Leveraging Pre-trained AudioLDM for Text to Sound Generation: A Benchmark Study

Yi Yuan[1*], Haohe Liu[1*], Jinhua Liang[2], Xubo Liu[1], Mark D. Plumbley[1], Wenwu Wang[1]

[1]Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK
[2]Centre for Digital Music (C4DM), Queen Mary University of London

***Abstract*—Deep neural networks have recently achieved breakthroughs in sound generation with text prompts. Despite their promising performance, current text-to-sound generation models face issues on small-scale datasets (e.g., overfitting), significantly limiting their performance. In this paper, we investigate the use of pre-trained AudioLDM, the state-of-the-art model for text-to-audio generation, as the backbone for sound generation. Our study demonstrates the advantages of using pre-trained models for text-to-sound generation, especially in data-scarcity scenarios. In addition, experiments show that different training strategies (e.g., training conditions) may affect the performance of AudioLDM on datasets of different scales. To facilitate future studies, we also evaluate various text-to-sound generation systems on several frequently used datasets under the same evaluation protocols, which allow fair comparisons and benchmarking of these methods on the common ground.**

***Index Terms*—Sound generation, Auditory evaluation, Benchmark system, Pre-trained networks, Transferring network**

I. Introduction

The development of deep learning models has led to a surge of interest in sound generation. Different strategies have been developed for sound generation tasks with input contents as diverse as tag [1], text [2], [3] and video [4]. Sound generation systems are useful tools for content creation in applications such as virtual reality, movies, music, and digital media [5]–[7].

Recently, significant progress has been achieved in high-fidelity text-to-sound generation [2], [8], [9]. Such sound generation systems are usually data-hungry to train. For example, AudioGen [2] collected ten different datasets for training. However, this is not viable in some real-world applications, (e.g., animal sound and environmental sound generation), where the collection and labelling work for this specific domain is a time-consuming and costly process, leading to datasets of limited scale in practice. How to overcome the data scarcity issue is a significant challenge in sound generation research. Several methods have been proposed to address this issue. Rongjie *et al.* [10] proposed to augment the quantity of data by generating novel combinations of sound events, while this approach is not suitable for scenarios with limited categories. Given these considerations, it is intuitive to ask: *can we find an effective solution to train a sound generative model with a small-scale dataset?*

Previous studies have shown that pre-trained models can improve performance in tasks with limited data [11], [12]. These models are obtained by pre-training on a massive corpus and can be fine-tuned into downstream tasks. Over the last few years, pre-training strategies have achieved enormous success across multiple fields including text [13]–[15], image [16], [17] and audio [18]–[20]. However, the effectiveness of a pre-trained model for text to sound generation is an under-explored topic. This paper takes the first step on investigating the effectiveness and feasibility of improving text-to-sound generation with pre-trained AudioLDM [9], the state-of-the-art audio generation model. Our results show that pre-trained models can achieve better performance on sound generation, especially for small-scale datasets. Moreover, empirical evidence suggests that the performance of AudioLDM on varying-size datasets can be influenced by the training conditions across different modalities.

Besides, previous sound generation studies used a range of different methodologies for evaluation, making it difficult for us to evaluate the model performance fairly. Aiming to provide an efficient and reliable reference for further sound generation research, this paper introduces a new benchmark with pre-trained AudioLDM on four commonly used audio datasets: AudioCaps [21], AudioSet [22], Urbansound8K (US8K) [23] and ESC50 [24]. Furthermore, our new benchmark contains most of the evaluation metrics applied in previous works [2], [8], [9], including Fréchet Distance (FD), Inception Score (IS) [25], Fréchet Audio Distance (FAD) [26] and Kullback-Leibler (KL) divergence. With several qualitative experiments, we also provide insights into the effectiveness of these metrics in evaluating sound generation systems. Our contributions are as follows:

- We demonstrate that transferring the pre-trained AudioLDM is beneficial for sound-generation tasks in both sample quality and training efficiency, especially for small-scale datasets.
- We benchmark the sound generation task by presenting the result of AudioLDM with multiple evaluation metrics on four commonly used sound datasets.

II. Related Work

Conditional sound generation. Kong *et al.* [27] took the first step on conditional generation by taking labels as input and generating waveforms with recurrent neural network (RNN). Then, Liu *et al.* [1] synthesised sound with latent discrete features

[*] Equal contributions

obtained from a vector quantised-variational autoencoder (VQ-VAE) [28] in the frequency domain (e.g. mel-spectrogram). By compressing the mel-spectrogram into a sequence of tokens, the model can generate sounds with long-range dependencies. Recently, remarkable progress has been made in text-to-sound generations. Diffsound [8] generated audio with a diffusion-based text encoder, a VQ-VAE-based decoder and a generative adversarial network (GAN)-based vocoder. Taking texts as input, Diffsound utilized a contrastive language image pre-training (CLIP) model [29] for text embedding before sending the condition to the encoder. To alleviate the scarcity of text-audio pairs, they proposed a text-generating strategy by combining mask tokens and sound labels. AudioGen [2] used a similar encoder-decoder structure to Diffsound [8], while generating waveform directly instead of using a vocoder. They used a transformer-based encoder to generate discrete tokens and a pre-trained Transfer Text-to-Text Transformer (T5) [30] for text embedding. To increase the quantity of sound, they mixed audio samples at various signal-to-noise ratios (SNR) and collect 10 large datasets.

Evaluation metrics for sound generation. Since subjective metrics for sound-generating systems usually require a huge amount of time and workload, various objective metrics were applied for this task. However, previous works often adopted different evaluation metrics, which makes it difficult to get intuitive comparisons. Kong *et al.* [27] used Inception Score [25] as the criterion. Liu *et al.* [1] trained a sound classifier to verify the sample quality. Diffsound [8] applied Fréchet Inception Distance (FID) [25] and Kullback-Leibler (KL) divergence to compute the sample fidelity, as well as a pre-trained audio caption transformer (ACT) to calculate a sound-caption-based loss. AudioGen [2] evaluated the result with KL divergence and Fréchet Audio Distance (FAD).

III. Proposed Method

A. AudioLDM

Our experiments are carried out with AudioLDM [9], a continuous latent diffusion-based model (LDM) for text-to-sound generations. Inspired by previous text-to-sound models, AudioLDM adopts an encoder, decoder, and vocoder architecture. By comparison, the text encoder in previous studies [2], [8], [10] is replaced by a Contrastive Language-Audio Pre-training (CLAP) model. Specifically, the CLAP consists of two encoders, a text encoder f_{text} that encodes text description y into text embedding $\boldsymbol{E}^y$ and an audio encoder f_{audio} that computes audio embedding $\boldsymbol{E}^x$ from audio samples x. CLAP trains two encoders along with two projection layers using a symmetric cross-entropy loss, resulting in an aligned audio-text latent space. By utilizing the audio embedding during training and text embedding during sampling, AudioLDM can significantly reduce the demand for text-sound pairs and enable a self-supervised paradigm of LDM optimization. The latent diffusion model contains two processes: 1) a forward process that gradually transforms the data into a standard Gaussian distribution; and 2) a reverse process that generates data from the Gaussian distribution by denoising in reverse order as the forward process. During the forward process, the continuous latent representation $\boldsymbol{z}_0$ from the mel-spectrogram is transformed into a standard Gaussian distribution $\boldsymbol{z}_n$ by gradually adding a scheduled Gaussian noise in N steps. The transition probability of each time step n is:

$$q(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}) = \mathcal{N}(\boldsymbol{z}_n; \sqrt{1-\beta_n}\boldsymbol{z}_{n-1}, \beta_n \boldsymbol{I}), \tag{1}$$

$$q(\boldsymbol{z}_n|\boldsymbol{z}_0) = \mathcal{N}(\boldsymbol{z}_n; \sqrt{\bar{\alpha}_n}\boldsymbol{z}_0, (1-\bar{\alpha}_n)\boldsymbol{\epsilon}), \tag{2}$$

where $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$ denotes the Gaussian noise with the noise level presented as $\alpha_n = 1 - \beta_n$. The latent diffusion model is trained with a re-weighted objective [9], [31], given by

$$L_n(\theta) = \mathbb{E}_{\boldsymbol{z}_0, \boldsymbol{\epsilon}, n} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_n, n, \boldsymbol{E}^x) \right\|_2^2, \tag{3}$$

where θ denotes the trainable parameters in LDM. Benefiting from the aligned audio-text space from CLAP, the reverse transition probability, $p_\theta(\boldsymbol{z}_{n-1}|\boldsymbol{z}_n, \boldsymbol{E}^y)$, can be parameterized by both $\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_n, n, \boldsymbol{E}^y)$ and $\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_n, n, \boldsymbol{E}^x)$ [9]. Reverse diffusion is then performed to generate data from a sample of standard Gaussian distribution with the reverse transition probability [31]. We will compare the difference between conditioning with $\boldsymbol{E}^x$ and $\boldsymbol{E}^y$ in our experiments.

B. Fine-tuned AudioLDM

The AudioLDM was trained originally with a combination of four large datasets, including AudioSet, AudioCaps, Freesound[1], and BBC Sound Effect library (BBC SFX)[2]. With totally 3.3M ten-second sound clips, AudioLDM is capable of generating a wide range of sound, including speech and music. However, for certain sound synthesis tasks (e.g., reproducing dog barking in the UrbanSound8K (US8K) dataset or keyboard typing in the ESC50 dataset), the quality of the generated sounds decreases considerably. In addition, AudioLDM may suffer from issues, such as overfitting or limited coverage of sound events when trained on small-scale datasets, as shown in the results in Section IV.

To further improve the performance of AudioLDM on a specific domain, we fine-tune and evaluate the pre-trained AudioLDM on three smaller datasets (i.e. US8K, ESC50, and AudioCaps). In order to conduct comprehensive comparisons on fine-tuned AudioLDM among datasets of various scales, we first benchmark this task by establishing the baseline results of the pre-trained AudioLDM on four commonly used sound datasets. During the fine-tuning process, we freeze the parameters of CLAP and the VAE encoder, leaving only the latent diffusion model for training. To study the effect of model pre-training, we also train and evaluate AudioLDM on different datasets from scratch. Besides, it was found in [9] that audio embedding is better than text embedding as the model condition information. To examine this observation in more datasets, we adopt a similar experimental setting and fine-tune AudioLDM with both text embedding and audio embedding as conditioning information. The performance comparison of these training strategies with various datasets is presented in next section.

[1] https://freesound.org/

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

IV. Evaluations and Experiments

A. Dataset

We perform experiments on four common datasets with different volumes. Two relatively small datasets used are US8K and ESC50. US8K contains 8000 sound clips with 10 classes and ESC50 has 50 classes with only 40 samples for each class. We randomly select 870 samples in US8K and 400 samples in ESC50 for evaluation. Apart from ESC50 and Urbansound8K, we also perform experiments on AudioCaps and AudioSet to further enhance our study. AudioSet is the largest dataset with 527 text labels and around 5000 hours of sound. AudioCaps contains around 47000 ten-second audio data with more diverse sound events. We establish the baseline results of AudioLDM on all four datasets, while we fine-tune it on the three smaller datasets (i.e. US8K, ESC50 and AudioCaps). Although AudioCaps is included for pre-training AudioLDM, we find that further fine-tuning on AudioCaps can improve model performance on the related evaluation set.

B. Evaluation

The evaluation is performed by comparing a set of generated audio files against a set of target audio files. For model evaluation, we follow the metrics used by AudioLDM, including Fréchet Distance (FD), Inception Score (IS), Fréchet Audio Distance (FAD), and Kullback–Leibler (KL) divergence. All four metrics are calculated based on the distance between logits value or embedding from audio classifiers. Specifically, IS calculates the entropy of label distribution, where a higher IS indicates a larger variety with vast distinction. KL divergence measures the similarity between generated and target audio by comparing the logits distributions. FAD first computes the multivariate Gaussian of two embedding values collected from a pre-trained VGGish [32]. Then, this score calculates the Fréchet distance between the Gaussian mean and variance. Both KL and FAD indicate better fidelity with lower scores. Besides the three common measurements (IS, FAD and KL) used in previous works [2], [8], [27], AudioLDM also adopt FD, which has a similar idea as FAD but uses PANNs [33], a pre-trained audio pattern recognition model, as the backbone classifier for feature embedding. To compare the effectiveness of these metrics, we perform evaluations between a set of audio files and their corrupted version by the following:

(1) *Adding noise and random masking.* We add Gaussian noise and mask content on the mel-spectrogram domain. As Figure 1 shows, all the metrics can detect with a repaid fall or rise.

(2) *Adding interference sound.* We randomly select ten irrelevant classes of audio clips and mix them directly with the target sound under the same SNR to verify whether these interfered sounds can be detected. As shown in Figure 1, we can see that KL and IS do not present significant changes with the increase of corrupted sounds. In comparison, FD and FAD can effectively detect changes with an apparent increase in scores.

(3) *Adding re-permutation order.* We testify the sensitivity of these metrics when acoustic events are placed in the wrong order. To simulate this change, the ground-truth data is composed of a group of different sound events and we randomly permute their orders. Figure 1 shows that with the increase of permuted events, only FD presents an increasing trend while other metrics stay stable with little fluctuations, indicating that only the FD score is capable on classifying sounds with order.

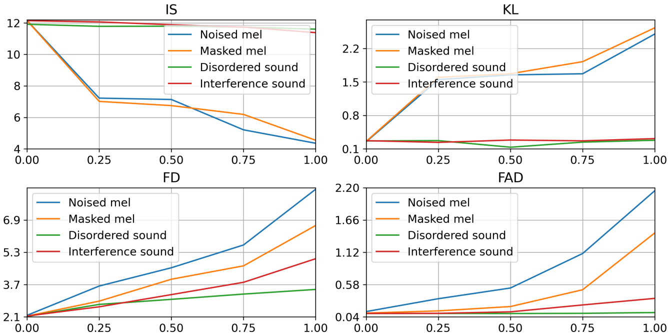

Fig. 1. The metrics are evaluated with the increase of the percentage (from 0 to 1) of pre-processed data on: 1) adding noise (random Gaussian noise); 2) masking value (one-second-sound); 3) making disordered sound events; 4) adding interfering sound events. A higher IS and a lower KL, FD, and FAD indicate better sample quality.

TABLE I
THE BASELINES OF THE PRE-TRAINED AUDIOLDM ON FOUR DATASETS

Dataset	Test Condition	FD↓	IS↑	KL↓	FAD↓
ESC50	Text	60.63	5.55	3.01	5.95
	Audio	47.46	6.68	2.08	4.81
US8K	Text	31.20	3.88	2.20	10.00
	Audio	32.79	4.04	1.44	13.74
AudioCaps	Text	23.63	6.68	2.36	4.94
	Audio	21.37	6.65	1.78	2.18
AudioSet	Text	20.30	7.56	2.34	4.26
	Audio	19.04	6.72	1.63	1.52

C. Results

Benchmark Study. As shown in Table I, we evaluate the performance of pre-trained AudioLDM[3] as our baselines for text-to-sound generations. Note that we do not perform any fine-tuning on AudioLDM in this section. Although the open-sourced version of AudioLDM is trained with audio embeddings, AudioLDM can perform sampling with either audio or text embedding. Table I shows the effect of conditions on different modalities, where AudioLDM conditioned with audio embedding performs better than text embedding in most cases. This suggests that the distribution of audio and text embedding is not completely aligned, and audio embedding is a more precise conditioning signal for sound generation, providing better sample quality.

Fine-tuning Study. Table II shows the experimental results of this fine-tuning study on smaller datasets, including ESC50, US8K and AudioCaps. In contrast to Table I, all the experiments

[3] https://github.com/haoheliu/AudioLDM

TABLE II

The comparison between different pre-training strategies. Experiments without pre-training involve building new models from scratch. Audio and text indicate whether the model is taking audio embedding or text embedding as the condition in training.

Dataset	Pre-training	Train Condition	Train Steps (K)	FD ↓	IS ↑	KL ↓	FAD ↓
ESC50	✗	Audio	240	44.75	7.44	3.31	4.02
	✗	Text	160	30.74	10.22	1.84	3.28
	✓	Audio	180	36.43	11.15	2.15	4.41
	✓	Text	80	**22.38**	**12.98**	**1.56**	**2.66**
US8K	✗	Audio	160	33.69	3.73	2.04	5.75
	✗	Text	350	28.45	**5.00**	**1.87**	**4.45**
	✓	Audio	20	31.21	3.84	2.11	7.39
	✓	Text	240	**28.44**	4.91	1.88	4.88
AudioCaps	✗	Audio	480	24.04	7.12	2.20	2.98
	✗	Text	480	24.84	6.91	2.25	2.47
	✓	Audio	80	**23.57**	7.21	**2.09**	2.98
	✓	Text	240	25.78	**7.95**	2.26	**1.67**

in this section only evaluate with text embedding as the input condition since we mainly focus on text-to-sound generation. We notice that the pre-trained AudioLDM is more advantageous than the model trained from scratch in most cases. With only 32 samples in each class, the performance of ESC50 can be significantly improved with pre-training. On US8K, the performance of pre-trained AudioLDM is slightly lower, which might be attributed to: 1) US8K is large enough for model optimization, with around 800 samples for each class; 2) US8K only contains 10 sound classes while the pre-trained AudioLDM is capable of generating sound with more diversity, which might degrade model performance on US8K evaluation set. Additionally, the pre-trained model can improve generation quality on AudioCaps, particularly on the FAD scores. We also notice that fine-tuning with text embedding on AudioCaps can further achieve a better IS score. This reason could be that the text embeddings provide weaker conditions as compared with audio embeddings, leading to results with less restriction and more diversity.

Furthermore, AudioLDM is trained in a self-supervised way using audio embedding as conditioning information because training data can be easily scaled up with this scheme. It was found earlier that taking audio embedding as the training condition was better than text embedding. However, our experiment shows this is not always the case on different datasets. As shown in Table II, results on small-scale datasets are usually better using text embedding. We believe this is because insufficient audio training data leads to sub-optimal learning of generative models, such as overfitting. This is also supported by our results, which show that training models with audio embedding achieve better performance with fewer training steps, such as 20k steps in US8K and 80k steps in AudioCaps. Conversely, texts or labels provide less detailed and diverse conditions, which can regularize the model to learn data distribution with less chance of overfitting, leading to model convergence with more training steps at the same time.

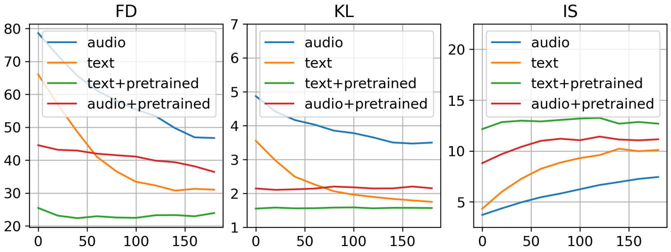

Fig. 2. The performance of AudioLDM on ESC50 as a function of thousand training steps. The four curves show the AudioLDM optimized with 1) audio embeddings; 2) text embeddings; 3) text embedding with the pre-trained model parameters; and 4) audio embedding with the pre-trained model parameters.

Figure 2 illustrates the performance of AudioLDM on different training epochs with and without pertaining and different modalities as condition information. The experiment is performed on the ESC50 dataset. We notice that 1) the pre-trained model can reach coverage quickly with text-embedding, within 20k training steps; 2) AudioLDM can achieve better performance with text-embedding on ESC50; 3) AudioLDM trained from scratch converges more slowly with a larger number of steps as compared with the pre-trained model.

V. Conclusion

We have presented a study of using pre-trained AudioLDM for text to sound generation tasks, with various settings and datasets. We have shown that the pre-trained model can improve the sample quality and reduce the training time, especially for datasets of relatively small scales. This serves as evidence for future studies on audio generation in data-scarcity scenarios. In addition, we have found that text embedding is preferred as the condition information on small-scale datasets, which alleviates overfitting during training. Finally, a new benchmark is established for text to sound generation tasks with four commonly used datasets. These baseline results can be used as benchmarks for future studies of text-to-sound generation.

VI. Acknowledgment

This research was partly supported by a research scholarship from the China Scholarship Council (CSC) No.202208060240, the British Broadcasting Corporation Research and Development (BBC R&D), Engineering and Physical Sciences Research Council (EPSRC) Grant EP/T019751/1 "AI for Sound", and a PhD scholarship from the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising.

References

[1] X. Liu, T. Iqbal, J. Zhao, Q. Huang, M. Plumbley, and W. Wang, "Conditional sound generation using neural discrete time-frequency representation learning," *IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, 2021.

[2] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "AudioGen: Textually Guided Audio Generation," in *International Conference on Learning Representations*, 2023.

[3] X. Tan, J. Chen, H. Liu, J. Cong, C. Zhang, Y. Liu, X. Wang, Y. Leng, Y. Yi, L. He *et al.*, "Naturalspeech: End-to-end text to speech synthesis with human-level quality," *arXiv preprint arXiv:2205.04421*, 2022.

[4] V. Iashin and E. Rahtu, "Taming Visually Guided Sound Generation," *British Machine Vision Conference*, 2021.

[5] X. Liu, H. Liu, Q. Kong, X. Mei, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, "Separate what you describe: Language-queried audio source separation," in *Interspeech*, 2022.

[6] S. Ghose and J. J. Prevost, "Autofoley: Artificial synthesis of synchronized sound tracks for silent videos with deep learning," *IEEE Transactions on Multimedia*, vol. 23, pp. 1895–1907, 2021.

[7] H. Liu, W. Choi, X. Liu, Q. Kong, Q. Tian, and D. Wang, "Neural Vocoder is All You Need for Speech Super-resolution," in *Interspeech*, 2022.

[8] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, "Diffsound: Discrete Diffusion Model for Text-to-sound Generation," *arXiv preprint arXiv:2207.09983*, 2022.

[9] 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.

[10] 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.

[11] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *International Conference on Machine Learning*, vol. 97, 2019, pp. 2712–2721.

[12] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning?" in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, vol. 9, 2010, pp. 201–208.

[13] Z. Guo, M. Yan, J. Qi, J. Zhou, Z. He, Z. Lin, G. Zheng, and X. Wang, "Few-Shot Table-to-Text Generation with Prompt Planning and Knowledge Memorization," *arXiv preprint arXiv:2302.04415*, 2023.

[14] X. Liu, X. Mei, Q. Huang, J. Sun, J. Zhao, H. Liu, M. D. Plumbley, V. Kilic, and W. Wang, "Leveraging pre-trained bert for audio captioning," in *European Signal Processing Conference*, 2022, pp. 1145–1149.

[15] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.

[16] B. Li, X. Wang, X. Xu, Y. Hou, Y. Feng, F. Wang, and W. Che, "Semantic-Guided Image Augmentation with Pre-trained Models," *arXiv preprint arXiv:2302.02070*, 2023.

[17] Y. Zhang, S.-C. Huang, Z. Zhou, M. P. Lungren, and S. Yeung, "Adapting pre-trained vision transformers from 2d to 3d through weight inflation improves medical image segmentation," in *Machine Learning for Health*. PMLR, 2022, pp. 391–404.

[18] A. Ghanbarzade and H. Soleimani, "Self-Supervised In-Domain Representation Learning for Remote Sensing Image Scene Classification," *arXiv preprint arXiv:2302.01793*, 2023.

[19] X. Mei, X. Liu, H. Liu, J. Sun, M. D. Plumbley, and W. Wang, "Language-based audio retrieval with pre-trained models," Tech. Rep., DCASE Challenge, 2022.

[20] X. Mei, X. Liu, J. Sun, M. D. Plumbley, and W. Wang, "On Metric Learning for Audio-Text Cross-Modal Retrieval," in *Interspeech*, 2022.

[21] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *NAACL-HLT*, 2019.

[22] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "AudioSet: An ontology and human-labeled dataset for audio events," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 776–780.

[23] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *ACM International Conference on Multimedia*, 2014, pp. 1041–1044.

[24] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the Annual ACM Conference on Multimedia*, 2015, pp. 1015–1018.

[25] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[26] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet Audio Distance: A Metric for Evaluating Music Enhancement Algorithms," in *Interspeech*, 2019.

[27] Q. Kong, Y. Xu, I. Iqbal, Y. Cao, W. Wang, and M. Plumbley, "Acoustic scene generation with conditional samplernn," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 925–929, 2019.

[28] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[29] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021, pp. 8748–8763.

[30] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[31] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Conference on Neural Information Processing Systems*, 2020.

[32] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations*, 2015.

[33] 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*, vol. 28, pp. 2880–2894, 2020.