EDMSound: Spectrogram Based Diffusion Models for Efficient and High-Quality Audio Synthesis

Ge Zhu[*] **Yutong Wen**
Department of ECE, University of Rochester
{ge.zhu,yutong.wen}@rochester.edu

Marc-André Carbonneau
Ubisoft La Forge
marc-andre.carbonneau@ubisoft.com

Zhiyao Duan
Department of ECE, University of Rochester
zhiyao.duan@rochester.edu

Abstract

Audio diffusion models can synthesize a wide variety of sounds. Existing models often operate on the latent domain with cascaded phase recovery modules to reconstruct waveform. This poses challenges when generating high-fidelity audio. In this paper, we propose EDMSound, a diffusion-based generative model in spectrogram domain under the framework of elucidated diffusion models (EDM). Combining with efficient deterministic sampler, we achieve similar Fréchet audio distance (FAD) score as top-ranked baselines with only 10 steps and reach state-of-the-art performance with 50 steps on the DCASE2023 foley sound generation benchmark. We also reveal a potential concern regarding diffusion based audio generation models that they tend to generate samples with high perceptual similarity to the data from training set. Project page: https://agentcooper2002.github.io/EDMSound/

1 Introduction

Audio synthesis research has a long history [25]. With the development of deep generative models in recent years, data-driven audio synthesis methods have become more and more popular. In particular, diffusion models [33, 10] have led to transformative changes in audio synthesis tasks, resulting in higher quality audio samples. Current diffusion-based audio generation models utilize a cascaded system design [20, 7, 14, 38, 13, 9] to circumvent the complexity of generating sound in the temporal domain [6]. They typically involve converting waveforms into spectrograms to train a base diffusion generator. A secondary phase recovery network then converts the spectral representation back into the temporal domain [13, 9]. To further reduce the computational complexity for the base diffusion model [20, 41], a variational autoencoder (VAE) can be used to transform a mel-spectrogram into a lower-dimensional latent representation. However, a recent survey study [26] suggests that current audio generation models might not be ready for professional sound production and the most significant challenge is presented in audio quality (e.g., fidelity, sampling rate, and level of noise). This audio fidelity degradation may be caused by the cumulative errors across modules in cascaded frameworks [28]. In addition, existing diffusion-based audio generation systems are inefficient at inference time and vanilla samplers will typically take hundreds of neural function evaluations (NFEs). For instance, AudioLDM [20] takes 100-200 NFEs with denoising diffusion implicit model (DDIM)[36] for better sample quality.

In this study, we target at improving the generation fidelity in an end-to-end manner by developing an audio generation model in the complex spectrogram domain. Compared to magnitude spectrum

[*]This work was partially done during an internship at Ubisoft La Forge.

37th Conference on Neural Information Processing Systems (2023) Machine Learning for Audio Workshop.

Table 1: FAD score and relative dataset similarity score comparison of the generated audio samples on DCASE2023 task7. Baseline systems that rank 1st in the challenge, both achieved top3 official rank. 'mean' represents the average value of the experiments. 'best' represents the best one.

System	Dog Bark	Footstep	Gunshot	Keyboard	Moving Motor Vehicle	Rain	Sneeze Cough	Overall
FAD score (↓)								
Scheibler *et al* [31]	3.68	8.07	3.65	**2.78**	**7.42**	5.23	2.61	4.78
Yi *et al* [42]	3.62	5.10	5.74	3.04	9.80	5.96	1.90	5.02
Jung *et al* [4]	3.34	3.99	3.50	4.07	14.86	**3.53**	1.87	5.02
EDMSound-mean (Ours)	**2.93**	**3.22**	**3.61**	3.73	11.10	6.01	**1.27**	**4.56**
Relative dataset similarity								
Scheibler *et al* [31]	**-0.02**	**-0.04**	**-0.04**	**-0.07**	**-0.02**	-0.09	**0.03**	**-0.04**
Yi *et al* [42]	-0.05	-0.07	-0.11	-0.08	-0.03	-0.04	-0.05	-0.06
Jung *et al* (closed) [4]	-0.14	-0.11	-0.11	-0.18	-0.10	-0.17	-0.11	-0.13
EDMSound-best (Ours)	-0.05	-0.06	-0.06	-0.08	-0.02	**-0.02**	-0.11	-0.05

and phase spectrum, the real and imaginary components of the complex spectrograms exhibit clear structures and are suitable for deep learning models [29]. Compared to raw waveform modeling [28], spectral features have fewer temporal redundancies [21]. To improve generation fidelity, we build our diffusion generators within the framework of EDM [15] due to its SoTA performance in several image generation benchmarks. To accelerate the generation while maintaining similar sample quality, we use exponential integrator (EI) based ordinary differential equation (ODE) samplers during inference [43, 22, 23]. We validate our method on different sound categories using DCASE2023 foley sound generation benchmark and Speech Command 09 (SC09) [39] dataset (containing spoken digits from 'zero' to 'nine') using Fréchet distance as evaluation metric for its correlation with human perception [16].

While diffusion-based models are capable of generating high quality audio samples, it can unintentionally replicate training data [34]. Replicating data might also harm the audio generation diversity. Although similar concerns have been explored in computer vision by [34, 35], examination of this issue in audio generation remains an open research area. In our work, we answer the question of whether diffusion-based models generate audio samples with replicated contents.

To summarize, we introduce an end-to-end audio diffusion model, EDMSound, in the complex spectrogram domain. At inference time, we use EI-based ODE samplers to accelerate the generation speed. We achieve the SoTA performance on DCASE2023 foley sound generation benchmark and competitive performance on SC09 dataset in terms of Fréchet distance. We propose a method to examine the memorization issue, *i.e.,* content replication on a range of diffusion-based audio generation models on the DCASE2023 benchmark dataset. Qualitative and quantitative analysis show that our proposed model does not generate exact copies of training data. Instead, it is able to generate audio samples that match the sound timbre of the training samples but with varied temporal patterns.

2 Method

Diffusion probabilistic models (DPMs) [10, 33] involve (1) corrupting training data with increasing noise levels into normal distribution and (2) learning to reverse each step of this noise corruption with the same functional form. It can be generalized into score-based generative models [37] which employ an infinite number of noise scales so that both forward and backward diffusion processes can be described by stochastic differential equations (SDEs). During inference, the reverse SDE is used to generate samples with numerical approaches starting from the standard normal distribution. A remarkable property of the reverse SDE is the existence of a deterministic process, namely *probability flow ODE*, whose trajectories share the same marginal probability as the original SDE [37]. As a result, one can employ ODE solvers. These solvers, in contrast to SDE solvers, allow for larger step sizes, primarily because they are not influenced by the inherent randomness of the SDE [22].

EDM on Complex Spectrogram We train our diffusion model using EDM [15] which formulates the above diffusion SDE with noise scales instead of drift and diffusion coefficients. Practically, it presents a systematic way to design both training and sampling processes. To ensure that the neural network inputs are scaled within $[-1, 1]$ required by the diffusion models, we apply an amplitude transformation on the complex spectrogram inputs, $\tilde{c} = \beta|c|^{\alpha}e^{i\angle c}$ following [29], where $\alpha \in (0, 1]$ is a compression factor which emphasize time-frequency bins with low energy, $\angle c$ represents the angle of the original complex spectrogram, and $\beta \in \mathbf{R}_+$ is a scaling factor to normalize amplitudes roughly to within $[0, 1]$. Such compression technique was originally proposed for speech enhancement [2], but we found it also effective in general sounds. We adopt 2D efficient U-Net proposed in Imagen [30] as

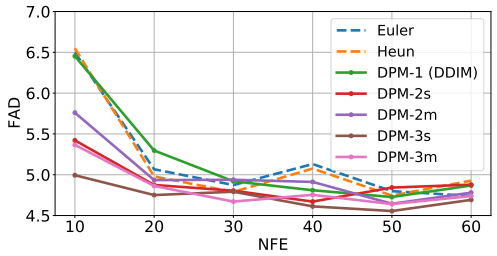


Figure 1: Comparison of FAD scores using different ODE samplers on DCASE 2023 Task 7. In DPM based samplers, the number indicates the order of the solvers, 's' represents 'singlestep' and 'm' represents 'multistep'. We use CFG with a scale of 2.0 and repeat experiments three times.

Model	Params	FID ↓	IS ↑	mIS ↑	AM ↓
Autoregressive					
SampleRNN [24]	35.0M	8.96	1.71	3.02	1.76
WaveNet [27]	4.2M	5.08	2.27	5.80	1.47
Sashimi [8]	4.1M	1.99	4.12	24.57	0.90
Non-autoregressive					
WaveGAN [5]	19.1M	2.03	4.90	36.10	0.80
DiffWave [19]	24.1M	1.92	5.26	51.21	0.68
w/ Sashimi	23.0M	1.42	5.94	69.17	0.59
ASGAN (Mel.) [1]	38.0M	0.56	7.02	**162.8**	0.56
ASGAN (HuBERT)	-	**0.14**	**7.67**	**226.7**	**0.26**
EDMSound (Ours)	45.2M	**0.14**	7.17	160.2	0.33
Train	-	0.00	8.56	292.5	0.16
Test	-	0.02	8.33	257.6	0.19

Table 2: Comparison of unconditional generation with automated metrics on SC09 dataset. FID (Fréchet Inception Distance), IS (Inception score), modified IS, and AM score are measures for generated diversity and quality.

our diffusion model backbone due to its high sample quality, faster convergence speed and memory-efficiency. During training, we use preconditioned denoising score matching as our training objective following [15]. *i.e*, $\mathbb{E}_{\mathbf{x}}\mathbb{E}_{\mathbf{n}}[\lambda(\sigma)\|D(\mathbf{x}+\mathbf{n};\sigma)-\mathbf{x}\|_2^2]$, where $\mathbf{x}$ is the training data and $\mathbf{n} \in \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. We apply classifier free guidance (CFG) [11] at the sampling stage in the conditional generation task.

Efficient Diffusion Samplers Within EDM, the probability flow ODE can be simplified as a nonlinear ODE, enabling the direct application of standard off-the-shelf ODE solvers. It is found that EI based ODE solvers have the minimum error with limited steps in a semilinear structured ODE [43], a combination of a linear function of the data variable and a nonlinear neural network term. Though such probability flow ODE only contains the non-linear neural network in EDM case, it is still beneficial to integrate EI approach shown in [43]. Therefore, we use high order EI based ODE solvers [22], *i.e.,* singlestep and multistep DPM-solvers [22, 23].

Content Replication Detection We define content replication as the presence of generated samples that are either *complete duplicates* or a *substantially similar portions* of the training samples. It is found that representations trained with full supervision or self-supervised learning can perform as well as detectors specially trained for content replication detection [34]. Since there is no existing content replication detectors for sound effects generation, we employ off-the-shelf pretrained audio representations including CLAP [40], AudioMAE [12], and PANNs [18], and compute the cosine similarity score to measure the degree of content replication. To better adapt the audio descriptors for this task, we conduct an additional fine-tuning stage: We attach multi-layer perceptrons to the frozen pre-trained audio encoders and then train with triplet margin loss [32]. To enhance the robustness of the descriptor, we incorporate data augmentation by injecting Gaussian noise, random amplitude scaling, and temporal shifting to audio samples. We first choose one audio sample as an *anchor sample* and a *positive pair* with the same audio with the above augmentation. Then, we randomly select another audio sample within the same class as the *negative pair* with data augmentation. After the fine-tuning step, we search the training set based on the cosine similarity for each generated audio sample. We identify *matched audio samples* within the training set with the top-1 similarity scores. These identified training samples are regarded as top matches for their corresponding generated audio counterparts. We then analyze the content replication behavior within these matched pairs.

3 Experiment

Experimental setup We benchmark our model, EDMSound, on DCASE2023 task7 and SC09 dataset. DCASE2023 foley sound generation [3] aims at improving audio fidelity, fit-to-category, and diversity for foley sound generation and it provides a standardized evaluation framework for different systems. It includes an open track and a closed track regarding the training dataset scale: the open track allows participants to leverage the datasets beyond the development set, while the closed track limits the dataset usage. We compare with strong baseline systems on both tracks in terms of FAD. Yi *et al.* [42] (ranked 1st officially) and Scheibler *et al.* [31] (achieved the highest FAD score on the open track) use LDMs. Jung *et al.* [4] use a GAN-based model, and ranked 1st in FAD score on the closed track. For the SC09 benchmark for unconditional generation, we retrain EDMSound without CFG following

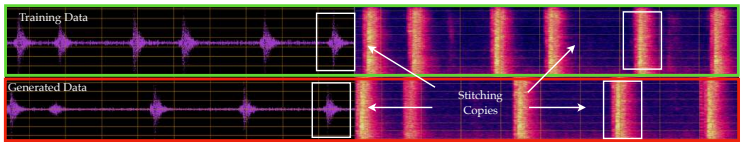


Figure 2: A matched pair of audio samples found by fine-tuned CLAP that shows a clear sign of stitching copy. The sound sources in the two audio samples show high similarity.

the best sampler choice and compare our result with baselines including autoregressive models as well as non-autoregressive models.

Sound Generation Evaluation We present our average FAD with a comparative assessment against baseline models on DCASE2023 foley sound dataset. We first compare generic ODE solvers Euler and Heun with EI-based DPM-solvers shown in Fig. 1. It can be seen that the higher order EI-based ODE solvers yield faster convergence, therefore fewer NFEs are required during inference. Particularly, the 3rd order single-step DPM-solver (DPM-3s) reaches the similar FAD as Yi *et al.* [42] with only 10 steps and achieves the best score 4.56 at the 50th step. Tab. 1 presents the class-wise and overall FAD scores for the DACSE2023 task7 challenge. Our proposed method, EDMSound, outperforms the baseline models in terms of the overall FAD in both open and closed tracks, exhibiting better performance across the majority of class-specific FAD scores as well. In Tab. 2, we present the SC09 benchmark result. Particularly, we achieved the lowest FID score without pretrained speech representations in [1]. These results underline the efficacy of our proposed methodology across diverse evaluation benchmarks.

Copy Detection Evaluation We evaluate whether the generative models produce samples copied from training dataset on the DCASE2023 task7 dataset [3] which contains a wide variety of sound types. We compare four systems: the EDMSound-best, Scheibler *et al.* [31] and Jung *et al.* [4] with the FAD scores being 4.36, 4.78, and 5.02 respectively. For content replication detection, we report our results using the fine-tuned CLAP audio encoder based on two observations: First, the similarity score distribution is significantly broader when fine-tuned with CLAP, highlighting its capability in distinguishing similar samples. Secondly, it demonstrates close alignment with human perception, as verified through manual examination. Further details regarding the ablation study on other audio descriptors can be found in Appendix B.

After mining the top-1 matched audio samples, we observe high resemblance between a number of pairs from our EDMSound-best. Fig. 2 illustrates the waveform and spectrogram of a paired audio segment capturing the sound of keyboards. Despite variations in the temporal alignment of key presses, the spectral coherence strongly suggests a common sound source originating from the same keyboard. We thereby infer that our diffusion model imitates the characteristics of its training dataset. We term the phenomenon where generated samples closely match the training samples in spectral coherence, but with relaxed temporal alignment, as *'stitching copies'*. After listening to samples from all systems, we find that the model from Scheibler *et al.* is prone to producing samples that are 'exact copies' with training samples. This strong similarity probably suggests over-fitting in the task of generating foley sounds using a large, pre-trained LDM-based model with more than 850 million parameters [7]. For a more comprehensive visual representation of this phenomenon, please refer to our project page. To quantify the overall similarity between the generated data and the training data, we use the 95-percentile similarity score of all *matched audio samples* defined in Sec. 2. To better compare the distribution difference, we compute the relative similarity by subtracting the training dataset similarity scores from the generated ones shown in the lower part of Tab. 1. Despite the fact that there are instances from generated samples showing a high degree of similarity, the overall negative relative similarity scores indicate that none of the generative models replicate their training set more than the intrinsic similarity within the training set itself.

4 Conclusion

This paper introduced EDMSound, a simple and effective end-to-end diffusion model working on the complex spectral domain implementing efficient ODE solvers. EDMSound synthesizes high quality audio improving the state-of-the-art in terms of FAD on two standard benchmark datasets (SC09 and DCASE2023 challenge task7). Furthermore, we proposed fine-tuned CLAP to examine the issue of content replication in the audio domain.

5 Acknowledgement

This work was partially supported by a Goergen Institute for Data Science (GIDS) Seed Funding Award at the University of Rochester.

References

[1] M. Baas and H. Kamper. GAN you hear me? reclaiming unconditional speech synthesis from diffusion models. In *2022 IEEE Spoken Language Technology Workshop*, pages 906–911. IEEE, 2023.

[2] C. Breithaupt and R. Martin. Analysis of the decision-directed snr estimator for speech enhancement with respect to low-snr and transient conditions. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 19(2):277–289, 2010.

[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. *arXiv preprint arXiv:2304.12521*, 2023.

[4] H. C. Chung, Y. Lee, and J. H. Jung. Foley sound synthesis based on gan using contrastive learning without label information. Technical report, Tech. Rep., June, 2023.

[5] C. Donahue, J. McAuley, and M. Puckette. Adversarial audio synthesis. In *International Conference on Learning Representations*, 2019.

[6] C. Gârbacea, A. van den Oord, Y. Li, F. S. Lim, A. Luebs, O. Vinyals, and T. C. Walters. Low bit-rate speech coding with vq-vae and a wavenet decoder. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 735–739. IEEE, 2019.

[7] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria. Text-to-audio generation using instruction guided latent diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 3590–3598, 2023.

[8] K. Goel, A. Gu, C. Donahue, and C. Ré. It's raw! audio generation with state-space models. In *International Conference on Machine Learning*, pages 7616–7633, 2022.

[9] C. Hawthorne, I. Simon, A. Roberts, N. Zeghidour, J. Gardner, E. Manilow, and J. Engel. Multi-instrument music synthesis with spectrogram diffusion. In *ISMIR 2022 Hybrid Conference*, 2022.

[10] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[11] J. Ho and T. Salimans. Classifier-free diffusion guidance, 2022.

[12] P.-Y. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer. Masked autoencoders that listen. *Advances in Neural Information Processing Systems*, 35:28708–28720, 2022.

[13] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank, et al. Noise2music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*, 2023.

[14] 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. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 13916–13932, 23–29 Jul 2023.

[15] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

[16] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi. Fréchet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

[17] P. E. Kloeden, E. Platen, P. E. Kloeden, and E. Platen. *Stochastic differential equations*. Springer, 1992.

[18] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley. PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

[19] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

[20] 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. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 21450–21474, 23–29 Jul 2023.

[21] X. Liu, H. Liu, Q. Kong, X. Mei, M. D. Plumbley, and W. Wang. Simple pooling front-ends for efficient audio classification. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE, 2023.

[22] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. DPM-Solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[23] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[24] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio. SampleRNN: An unconditional end-to-end neural audio generation model. In *International Conference on Learning Representations*, 2017.

[25] D. Moffat, R. Selfridge, and J. D. Reiss. Sound effect synthesis. *Foundations in Sound Design for Interactive Media (New York, NY: Routledge*, pages 274–99, 2019.

[26] S. Oh, M. Kang, H. Moon, K. Choi, and B. S. Chon. A demand-driven perspective on generative audio ai. *arXiv preprint arXiv:2307.04292*, 2023.

[27] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[28] S. Pascual, G. Bhattacharya, C. Yeh, J. Pons, and J. Serrà. Full-band general audio synthesis with score-based diffusion. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE, 2023.

[29] J. Richter, S. Welker, J.-M. Lemercier, B. Lay, and T. Gerkmann. Speech enhancement and dereverberation with diffusion-based generative models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[30] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[31] R. Scheibler, T. Hasumi, Y. Fujita, T. Komatsu, R. Yamamoto, and K. Tachibana. Class-conditioned latent diffusion model for dcase 2023 foley sound synthesis challenge. Technical report, Tech. Rep., June, 2023.

[32] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[33] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265, 2015.

[34] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023.

[35] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein. Understanding and mitigating copying in diffusion models. *arXiv preprint arXiv:2305.20086*, 2023.

[36] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

[37] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[38] Y. Wang, Z. Ju, X. Tan, L. He, Z. Wu, J. Bian, and S. Zhao. AUDIT: Audio editing by following instructions with latent diffusion models. *arXiv preprint arXiv:2304.00830*, 2023.

[39] P. Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[40] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023.

[41] 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.

[42] Y. Yuan, H. Liu, X. Liu, X. Kang, M. D. Plumbley, and W. Wang. Latent diffusion model based foley sound generation system for dcase challenge 2023 task 7. *arXiv preprint arXiv:2305.15905*, 2023.

[43] Q. Zhang and Y. Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023.

[44] W. Zhao, L. Bai, Y. Rao, J. Zhou, and J. Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. *arXiv preprint arXiv:2302.04867*, 2023.

Appendices

A Training and sampling within EDMs

When using complex spectrograms as the diffusion model inputs, the real and imaginary components are treated as two separate channels corrupted by Gaussian noise in the forward process. And as a result, the phase information is gradually destroyed. In the reverse sampling process, the real and imaginary channels are gradually recovered through the score network and thereby recover the phase.

A.1 Training

In DPMs, the neural networks are usually used to model the score [37] of the data at each noise level, $\nabla_{\mathbf{x}} \log(\mathbf{x}; \sigma)$, where $\mathbf{x}$ is the data and σ is the noise level. *i.e.*, the the gradient of the log probability desnity with respect to data. Or equivalently, it can be seen as training a denoiser function [15] $D(\mathbf{x}; \sigma)$ to recover clean data given corrupted versions, where $\nabla_{\mathbf{x}} \log(\mathbf{x}; \sigma) = (D(\mathbf{x}; \sigma) - \mathbf{x})/\sigma^{\mathbf{2}}$. However, its magnitude varies significantly on given noise level. To decorrelate the magnitude of the network prediction with σ, we follow the preconditioning scales on the denoiser function proposed in [15] with $c_{skip}(\sigma)\mathbf{x} + c_{out}(\sigma)F_\theta(c_{in}(\sigma)\mathbf{x}; c_{noise}(\sigma))$, where $F_\theta(\cdot)$ is the neural network output, c_{in}, c_{out} are set to ensure a unit variance for the network inputs and outputs, c_{skip} is set to minimize F_θ prediction errors scaled by c_{out}, and $\lambda(\sigma)$ is set to $1/c_{out}^2(\sigma)$ to equalize the initial training loss. Following Karras et al. [15], the desnoiser preconditioning can be written as:

$$D(\mathbf{x}; \sigma) = \frac{\sigma_{data}^2}{\sigma_{data}^2 + \sigma^2}\mathbf{x} + \frac{\sigma \cdot \sigma_{data}}{\sqrt{\sigma_{data}^2 + \sigma^2}} F_\theta\Big(\frac{\mathbf{x}}{\sqrt{\sigma_{data}^2 + \sigma^2}}; \frac{\ln(\sigma)}{4}\Big). \tag{1}$$

During training, we use $\sigma_{data} = 0.2$ as the approximation for the standard deviation of the compressed input spectrogram magnitude values. For σ, we use the log normal noise distribution with mean of -3.0 and variance of 1.0. Notice that we did not tune these distribution parameters due to insufficient computation budgets, but we found that when synthesizing sounds without much semantic information, the final performance is robust to reasonable parameters. Finally, we can write the training objective as:

$$\mathbb{E}_{p_{data}(\mathbf{x}),\epsilon,\sigma}[\lambda(\sigma)\|D(\mathbf{x} + \sigma\epsilon; \sigma) - \mathbf{x}\|_2^2], \tag{2}$$

where $p_{data}(\mathbf{x})$ represents the training data distribution, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the standard normal distribution, σ is the noise levels during training, and $\lambda(\sigma)$ is the loss weighting factor.

A.2 Sampling

Efficient samplers for DPMs can be categorized into training-based methods and training-free methods. Of all the efficient samplers, training-free methods offer the advantage of direct applicability to pretrained DPMs without necessitating additional training phases [44]. Recently proposed fast training-free samplers are using ODE solvers since SDE solvers are hard to converge within a few steps due to the fact that discretizing SDEs is generally difficult in high dimensional space and is limited by the randomness of the Wiener process [22, 17]. Another benefit of using ODE solvers is that such deterministic sampling is able to map the input data into corresponding latent representations and useful for editing.

In EDM, the probability flow ODE can be formulated as:

$$\mathrm{d}\mathbf{x} = -\dot{\sigma}(t)\,\sigma(t)\,\nabla_{\mathbf{x}} \log p\big(\mathbf{x}; \sigma(t)\big)\,\mathrm{d}t. \tag{3}$$

This simplification enables the direct application of standard off-the-shelf ODE solvers. When numerically solving ODEs, each step introduces a local error, which cumulatively results in a global error over a specified number of steps. The commonly used Euler's method is a first order ODE solver with global error linearly proportional to step size. Higher order solvers have lower global error at a given time but require multiple NFEs at each step. The second order Heun solver [15] provides a good trade-off between global error and NFE. With the advantage of EI-based ODE solvers, we apply DPM solvers [23], and our DPM-solvers samplers codebase are adapted from the implementation of k-diffusion [2] and DPM-solver official codebase [3]. During inference, we use Karras scheduler

[2] https://github.com/crowsonkb/k-diffusion

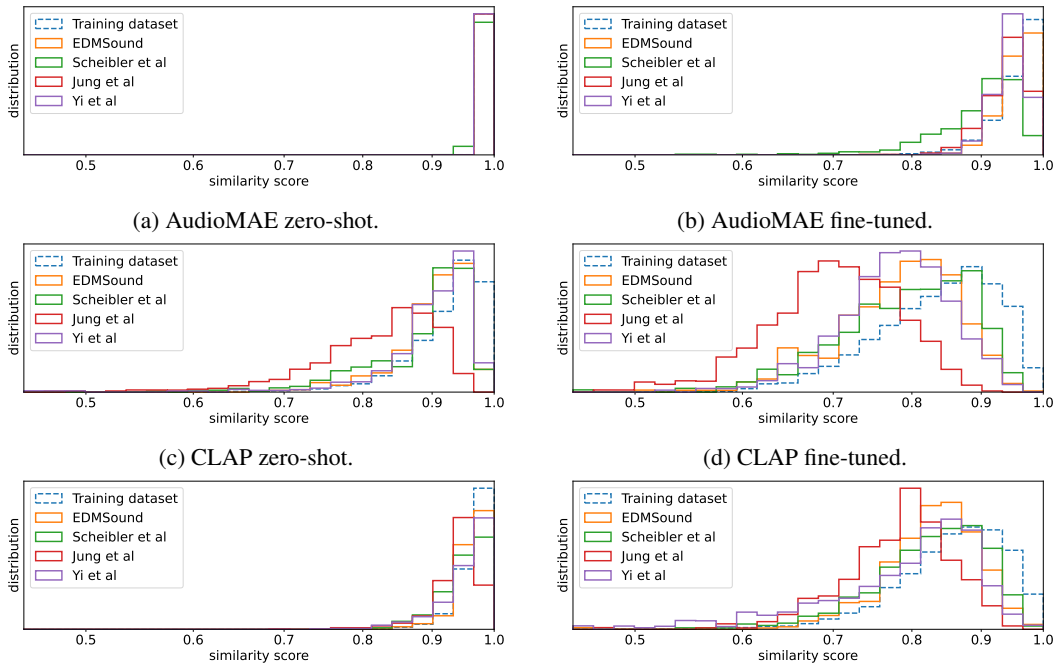
[3] https://github.com/LuChengTHU/dpm-solver

proposed in EDM with a ρ of 7.0, σ_{min} of 0.0001 and σ_{max} 3.0. We also a dynamic threshold of 0.99 following Imagen [30]. For conditional generation, we use CFG scale of 2.0 as we found it achieved the best performance across different samplers.

A.3 Neural networks

We applies efficien U-Net as our denoiser function backbone, which is designed to be memory efficient and converge fast. It reverses the order of downsampling/upsampling operations in order to improve the speed of the forward pass. For more detailed descriptions of the architecture, we encourage the readers to Appendix B from [30]. Our efficient U-Net is adapted from open source Imagen[4], For the input complex spectrogram, we use short-time Fourier transform (STFT) with window size of 510 samples and hop size of 256 samples. We use an input channel of 2 for the real and imaginary components, 128 as the base dimension and channel multipliers of [1, 2, 2, 2]. For each downsampling/upsampling block, we use 2 ResNet blocks with 2 attention heads in self-attention layer. The model has a total of 45.2 million trainable parameters. We use class label and $\log \sigma$ as efficient U-Net conditional inputs. For class conditioning, we represent class labels as one-hot encoded vectors, and then feed them through a fully-connected layer.

B Comparison of audio descriptors in copy detection



(a) AudioMAE zero-shot.

(b) AudioMAE fine-tuned.

(c) CLAP zero-shot.

(d) CLAP fine-tuned.

(e) PANNs zero-shot.

(f) PANNs fine-tuned.

Figure 3: Dataset similarity distribution of top1 matched pairs comparison of AudioMAE, CLAP and PANNs.

In this section, we compare the dataset similarity distributions from the DCASE2023 challenge task7 systems computed using the three audio encoder both with and without the fine-tune process. Fig. 3 shows the comparison results. In the left column, we present the similarity distribution of audio encoders in a zero-shot copy detection scenario (*i.e.*, without fine-tuning). The right column presents the outcomes post fine-tuning. From the figure, we can observe that before fine-tuning, the similarity scores are close to 1 especially for AudioMAE, and this suggests that the audio representations are too close for intra-class samples. This fine-tuning helps to discriminate audio samples within the same class as the similarity score distribution masses shift to the left and spread out compared to

[4]https://github.com/lucidrains/imagen-pytorch

the models without fine-tuning. Upon evaluation, the fine-tuned CLAP model exhibits the most distinctive distribution spread compared to other models. Manual listening evaluations of matched pairs from all models further confirm that the fine-tuned CLAP and PANNs consistently produce pairs that match with human auditory perception. In conclusion, we use the fine-tuned CLAP in our copy detection analysis in the main text.