AINUR: HARMONIZING SPEED AND QUALITY IN DEEP MUSIC GENERATION THROUGH LYRICS-AUDIO EMBEDDINGS

Giuseppe Concialdi[†‡*], *Alkis Koudounas*[‡*], *Eliana Pastor*[‡], *Barbara Di Eugenio*[†], *Elena Baralis*[‡]

[†]University of Illinois at Chicago, Chicago, Illinois, US
[‡]Politecnico di Torino, Turin, Italy

ABSTRACT

In the domain of music generation, prevailing methods focus on text-to-music tasks, predominantly relying on diffusion models. However, they fail to achieve good vocal quality in synthetic music compositions.

To tackle this critical challenge, we present Ainur, a hierarchical diffusion model that concentrates on the lyrics-to-music generation task. Through its use of multimodal Lyrics-Audio Spectrogram Pre-training (CLASP) embeddings, Ainur distinguishes itself from past approaches by specifically enhancing the vocal quality of synthetically produced music. Notably, Ainur's training and testing processes are highly efficient, requiring only a single GPU. According to experimental results, Ainur meets or exceeds the quality of other state-of-the-art models like MusicGen, MusicLM, and AudioLDM2 in both objective and subjective evaluations. Additionally, Ainur offers near real-time inference speed, which facilitate its use in practical, real-world applications.

Index Terms— deep music generation, multimodal representation, generative modeling, lyrics-to-music

1. INTRODUCTION

In recent years, deep music generation has witnessed remarkable advancements driven by the integration of cutting-edge machine learning techniques. Generating music poses substantial challenges, as it requires proficient modeling of long-range sequences, generating high-fidelity, coherent audio with the limited availability of paired audio-text data, and dealing with substantial computational resource requirements. Several models have demonstrated impressive skills in music generation from text while differing in their conditioning. Jukebox [1] specifically focused on lyrics-to-music generation. Other models, such as MusicGen [2] and MusicLM [3], aimed for a more general text-to-music generation, conditioning musical composition on arbitrary textual descriptions rather than lyrics. Nevertheless, they still struggle with producing vocals of satisfactory quality, often yielding unclear and unintelligible outputs. Furthermore, the potential of leveraging lyric content to enhance vocal coherence and overall musical output remains underexplored.

To tackle these challenges, we propose Ainur[1], a novel three-stage hierarchical diffusion model specifically tailored for the task of lyrics-to-music generation. In the first stage, we capture global musical structure and context, which are then used to create intermediate representations with greater dimensionality in the second stage. In the third stage, Ainur leverages these representations to produce high-quality audio outputs. Through this workflow, Ainur can generate 48 kHz stereo audio samples 22 seconds long in near real-time, achieving an excellent balance between producing high-quality musical content and minimizing creation time requirements.

Inspired by CLAP [4], we propose Contrastive Lyrics-Audio Spectrogram Pre-training (CLASP) embeddings to guide the generation process and enhance vocal quality. CLASP, which finetunes CLIP [5] with synced lyrics and audio spectrogram data, improves the structural consistency of the vocals. We argue that integrating lyrics to guide the generation process further enhances the quality and expressiveness of the synthesized music compositions. Our model excels in lyrics-to-music and text-to-music generation tasks by accommodating both textual descriptions and lyrics as conditioning inputs.

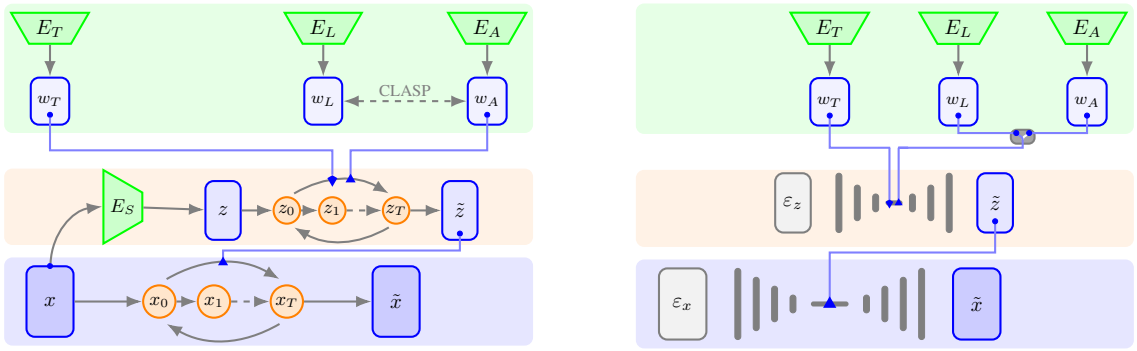
We train the model on a curated 2k-hour dataset and conduct objective and subjective evaluations, revealing Ainur's competitive performance over state-of-the-art approaches in the field. Ainur is designed to run on single, consumer-grade GPUs, bringing state-of-the-art music generation to a wider audience and mitigating the need for massive computational resources. We release the model[2] for practitioners to use.

Our contributions. We introduce Ainur, an efficient model capable of producing high-quality 22-second stereo music samples at 48 kHz in near real-time. Our model serves a dual purpose by enabling both lyrics- and text-conditioned music generation. We demonstrate the quality and adherence of the generated audio with the provided lyrical and textual prompts through objective and human evaluations.

[*] These authors contributed equally to this work.

[1] The name "Ainur" is inspired by Tolkien's *The Silmarillion*, where the Ainur are divine spirits who create the world through their music. This story is known as the Ainulindalë or "The Music of the Ainur" in Elvish language.

[2] Repository: `https://github.com/Ainur-Music/Ainur`



(a) `Ainur` training. (b) `Ainur` inference.

Fig. 1. The architecture of the proposed approach, `Ainur`, with (a) training and (b) inference workflows. ♦ represents the cross-attention operation; ▲ denotes the conditioning of the diffusion process via latent injection.

2. RELATED WORK

Text-to-audio generation has recently gathered significant attention, with several approaches tackling this task's challenges. DiffSound [6] uses CLIP [5] embeddings as input to a diffusion model that predicts mel spectrogram features. AudioGen [7] applies an encoder-decoder structure with T5 encoding text and an autoregressive transformer decoding latent audio codes. AudioLDM [8] learns a VAE latent space from spectrograms and reconstructs audio from latent codes while AudioLDM2 [9] bridges GPT-2 text encoding and latent diffusion reconstruction via AudioMAE [10] features.

More closely related to music generation, some works focus on synthesizing music conditioned from the text. Riffusion [11] fine-tunes Stable Diffusion on mel spectrograms with a paired music-text dataset. MusicLM [3] extends AudioLM's [12] autoregressive modeling to incorporate text inputs. MusicGen [2] similarly uses an autoregressive transformer conditioned on text or melody representations. Moûsai [13] uses a two-stage cascading diffusion process with a 1D U-Net architecture. Ainur enhances audio coherence using combined text-audio embeddings and mel-spectrogram techniques, distinguishing itself from Moûsai, which concentrates solely on amplitude spectrograms and text. Unlike Moûsai, Ainur (i) incorporates multimodal audio-text conditioning, (ii) features lyrics-driven vocal generation for linguistic alignment, and (iii) employs mel-spectrograms rather than amplitude spectrograms.

The only prior work explicitly targeting the lyrics-to-music task is Jukebox [1], which employs a maximum likelihood autoregressive sparse transformer trained on a discrete audio-encoded space. In contrast, Ainur focuses on generating music directly from lyrics using a hierarchy-of-experts diffusion conditioned on lyrics-audio embeddings. Inspired by Contrastive Language-Audio Pre-training (CLAP) [4], which fine-tunes CLIP [5] with language-audio data, we align synced lyrics and audio spectrograms for enhanced vocal consistency. This multi-modal approach enables Ainur to adapt to both lyrics-to-music and text-to-music generation tasks, utilizing textual and lyrical inputs for conditioning.

3. METHOD

In this section, after overviewing CLASP embeddings, we describe the architecture, the training, and inference procedures. We illustrate Ainur's hierarchical architecture in Fig. 1(a).

3.1. CLASP Embeddings

Taking inspiration from CLIP [5] and CLAP [8], we introduce the CLASP embeddings to guide the coherent lyrics generation process. Like CLIP strengthens the link between images and text, CLAP aligns natural language and audio by transforming the audio into a 2D representation using STFT and applying contrastive learning. The CLASP model learns embeddings to represent lyrics (E_L) and audio spectrograms (E_A) such that lyric-audio pairs from the same song are close in the shared embedding space while unrelated pairs are distant. This ensures coherence between textual lyrics and corresponding audio fragments. The embeddings also capture temporal and sequential relationships to align lyrics with audio over time. CLASP pre-training thus aims to fuse linguistic and acoustic domains analogously to how CLIP and CLAP integrate images/text and audio/text modalities, respectively.

The Mel Spectrograms are generated using FFTs, with 80 frequency channels, 1024 window length, and 256 hop size. The spectrogram encoder (E_A) is based on a simple Vision Transformer (ViT) architecture [14]. It operates on Mel-spectrograms extracted from the audio, using patches of size 32×32 pixels, and generates embeddings of 512 features. The lyrics encoder is a transformer model, specifically adopting the same architecture as the text encoder in CLIP. We fine-tune a publicly-available CLIP checkpoint [15] to adapt it specifically to our domain.

3.2. Ainur Architecture

Ainur integrates three modalities: text descriptions, lyrics, and audio through spectrograms. The multi-stage structure focuses on different levels of abstraction, from global patterns to local details. This allows the model to learn representations capturing overall and granular musical structures within the audio data. By hierarchically incorporating multimodal data, Ainur is able to steer music creation based on language and audio inputs at multiple levels of representation.

CLASP and Text Embeddings. In the first stage (Fig. 1(a), top), we generate a low-dimensional representation of input data (i.e., lyrics and audio) using pre-trained encoders, emphasizing global structure and contextual information. Alongside CLASP embeddings (w_L and w_A), we use natural language descriptions (e.g., genre, style, artists, and song progression) to influence music synthesis. A frozen T5 [16] encoder (E_T) processes the text description (w_T). We employ cross-attention with the resulting embeddings to guide the generation toward a specific style and part of the track.

Diffusion Prior. In the second stage (Fig. 1(a), middle), a 1D encoder compresses the spectrogram, creating a low-dimensional prior essential for real-time, high-quality audio generation. This prior, denoted as z, is a condensed representation of the original audio signal x, formed through STFT and Mel scale mapping transformations. Rather than using conventional high-dimensional audio cross-attention, this operation occurs within the latent space of an autoencoder. Consequently, the model can assimilate more specific details about the content and structure of the generated music.

Diffusion Autoencoder (DAE). In the final stage (Fig.1(a), bottom), Ainur employs the second stage's representations to craft the high-quality 48 kHz stereo audio signal, $\tilde{x}$. This precision stems from the diffusion autoencoder[17], combining the U-Net architecture's 1D design and the integration of the prior, $\tilde{z}$, from the previous stage. The U-Net's design facilitates efficient downsampling and upsampling. By merging the prior with noise during the reverse denoising process, Ainur functions both as a decoder and a vocoder, enhancing its generation capabilities while maintaining near real-time efficiency. We employ a pre-trained DAE from the ArchiSound [17] library, compressing input signals to a 32x latent representation. This compression effectively balances dimensionality reduction with data representation quality.

Loss Functions. We optimize the cosine similarity between audio and lyric embeddings during contrastive pre-training to align their representations. In the prior and autoencoding diffusion stages, we use the v-objective loss function:

$$\mathcal{L} = \mathbb{E}_{t\sim[0,1],\sigma_t}\left[||\widehat{\boldsymbol{v}}_{\sigma_t} - \boldsymbol{v}_{\sigma_t}||_2^2\right] \quad (1)$$

Here the traditional diffusion equation is modified to involve $\widehat{\boldsymbol{v}}\sigma_t$ and $\boldsymbol{v}\sigma_t$, representing generated and original velocities, respectively, computed as the derivative of data points ($\boldsymbol{x}_{\sigma_t}$) concerning noise level changes (σ_t). This measures how data points change with slight adjustments in noise level [18].

Inference. We outline the inference process in Fig.1(b). Ainur's main task is to generate music from lyrics. Unlike its training phase, Ainur operates without any extra input, leveraging only lyrics and, optionally, text descriptors and audio. These inputs create a latent representation $\tilde{z}$ from Gaussian noise ε_z, which is then processed by the decoding U-Net, along with high-dimensional noise ε_x, to produce music samples that reflect the learned style. The effectiveness of this process depends on the number of steps to convert noise to music and the specific embedding and inputs used.

4. EXPERIMENTAL SETUP

Dataset. To train and evaluate Ainur, we use an internal dataset comprising over 31,000 not-licensed music tracks, amounting to approximately 2k hours of high-quality 48 kHz stereo music. This dataset includes metadata about authors, genres, and time-synced lyrics embedded within the songs. The training samples are segmented into 22-second-long pieces, increasing the exposure to various training samples during each epoch. For evaluation, we set aside approximately 1,000 samples, which include validation and test sets encompassing 10 and 50 hours of music data, respectively.

Training. We extract lyrics and textual descriptions from metadata and normalize them. Synchronized audio and lyrics segments are subject to random cropping, resulting in a window of approximately 22 seconds. We combine information related to the artist, genre, style, and sequence to construct the text descriptors. Audio data remains unaltered and unscaled to enable the model to capture and learn variations in loudness levels and pitch, characteristic of specific music genres.

We train Ainur on a single Nvidia V100 GPU. CLASP training spans 120,000 iterations. The prior undergoes training for over 1 million iterations, while the Mel spectrogram-based DAE is pre-trained. We provide detailed information about the hyperparameter setup, the training procedure, and the hardware used in the official project repository[2].

Metrics. We assess Ainur's performance on four aspects: the inference time, the quality of the synthesized audio, how well it aligns with the provided text description, and the overall vocals quality.

Inference Time. This metric evaluates the time required to synthesize a single music sample and reflects the model's applicability in real-world scenarios.

[3]While Jukebox can generate samples up to 22 seconds, it requires significant computational resources and time. We could only generate one second when tested on a typical consumer-grade GPU. Hence, the results we present may not fully encapsulate Jukebox capabilities.

[4]MusicLM is capable of producing 22-second music samples, but it requires approximately 5 minutes, making our extensive objective evaluation impractical. Still, we evaluated 22-second audios for the subjective one.

Table 1. Objective evaluation. The best results for 22-second samples are in bold, best overall are underlined. The rate column indicates the sampling rate in KHz and the channels, 1 for mono and 2 for stereo.

MODEL	RATE [kHz]	LENGTH [s]	#PARAMETERS [M]	INFERENCE [s] ↓	$\text{FAD}_{\text{VGGISH}}$ ↓	$\text{FAD}_{\text{YAMNET}}$ ↓	$\text{FAD}_{\text{TRILL}}$ ↓
AINUR	48@2	22	910	14.5	**8.38**	**20.70**	0.66
AINUR W/OUT CLASP	48@2	22	910	14.7	8.40	20.86	0.64
AUDIOLDM	16@1	22	181	**2.20**	15.5	784.2	**0.52**
AUDIOLDM 2	16@1	22	1100	20.8	8.67	23.92	**0.52**
MUSICGEN	16@1	22	300	81.3	14.4	53.04	0.66
JUKEBOX[3]	16@1	1	1000	538	20.4	178.1	1.59
MUSICLM[4]	16@1	5	1890	153	15.0	61.58	<u>0.47</u>
RIFFUSION	44.1@1	5	890	6.90	<u>5.24</u>	<u>15.96</u>	0.67

Table 2. Subjective evaluation. Best results in bold.

MODEL	INFERENCE [s] ↓	WINS ↑	REL↑	VCL↑
AINUR	**14.50**	**24**	**3.12**	**1.94**
AUDIOLDM 2	20.80	13	3.04	1.60
MUSICGEN	81.30	12	3.04	1.36
MUSICLM	290.2	1	2.86	1.16

Fréchet Audio Distance (FAD). The FAD [19] metric determines the dissimilarity between two probability distributions, one derived from generated audio while the other from reference audio samples. FAD provides an objective measure of audio quality, demonstrating a strong correlation with human perception. Models that yield low FAD scores are generally expected to produce realistic and plausible audio. We report the FAD based on three audio embedding models: (1) VGGish [20], (2) Trill [21], and (3) YAMNet [22].

Subjective Evaluation. We conducted a user study to evaluate the audio samples using three criteria: Relevance to Text Input (REL), Vocal Quality (VCL), and number of wins (Wins). In assessing REL, participants were asked to rate the alignment between audio and provided text input on a scale from 1 (low) to 5 (high). For VCL, they were asked to judge the vocal quality on a 1 to 5 scale (best). We then asked participants to select the best sample from four options for the same prompt. We report the number of wins for each approach.

5. RESULTS AND DISCUSSION

We compare Ainur's performance against several state-of-the-art approaches, including AudioLDM [8], AudioLDM2 [9], MusicGen [2], MusicLM [3], Jukebox [1] and Riffusion [11]. Table 1 shows the results of the objective evaluation. We treated all generated samples as mono 16kHz audios to ensure a fair comparison. Ainur stands out as the sole model able to generate stereo high-quality music samples. Within the models capable of producing 22-second-long samples, Ainur outperforms all other approaches on $\text{FAD}_{\text{VGGish}}$ and $\text{FAD}_{\text{YAMNET}}$ metrics while remaining competitive on $\text{FAD}_{\text{Trill}}$. Ainur also exhibits competitive inference times versus all models, slower only than AudioLDM. Jukebox, while assessed on only one second of audio due to computational constraints, underperforms. On the other hand, Riffusion achieves the best overall scores but generates only 5-second clips. We also examine the performance of our approach if conditioned only with text, without lyrics (second row in Table 1). The results show that text-only guidance achieves lower performance on $\text{FAD}_{\text{VGGish}}$ and $\text{FAD}_{\text{YAMNET}}$, and comparable outcomes for $\text{FAD}_{\text{Trill}}$.
We also conduct a subjective evaluation of the approaches capable of producing 22 seconds of audio to assess the quality of the vocals and the coherence of the generated music to the textual conditioning prompts. A total of 200 generated samples were evaluated by 10 human subjects, with preferences reported as the number of wins, relevance, and vocal quality in terms of MOS. Table 2 highlights the results, showing that our approach outperforms all other methods. Its dominance in the vocal quality MOS rating highlights the benefits of training specifically for the lyrics-to-music generation. This confirms Ainur leads in human perceptions for applications requiring long, high-quality audio.

6. CONCLUSION

We propose Ainur, a novel music generation model that tackles the challenges of lyrics-to-music generation. Our results demonstrate the high vocal quality and best adherence to the text input of our generated music samples and their highest user preference among state-of-the-art techniques.

7. ACKNOWLEDGMENTS

This work is partially supported by FAIR - Future Artificial Intelligence Research (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013) and the spoke "FutureHPC & BigData" of the ICSC - Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing, funded by the European Union - NextGenerationEU. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible.

8. REFERENCES

[1] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[2] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez, "Simple and controllable music generation," *arXiv preprint arXiv:2306.05284*, 2023.

[3] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al., "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[4] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang, "Clap learning audio concepts from natural language supervision," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[6] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu, "Diffsound: Discrete diffusion model for text-to-sound generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1720–1733, 2023.

[7] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi, "Audiogen: Textually guided audio generation," in *The Eleventh International Conference on Learning Representations*, 2022.

[8] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley, "AudioLDM: Text-to-audio generation with latent diffusion models," in *Proceedings of the 40th International Conference on Machine Learning*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, Eds. 23–29 Jul 2023, vol. 202 of *Proceedings of Machine Learning Research*, pp. 21450–21474, PMLR.

[9] Haohe Liu, Qiao Tian, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley, "Audioldm 2: Learning holistic audio generation with self-supervised pretraining," *arXiv e-prints*, pp. arXiv–2308, 2023.

[10] Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer, "Masked autoencoders that listen," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28708–28720, 2022.

[11] Seth Forsgren and Hayk Martiros, "Riffusion," [Online] Available: `https://github.com/riffusion/riffusion`, Dec. 2022.

[12] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al., "Audiolm: a language modeling approach to audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[13] Flavio Schneider, Zhijing Jin, and Bernhard Schölkopf, "Moûsai: Text-to-Music Generation with Long-Context Latent Diffusion," *arXiv e-prints*, p. arXiv:2301.11757, Jan. 2023.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, Oct. 2020, pp. 38–45, Association for Computational Linguistics.

[16] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter 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.

[17] Flavio Schneider, "ArchiSound: Audio Generation with Diffusion," *arXiv e-prints*, p. arXiv:2301.13267, Jan. 2023.

[18] Tim Salimans and Jonathan Ho, "Progressive distillation for fast sampling of diffusion models," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. 2022, OpenReview.net.

[19] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms.," 2019.

[20] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al., "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[21] J. Shor, A. Jansen, R. Maor, O. Lang, O. Tuval, F. de Chaumont Quitry, M. Tagliasacchi, I. Shavitt, D. Emanuel, and Y. Haviv, "Towards learning a universal nonsemantic representation of speech," in *INTERSPEECH*, 2020.

[22] M. Plakal and D. Ellis, "Yamnet," [Online] Available: `https://github.com/tensorflow/models/tree/master/research/audioset/yamnet`, Jan 2020.