single index while corresponding "sound" covers diverse and different sound examples. To generate diverse sounds for a given category, we adopt VITS, a text-to-speech (TTS) model with variational inference. In addition, we apply various techniques from speech synthesis including PhaseAug and Avocodo. Different from TTS models which generate short pronunciation from phonemes and speaker identity, the category-to-sound problem requires generating diverse sounds just from a category index. To compensate for the difference while maintaining consistency within each audio clip, we heavily modified the prior encoder to enhance consistency with posterior latent variables. This introduced additional Gaussian on the prior encoder which promotes variance within the category. With these modifications, we propose VIFS, variational inference for end-to-end Foley sound synthesis, which generates diverse high-quality sounds.

Index Terms— Generative models, DCASE, sound synthesis

1. INTRODUCTION

Foley sound synthesis (FSS) involves the generation of various sound effects for movies. While FSS could be implemented by text-to-audio to create detailed text-conditioned sound effects [1, 2], the focus of the DCASE 2023 Challenge Task 7 is to begin the challenge with a simpler task, "category-to-sound", which aims to generate sounds based on a simple category (or class) index [3, 4]. The objective is to generate a wide range of sound examples based on a given category, where each category is represented by a single index. It is important to note that within each category of sound events, there exist diverse acoustic characteristics due to the various entities capable of producing those sound events [5, 6]. For instance, consider the sound of a dog barking. There are dogs of different species, each has different sizes and each is grown up in different environments. Such diverse entities of dogs result in significant variations in their barking sounds. As a result, the category-to-sound problem presents a unique challenge in generating diverse and distinct sounds solely based on categorical information, while accounting for the inherent variations within each category.

To address the aforementioned challenge, we draw upon the advancements in text-to-speech (TTS) which shares similarities with the category-to-sound problem in that both aim to generate audio output. Since TTS focuses on simulating the pronunciation of given input text, it provides valuable insights and methodologies that can be adapted to the category-to-sound synthesis. Our approach is based on VITS [7], which showed exceptional performance with an end-to-end framework. VITS consists of conditional variational auto-encoder (cVAE) [8], normalizing flow [9], and generative adversarial network (GAN) [10]. By incorporating VITS into the category-to-sound synthesis, we can effectively generate a wide range of sounds that align with the given categories using an end-to-end framework, while the baseline [3] requires multiple stages including auto-regressive model, VQ-VAE structure [11] and vocoder [12].

However, the category-to-sound problem presents a significant difference from the TTS approaches. While category-to-sound has to generate sound clips spanning the whole clip just from a category index, TTS focuses on generating short pronunciations based on phonemes and speaker identities. To consider the difference between TTS and category-to-sound, we made heavy modifications to the prior encoder. By adopting the prior encoder to handle longer sound events while maintaining coherence and fidelity, we effectively tackled the challenges posed by the category-to-sound problem. By adapting VITS to category-to-sound task, we propose Variational Inference for Foley sound Synthesis (VIFS). Our proposed method showcases the ability to generate high-quality sound clips with diversity, bridging the gap between category information and realistic audio representation for Foley sound synthesis. The official implementation code is available on GitHub[1].

2. VIFS

VIFS is heavily inspired by TTS studies. The architecture of VIFS is based on VITS [7], therefore it consists of posterior encoder, prior encoder, flow, decoder, and discriminators. Furthermore, we have applied various modifications by referring to previous works in speech synthesis. Figure 1 illustrates the overall training process.

2.1. Category Embedding

VITS modules, including the posterior encoder, the flow, and the decoder, are conditioned by the speaker embedding to provide conditions for the specific speaker in the dataset. Instead of the speaker embedding in VITS, we adopt category embedding for FSS to generate sounds for specific categories. The dimensionality of the category embedding is identical to the hidden dimension of the coupling layers, which is 192. Similar to the speaker embedding in VITS, the

∗ Equal contribution

[1] https://github.com/junjun3518/vifs

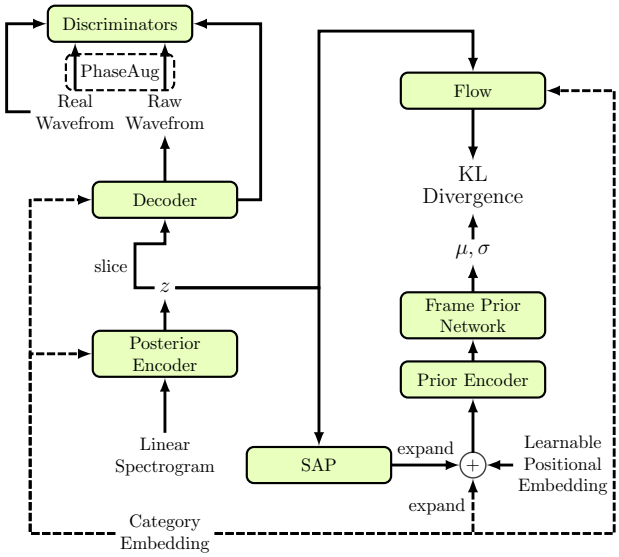

Figure 1: Overview of training procedure of VIFS.

category embedding conditions the posterior encoder, the decoder, and the flow. Furthermore, the category embedding serves as an input to the prior encoder, and its dimensionality is expanded to 344 to accommodate the time length of the latent variables from the prior encoder. Considering that prior encoder in VITS takes phonemes as input, category embedding in VIFS replaces pronunciation information as well.

2.2. Posterior Encoder and Flow

We used the architecture of the posterior encoder and flow of VITS without modification. However, FSS do not require monotonic alignment search (MAS), which is used for phoneme duration search in TTS. Therefore, VIFS omitted MAS and just calculates Kullback–Leibler (KL) divergence without MAS and length regulator for prior latent variables.

2.3. Prior Encoder

Different from TTS, the challenge requires to synthesize diverse sounds solely based on a single category index. Unlike phonemes, which typically have a duration of less than 100 million seconds, sound events span a much longer duration. For DCASE 2023 Challenge Task 7, the sound events can reach a maximum length of 4 seconds. To address this distinction and consider the different requirements of category-based sound synthesis, careful modifications need to be made to the prior encoder which encodes category index into various sound representations.

From the early experiments which did not consider clip-level consistency and were only conditioned by time-expanded category embedding, the model generated sounds often containing different events and abrupt ends, compromising naturalness. For example, generated dog bark sound often involved barking from other dogs or even the echo of a gunshot, which corresponds to different category. Some sounds abruptly ended in the middle of events and restarted. To address these problems, we introduce additional conditions for the prior encoder including category embedding, learnable positional embedding, and latent conditioning.

First, to generate sounds with clip-level consistency, the prior encoder needs to model latent variables considering temporal position. Thus, we add learnable positional embedding along time axis to expanded category embedding for conditioning positional information. The learnable positional embedding has a fixed dimension, which has an identical dimension with expanded category embedding. In addition, we apply large-kernel frame prior networks [13] to give more positional information to the model. However, we observed that while position information improves synthesized sounds' consistency within each clip, it lowers the diversity within generated dataset.

Second, to enhance diversity, we tried adding time-expanded Gaussian noise to expanded category embedding. However, just adding random Gaussian noise significantly increases KL divergence between flow outputs and prior latent variables which states given data is not modeled well with those structures. Therefore, we give another condition from the posterior latent variables to the prior encoder instead. The posterior latent variables z are compressed to a single vector by self-attentive pooling (SAP) [14] and expanded in time dimension to match the size of category embedding. During the inference, Gaussian noise is sampled for input instead of SAP latent variables since we would not have posterior latent during inference. To enforce the compressed vectors' distribution to standard normal Gaussian, we add L2 loss for the mean and standard deviation of the compressed vectors in a batch.

2.4. GAN

From the original GAN architecture of VITS which was adopted from HiFi-GAN [12], we modified several features following Lee *et al*. [15]. This reflects the methodologies of Avocodo [16] which removes unintended artifacts such as aliasing and imaging artifacts. The resultant GAN architecture is composed of the decoder and the discriminator in Figure 1. In addition, we also adopt PhaseAug [17] for adversarial networks to prevent periodicity artifacts of the non-autoregressive vocoder structure in VIFS.

2.5. Implementation Details

Without the prior encoder, other details are identical to those of VITS [7]. The frame prior network consists of 6 residual layers, each of which incorporates leaky ReLU using a negative slope 0.2, Conv1D with a kernel size of 35, and a residual connection. SAP calculates computes attention using the mean of the posterior latent variables. 4 seconds length of sounds are corresponding to spectrogram length 344 by the short-time Fourier transform with size 1024 and hop size 256. This length serves as the expansion size for the category embedding and Gaussian input to the prior encoder. The model training was performed using 4 V100 GPUs.

3. EXPERIMENTS

3.1. Dataset

We only use given dataset from the DCASE 2023 Challenge Task 7 [3], which is sampled from UrbanSound8K [18], FSD50K [19], and BBC Sound Effects[2]. This dataset consists of 7 distinct categories and total 4,850 files as illustrated in Table 1. All files within the dataset are mono recordings with a bit depth of 16 bits, a sampling rate of 22,050 Hz, and a duration of 4 seconds.

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

Table 1: Number of files on each category.

Class ID	Category	Number of clips
0	DogBark	617
1	Footstep	703
2	GunShot	777
3	Keyboard	800
4	MovingMotorVehicle	581
5	Rain	741
6	Sneeze/Cough	631

3.2. Data Length

The training dataset provided has been zero-padded to align all samples to a duration of 4 seconds [3]. To consider the adverse effect of zero padding applied to the training dataset, we found the true data length by removing the zero padding and showed a histogram plots of each category on Figure 2. Total number of audio clips corresponding to each category is shown in Table 1 for the reference. From the histograms, we can observe that more than half of the sound clips in training dataset are 4 seconds long before zero padding. These would be mostly audio clips trimmed at 4 seconds, though they were longer at first. From the histogram, we can observe that for categories of keyboard, moving motor vehicle and rain, the clips those are 4 seconds long are almost or more than 90%. It is due to their characteristics that they usually occur longer than 4 seconds. On the other hand, categories such as dog bark, gun shot and sneeze & cough, the clips with 4 seconds long are less than 70%. These sound events usually happen shortly. Nonetheless, total 76.1% of training dataset is composed of sound clips 4 seconds long. Also, among the files those are 4 second-long, there are noise from other sound sources at front and behind of the actual sound events corresponding to the category. Therefore, we concluded that taking account of the zero padded would not significantly improve the model and we rather tried to exclude the effect of zeros padded or other noises within the sound clips within the prior encoder architecture as discussed in Section 2.3.

3.3. Evaluation Metric

We used Fréchet Audio Distance (FAD) to evaluate traind model [20]. FAD is an object evaluation metric that measures the difference between distributions between the training dataset and generated dataset for each categories. The distributions are composed of the representations of audio clips, extracted by inserting the audio clips to a VGGish model trained using AudioSet [21]. Since we perform FAD on the distributions of generated dataset and evaluation dataset, lower FAD implies that generated dataset is closer to evaluation dataset thus better the performance of FSS. While FAD is the best option to perform objective evaluation on sound generative model, we should note the limitations of FAD: the trained VGGish model is not guaranteed to sufficiently working well on classification. In addition, whether it is good classifier or not, the representation extracted from the VGGish model might not "represent" the audio clips well too. With these limitations, subjective tests are required to evaluate quality of generated models more thoroughly.

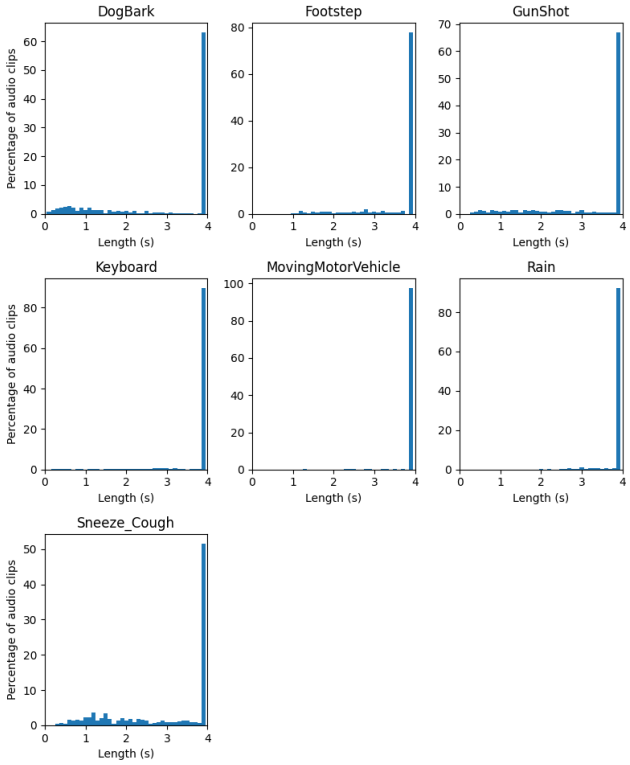

Figure 2: Histograms of data lengths after removing zero padding.

3.4. Checkpoints and Noise Scales

We used FAD to sort out the best models for each category, and chose models with four checkpoints and different noise scales. The noise scale is a factor applied to obtain the prior representation, and it is multiplied with Gaussian random vectors. A higher noise scale introduces more variance in the generated sound clips, but it may also lead to the generation of clips that significantly differ from the samples in the training dataset. Conversely, a lower noise scale produces generated sound clips that are closer to the training dataset samples but with reduced variance. To determine the optimal settings, we selected six checkpoints corresponding to 270k, 290k, 310k, 330k, 350k, and 370k steps. Among these checkpoints, we identified that four models achieve the best FAD for specific categories. We then fine-tuned the noise scale for each model to further optimize the category-wise FAD. We first conducted tests using noise scales ranging from 0.25 to 1.5 with an interval of 0.25, followed by additional tests with noise scales from 0.5 to 1.0 with an interval of 0.1.

4. RESULTS AND DISCUSSIONS

4.1. Results

Table 2 shows category-wise and average FAD on the baseline, six VIFS models and an ensemble VIFS model. The six VIFS models are chosen by procedure described on Section 3.4, and each represents model checkpoints and noise scales corresponding to each category's best category-wise FAD, as shown in Table 2 on third to eighth columns. The best FAD for each category are highlighted as bold. To further optimize the best model, we made an ensemble model by selecting the best model for each category, as indicated in

Table 2: Category-wise FAD with chosen checkpoints and noise scales. A lower FAD value indicates a better alignment between the distribution of the generated audio clips and the real audio clips in each category.

	baseline	**VIFS**						**ensemble**
# submission	-	-	3	-	-	2	1	4
# step	-	270k	270k	270k	290k	310k	330k	-
noise scale	-	0.6	0.7	1.0	0.8	0.6	0.8	-
dog bark	13.411	12.009	12.184	11.489	11.227	10.388	**8.805**	8.805
footstep	8.109	7.461	6.968	**6.638**	6.889	7.373	7.290	6.638
gunshot	7.951	7.535	**7.233**	12.440	9.860	8.091	9.392	7.233
keyboard	5.230	10.359	9.191	7.643	7.634	9.699	**6.387**	6.387
moving motor vehicle	16.108	**34.429**	34.880	37.516	39.905	37.056	37.818	34.429
rain	13.337	7.200	6.703	7.184	7.201	**6.636**	7.899	6.636
sneeze & cough	3.770	9.505	9.674	9.656	**9.283**	9.744	11.916	9.283
average w/o vehicle	8.635	9.007	8.659	9.175	8.682	8.655	8.615	7.497
average	9.702	12.638	12.405	13.224	13.142	12.712	12.787	11.344

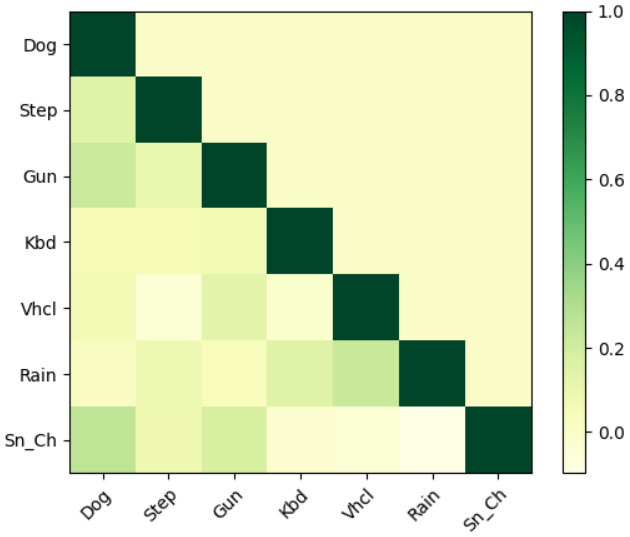

Figure 3: Cosine similarity between trained category embedding. Upper triangular region is excluded due to the symmetry.

the right column of Table 2. For the challenge submission, we selected three individual models and the ensemble model, with their corresponding indices provided in Table 2 in the second row.

From the results, VIFS outperformed the baseline for the dog bark, footstep, gunshot, and rain categories. However, VIFS still requires improvement to surpass the baseline for the keyboard, moving motor vehicle, and sneeze & cough categories. The FAD for the moving motor vehicle category was exceptionally high, causing the averaged FAD of VIFS to fall short of surpassing the baseline. However, when we considered the moving motor vehicle category as an outlier and evaluated the averaged FAD without it, we observed that the ensemble model outperformed the baseline by 13.2%.

4.2. Category Embeddings

We present the cosine similarities between the trained category embeddings in Figure 3. While we discuss four checkpoints in this study, we observe that their category embeddings are nearly identical, so we only display the embeddings corresponding to the checkpoint at 270k steps. Considering the symmetry between upper and lower triangle, we exclude the upper triangle in the visualization for brevity. Additionally, to illustrate the contrast from the cosine similarity of 1, we include the diagonal elements. As depicted in the Figure 3, it is evident that the embeddings are weakly correlated with each other. The largest cosine similarity values are 0.261, 0.229, and 0.224, corresponding to the pairs of dog bark with sneeze & cough, moving motor vehicle with rain, and dog bark with gun shot, respectively. These values are sufficiently small, indicating that the category embeddings have been effectively trained to differentiate between the different sound categories.

An interesting observation to discuss is that relatively higher cosine similarity values follow the acoustic characteristics of the sound event categories. Dog bark, gun shot, and sneeze & cough are all impulsive sounds, characterized by accentuated early parts of the sound waveform. They may occur once or in successive repeats. Similarly, moving motor vehicle and rain share similarities in terms of their stationary nature, where the spectral characteristics of these sounds rarely or slowly change over time [22]. Furthermore, both categories exhibit spectral characteristics that span a wide frequency range. In the earlier stages of the checkpoints, these categories appear to be mixed with each other in the generated sound clips. For instance, a sample of generated sneeze & cough sound contained gun shots and dog barking sounds during successive repeats. Similarly, a sample of moving motor vehicle sound resembled the sound of rain.

5. CONCLUSION

In this work, we propose VIFS, an end-to-end variational inference for FSS. With our heavily modified prior encoder, we could generate consistent sounds for each inference with high quality. In addition, techniques from speech synthesis increase the perceptual quality of synthesized sounds without multiple stages of training. As a result, we improved FAD of four categories, dog bark, footstep, gun shot and rain when compared to the baseline. While FAD for other categories are still behind the baseline, we would need to perform subjective test to compare the quality of generated sound.

6. REFERENCES

[1] 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*, 2023.

[2] 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.

[3] 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.

[4] K. Choi, S. Oh, M. Kang, and B. McFee, "A Proposal for Foley Sound Synthesis Challenge," *arXiv preprint arXiv:2207.10760*, 2023.

[5] H. Nam, B.-Y. Ko, G.-T. Lee, S.-H. Kim, W.-H. Jung, S.-M. Choi, and Y.-H. Park, "Heavily augmented sound event detection utilizing weak predictions," DCASE2021 Challenge technical report, Tech. Rep., 2021.

[6] H. Nam, S.-H. Kim, and Y.-H. Park, "Filteraugment: An acoustic environmental data augmentation method," *ICASSP*, 2022.

[7] J. Kim, J. Kong, and J. Son, "Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech," in *ICML*, 2021, pp. 5530–5540.

[8] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *ICLR*, 2014.

[9] R. T. Q. Chen, J. Behrmann, D. K. Duvenaud, and J.-H. Jacobsen, "Residual Flows for Invertible Generative Modeling," in *NeurIPS*, 2019, pp. 9916–9926.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *NeurIPS*, 2014, pp. 2672–2680.

[11] X. Liu, T. Iqbal, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, "Conditional sound generation using neural discrete time-frequency representation learning," *arXiv preprint arXiv:2107.09998*, 2021.

[12] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis," in *NeurIPS*, 2020, pp. 17 022–17 033.

[13] Y. Zhang, J. Cong, H. Xue, L. Xie, P. Zhu, and M. Bi, "VISinger: Variational Inference with Adversarial Learning for End-to-End Singing Voice Synthesis," *IEEE ICASSP*, pp. 7237–7241, 2021.

[14] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *The Speaker and Language Recognition Workshop*, 2018.

[15] J. Lee, W. Jung, H. Cho, and J. Kim, "PITS: Variational Pitch Inference without Fundamental Frequency for End-to-End Pitch-controllable TTS," *arXiv preprint arXiv:2302.12391*, 2023.

[16] T. Bak, J. Lee, H. Bae, J. Yang, J.-S. Bae, and Y.-S. Joo, "Avocodo: Generative Adversarial Network for Artifact-free Vocoder," *arXiv preprint arXiv:2206.13404*, 2022.

[17] J. Lee, S. Han, H. Cho, and W. Jung, "Phaseaug: A differentiable augmentation for speech synthesis to simulate one-to-many mapping," in *ICASSP*, 2023.

[18] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, p. 1041–1044.

[19] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: An open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 30, 2021.

[20] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A metric for evaluating music enhancement algorithms," *arXiv preprint arXiv:1812.08466*, 2019.

[21] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.

[22] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, "Frequency Dynamic Convolution: Frequency-Adaptive Pattern Recognition for Sound Event Detection," in *Proc. Interspeech*, 2022.