Fine-Tuning and Evaluation of AudioLDM2 for Text-Conditioned Music Generation

Niranjan Kumar Kishore Kumar[a]

[a]Yeshiva University

Abstract. This research aims to enhance text-to-music generation by fine-tuning the AudioLDM2 and developing a custom model designed to outperform state-of-the-art(SOTA) benchmarks. Through meticulous fine-tuning on the MusicNet audio dataset, we address issues common in audio generation, including unwanted noise and limited dynamic range, striving to improve musical depth and clarity. While our fine-tuned model shows improvement in generating music from text prompts, it still exhibits some noise, and its performance remains below that of SOTA models. This ongoing work aims to further refine the model and continue development of the custom architecture, addressing current limitations to achieve a benchmark-comparable solution. The code is publicly available on GitHub for reproducibility and continued enhancement: GitHub repository

1 Introduction

Text-based generation has expanded significantly, encompassing the creation of images, videos, and audios from textual prompts. Text-to-image generation, for example, can produce both realistic and stylized images based on textual inputs, a capability that has proven valuable in diverse applications such as content creation and graphic design. Meanwhile, text-to-audio generation, though relatively new, is rapidly evolving, enabling the creation of sound effects, ambient audio, and background music tailored for movies, games, and other multimedia applications. Among these advancements, diffusion models[4] have demonstrated exceptional performance, particularly in their cascading forms, across multiple modalities, including music [5].These models enable hierarchical generation, where audio is encoded into compressed representations and subsequently expanded into high-fidelity output, supporting detailed control over output quality and length. Notably, models like Noise2Music[5] have successfully generated long, high-quality music compositions, achieving audio synthesis at 48kHz or higher.

Text-to-music generation, an emerging domain within deep generative modeling, has made significant strides due to advancements in both language models and diffusion models. This task involves generating high-quality, musically coherent compositions from text prompts, encompassing attributes such as genre, mood, tempo, and instrumentation, which can expand creative possibilities for musicians and enthusiasts[12].

A major challenge in this field is the availability and diversity of paired music-text datasets. Text-to-music systems require extensive data to capture musical nuances like harmony, melody, and rhythm. Recent efforts have addressed this challenge through creative data augmentation techniques, as seen in the MusicBench dataset, which includes augmented captions and control parameters that improve the alignment of generated music with text prompts [10].

The capacity to generate music that reflects nuanced textual prompts has become increasingly feasible due to the integration of powerful language modeling techniques with diffusion-based architectures. This paper builds upon such advancements, presenting a fine-tuned version of AudioLDM2[8], a model specifically refined to improve music generation from text prompts by leveraging the MusicNet dataset for domain-specific training. Additionally, in parallel to this fine-tuning, we are developing a custom model that incorporates knowledge distillation techniques, allowing us to transfer knowledge from a larger, more powerful teacher model to a smaller, more efficient student model. This distillation approach aims to retain the high-level musical nuances of the teacher model while improving computational efficiency in the student model.

Our experimental findings reveal that while the fine-tuned AudioLDM2 shows promising results, it does not yet achieve competitive performance across all benchmarks on the MusicNet dataset. However, the inclusion of knowledge distillation in our custom model points toward potential improvements in interpretability and generation quality, making it a valuable approach for producing diverse, high-quality music aligned with textual inputs.

Overall, our contributions are as follows:

1. We introduce a fine-tuned version of AudioLDM2 trained on the MusicNet dataset, optimized for text-to-music generation.

2. We develop a custom model using knowledge distillation to retain high-fidelity musical characteristics from a teacher model, improving both efficiency and quality in a more compact student model.

3. We conduct comprehensive evaluations, identifying key areas for enhancing the model's musical fidelity, diversity, and alignment with text prompts.

4.We outline future research directions, including further application of knowledge distillation for computationally efficient, high-quality music generation.

2 Related Work

Text-to-Audio and Text-to-Music Generation

Text-based generative models have made significant progress across multiple modalities, including text-to-image, text-to-video, and text-to-audio generation. Text-to-audio generation, though a relatively recent field, has garnered attention due to its applications in producing sound effects, background music, and ambient audio for multimedia contexts. However, generating coherent music from text is a complex task, as it requires a deep understanding of musical elements such as melody, harmony, tempo, and rhythm. Early models like AudioLM [1] and Noise2Music [5] leveraged language models and diffusion techniques to achieve high fidelity and alignment between generated audio and textual prompts.

Diffusion Models for Text-to-Music Generation

Diffusion models have demonstrated exceptional performance in text-to-music generation, especially when employed in cascading forms for hierarchical synthesis. Noise2Music [5] uses a two-stage diffusion approach, where an initial generator produces an intermediate representation conditioned on text, followed by a cascader that converts it into high-fidelity music. This model achieves accurate text alignment, capturing genre, mood, tempo, and instrumentation. Moûsai [12] extends this approach, implementing a cascading latent diffusion model capable of generating high-quality, extended music tracks at 48kHz, enabling real-time inference on a single GPU.

Other models such as MusicLDM [2] introduce unique diffusion-based methods to overcome data limitations and ensure novelty in generated outputs. By incorporating beat-synchronous mixup strategies, MusicLDM [2] addresses the risk of data overfitting and potential plagiarism, generating diverse music within the "convex hull" of training samples. Ernie-Music [12] focuses on text-to-waveform generation by directly conditioning the model on textual prompts, enhancing text-music relevance through weakly supervised web-sourced data.

Knowledge Distillation and Efficiency in Music Generation Knowledge distillation has proven effective for optimizing computational efficiency in music generation models. MeLoDy [13] exemplifies this approach, where a smaller student model distills knowledge from a larger teacher model. This dual-path diffusion method handles coarse and fine audio details separately, significantly reducing the number of diffusion steps required for high-quality music, thereby supporting real-time sampling and efficient resource utilization. JEN-1 [7] combines both autoregressive and non-autoregressive training for flexible text-guided music generation, achieving high fidelity and computational efficiency.

Controllability in Music Generation Controllability over specific musical attributes has become a key focus in recent works. Mustango [10] introduces MuNet, a music-domain-specific UNet module, allowing users to specify musical elements such as chords, beats, and tempo. This model expands the control over musical features beyond general text prompts, enabling refined adjustments to generated outputs. MusiConGen [6] addresses temporal control through rhythm and chord integration, generating music that adheres to user-defined beat and harmonic conditions, extending the capabilities of transformer-based models for music generation.

Data Augmentation for Enhanced Diversity and Novelty Given the limited availability of paired music-text datasets, data augmentation techniques are essential for improving model generalization and novelty. MusicLDM [2] incorporates beat-synchronous audio and latent mixup strategies, which encourage the model to interpolate between different musical samples, resulting in more diverse and original outputs. MusicRL [3] integrates Reinforcement Learning from Human Feedback (RLHF) to optimize text-music alignment and quality, creating a dataset of user preferences that informs model adjustments to align generated music with subjective preferences.

Symbolic Music Generation and Editing Symbolic music generation has also gained traction, offering enhanced flexibility for users to control and edit specific musical elements. MuseCoco [9] focuses on generating symbolic music from text descriptions, allowing users to manipulate musical attributes at a high level, such as chord structure, rhythm, and melody. This approach simplifies the editing process for end-users, facilitating greater customization. StemGen [11] takes an alternative approach, focusing on a model that listens to and responds to musical context, generating cohesive, multi-layered tracks that align with user-specified conditions.

Real-Time Feedback and Human-Preference Alignment Several recent models incorporate mechanisms to align generated music with user feedback and human preferences. MusicRL [3] exemplifies this by using reinforcement learning to refine model outputs based on a large dataset of user evaluations, ensuring that the generated music aligns with human tastes in terms of text adherence and quality. This approach has inspired similar efforts in other models to involve continuous user feedback, which allows for iterative refinement and real-time adaptation of generated music.

Advanced Text-to-Music Generation Models Newer models continue to push the boundaries of text-to-music generation by introducing novel architectures and methodologies. Seed-Music provides an all-in-one framework for high-quality music generation and editing, offering fine-grained style control over generated tracks, including vocal timbres, instrumentation, and post-production adjustments. MusicFlow combines flow matching and masked prediction, creating a zero-shot model capable of music infilling and continuation with high text coherence. MusicMagus focuses on music editing, enabling users to alter specific attributes like genre and mood without changing other musical elements, enhancing flexibility for users.

3 Method

As shown in Figure 1, the architecture consists of multiple stages to process text and audio.

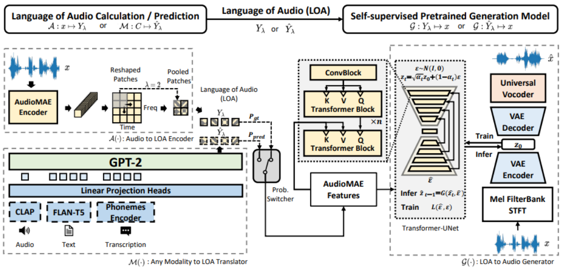

Language of Audio Calculation / Prediction
$\mathcal{A}: x \mapsto Y_\lambda$ or $\mathcal{M}: C \mapsto \hat{Y}_\lambda$
Language of Audio (LOA)
Y_λ or $\hat{Y}_\lambda$
Self-supervised Pretrained Generation Model
$\mathcal{G}: Y_\lambda \mapsto x$ or $\mathcal{G}: \hat{Y}_\lambda \mapsto x$
x
AudioMAE Encoder
Reshaped Patches
Pooled Patches
$\lambda = 2$
Freq
Time
Language of Audio (LOA)
Y_λ
$\hat{Y}_\lambda$
P_{gt}
P_{pred}
$\mathcal{A}(\cdot)$: Audio to LOA Encoder
GPT-2
Linear Projection Heads
CLAP
FLAN-T5
Phonemes Encoder
Audio
Text
Transcription
$\mathcal{M}(\cdot)$: Any Modality to LOA Translator
Prob. Switcher
ConvBlock
K V Q
Transformer Block
×n
K V Q
Transformer Block
AudioMAE Features
$\varepsilon \sim N(I, 0)$
$z_t = \sqrt{\alpha_t} z_0 + (1-\alpha_t)\varepsilon$
$\hat{\varepsilon}$
Infer $\hat{z}_{t-1} = G(\hat{z}_t, \hat{\varepsilon})$
Train $L(\hat{\varepsilon}, \varepsilon)$
Transformer-UNet
$\hat{x}$
Universal Vocoder
VAE Decoder
Train
z_0
Infer
VAE Encoder
Mel FilterBank STFT
x
$\mathcal{G}(\cdot)$: LOA to Audio Generator

Figure 1: Overview of the Text-to-Music Generation Architecture. This architecture includes a text encoder, a two-stage diffusion model, and a conditioned UNet that generates audio aligned with text prompts.Image adapted from [8].

3.1 Overview

In this work, we present a finetuned model of text-to-music generation model built upon AudioLDM 2 architecture. The model generates highly fidelity music clips conditioned on text prompts by leveraging a two-stage diffusion process and cross-attention mechanism. The AudioLDM 2 architecture, shown in Figure 1(adapted from [8]), consists of several modules that work in tandem to process both textual and audio data. Specifically, it utilizes a Language of Audio(LOA) framework to translate input features into the audio generation space, supported by a series of components for encoding, translating, and generating high-quality audio.

The architecture comprises three main components:

1. Language of Audio Calculation/Prediction (LOA)
2. Language of Audio (LOA) for intermediate representation
3. Self-supervised Pretrained Generation Model for final audio output

Each of these components is carefully designed to capture different aspects of audio and text information, enabling effective translation from textual descriptions to generated audio.

3.2 Architecture

3.2.1 Language of Audio Calculation/Prediction

The first stage in the AudioLDM 2 architecture focuses on transforming audio and text data into a unified representation, known as the Language of Audio (LOA), which serves as an intermediate layer for audio generation.

Audio Input and Embedding

1. Raw audio signals are processed through an AudioMAE (Masked Autoencoder for Audio) Encoder. The encoder divides the audio into smaller segments, or patches, each encapsulating specific frequency and time characteristics of the audio.
2. These patches are subsequently reshaped and pooled into a compact feature space, referred to as the LOA. This space captures essential attributes of the audio in a reduced form.

Text Input and Conditioning

1. For textual input, the system employs a GPT-2 model, which is pre-trained to handle text prompts. The text encoder processes the prompts and transforms them into discrete tokens, which are projected onto the LOA feature space.
2. Linear projection heads align and combine the inputs from multiple modalities (audio, text, transcription). This allows the system to integrate information from various sources and encode them into the LOA representation, which serves as a common language for both audio and text features.

Any Modality to LOA Translator

1. This component is represented by $M(\cdot)$, which translates input from any modality (e.g., audio, text) into the LOA. This enables the model to flexibly handle diverse inputs, whether they are audio samples, textual descriptions, or phoneme transcriptions, and produce a coherent LOA feature set.

3.2.2 Language of Audio (LOA) Intermediate Representation

The LOA representation serves as an essential bridge between the input features (from both audio and text) and the final audio generation model. This intermediate representation is pivotal for ensuring that the generated audio remains semantically aligned with the text prompt and is musically coherent.

Cross-Attention and Conditioning

1. The LOA representation is generated by the AudioMAE Encoder and conditioned using a probabilistic switcher, which alternates between ground-truth LOA features (P_{gt}) and predicted LOA features (P_{pred}). This switcher helps the model balance learning from both real and predicted data, improving its robustness in generating coherent outputs.
2. A cross-attention mechanism is employed, where the model learns to align the text-based conditioning with audio features. This helps the model understand subtle attributes specified in the text prompt, such as genre, mood, or instrumentation.

Transformer Blocks

1. A series of Transformer blocks process the LOA features, utilizing self-attention mechanisms to capture complex dependencies between audio and text representations. This processing allows the model to create a holistic representation that incorporates both audio patterns and the semantic nuances of the text.
2. The Transformer architecture is crucial for enabling long-range dependencies, which are necessary to generate music that unfolds in a structured and musically coherent manner, respecting rhythmic patterns, melody progressions, and harmonic structures.

3.2.3 Self-supervised Pretrained Generation Model

The final component is the Self-Supervised Pretrained Generation Model, which utilizes the refined LOA features to synthesize audio outputs. This model comprises a Transformer-UNet structure for diffusion-based audio generation, a Variational Autoencoder (VAE) for compact representation, and a Universal Vocoder for final audio synthesis.

Latent Diffusion Model with Transformer-UNet

1. The LOA features are passed through a latent diffusion model, implemented using a Transformer-UNet. This model refines the LOA representation through multiple layers of diffusion steps.
2. At each diffusion step, noise is progressively added and removed from the audio representation. The diffusion process is guided by the LOA conditioning, ensuring that the resulting audio aligns with the input text prompt.
3. The Transformer-UNet structure enables the model to handle complex temporal dependencies and produce high-fidelity audio outputs. The diffusion steps contribute to enhancing the richness and clarity of the generated music.

VAE for Compression and Expansion

1. The VAE component compresses the audio into a latent space during training and decompresses it for audio generation. This allows for efficient storage and manipulation of audio features, helping the model generate consistent and high-quality audio from complex input representations.
2. The VAE Decoder converts the compressed LOA features back into audio, while the Universal Vocoder further refines this output by synthesizing the audio waveform. This process ensures that the final output has the desired tonal and rhythmic qualities.

Universal Vocoder

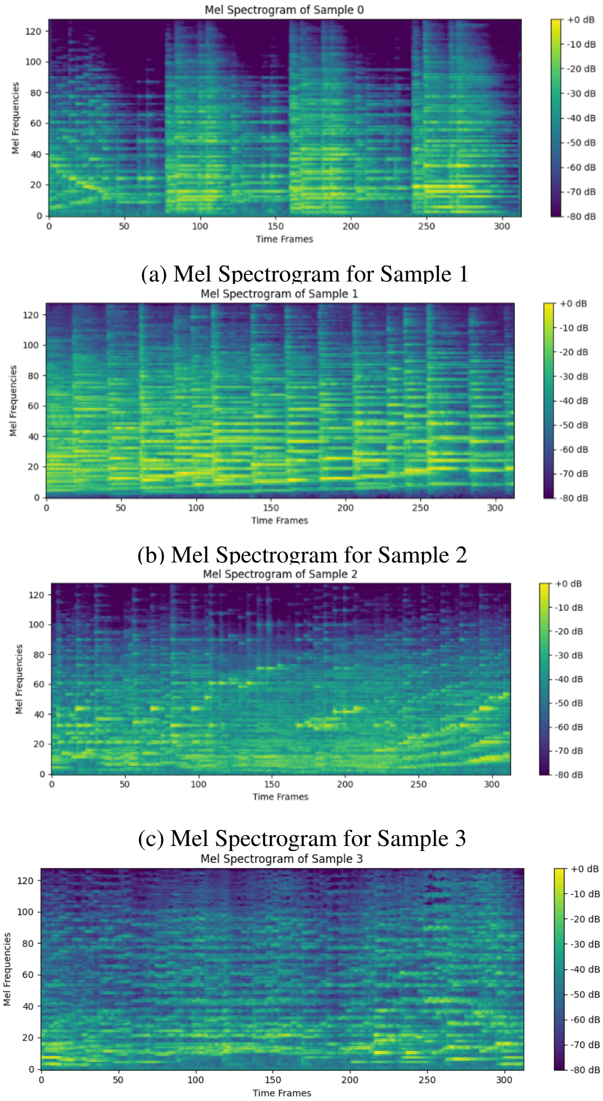
1. The final step in the generation pipeline involves a Universal Vocoder, which reconstructs the waveform from the processed LOA features. The vocoder synthesizes the high-resolution audio waveform, making it perceptually smooth and musically coherent.
2. Using the Mel FilterBank and Short-Time Fourier Transform (STFT), the Universal Vocoder ensures that the final output has high fidelity and closely matches the quality of natural audio.

3.3 Fine-Tuning

To adapt the AudioLDM2 model for our specific dataset, the MusicNet dataset, we implemented a fine-tuning process that optimizes the model's performance in generating musically coherent and contextually relevant outputs based on text prompts. The steps undertaken in the fine-tuning process are outlined below:

3.3.1 Data Preparation

- The MusicNet dataset was processed into Mel-spectrogram representations (see Figure 2) to ensure compatibility with the model's input requirements. Each audio file was resampled to a consistent sample rate, segmented into shorter clips if necessary, and converted into Mel-spectrograms of a fixed size.
- This step ensures a consistent format for all audio files, enhancing processing efficiency and compatibility with the model.



(a) Mel Spectrogram for Sample 1

(b) Mel Spectrogram for Sample 2

(c) Mel Spectrogram for Sample 3

(d) Mel Spectrogram for Sample 4

Figure 2: Mel Spectrograms for the Training Audio Samples: This figure illustrates the Mel spectrogram representations of four distinct audio samples from the training dataset. Each spectrogram provides a visual representation of the frequency components over time, with color intensity indicating the magnitude in decibels. These Mel spectrograms serve as key input features for the AudioLDM2 model during the training phase, allowing the model to capture time-frequency characteristics essential for generating high-quality, semantically aligned audio outputs. The spectrograms showcase the diversity in acoustic patterns across samples, which is critical for the model to learn a robust mapping from text prompts to audio outputs.

- Textual prompts were derived from metadata, containing information such as the composer, genre, and instrumental arrangement. These descriptions served as conditioning information for the model, guiding it to generate audio that aligns with these characteristics.
- An example prompt used in this study is as follows:

```
An evocative passage from Beethoven's
Piano Sonata No 29 in B-flat major
(OP106), particularly the movement: 1.
Allegro. This piece, performed by a Solo
Piano, exudes a rich Beethoven-esque
style with an approximate duration of
13 minutes. The ensemble's performance
reveals a profound interplay of
harmonies and thematic development.
A note (MIDI note 94) gracefully
played by Acoustic Grand Piano, adding
texture and emotion to the passage.
The note resonates from 31309790.00 to
31325150.00 seconds, evoking a sense of
movement and dynamic contrast.
```

3.3.2 Preprocessing Pipeline

The preprocessing pipeline was carefully designed to ensure the consistency and quality of the audio-text pairs used for fine-tuning. This pipeline includes several key steps:

- **Mel-Spectrogram Generation**: Each raw audio waveform is transformed into a Mel-spectrogram representation using `torchaudio.transforms.MelSpectrogram`. This transformation captures essential frequency characteristics of the audio, enabling the model to focus on relevant auditory features that serve as inputs for the generation process.

  Waveform Processing: Each waveform undergoes multiple preprocessing operations to ensure uniformity and quality:

  – *Resampling*: Each audio sample is resampled to a standard rate of 22,050 Hz to maintain consistency across the dataset.

  – *Mono Conversion*: Stereo audio files are converted to mono by averaging the channels, simplifying the data representation.

  – *Amplitude Normalization*: The waveform amplitude is normalized to prevent extreme values that could distort the Mel-spectrogram representation.

  – *Silence Removal*: Silent segments are removed using `librosa.effects.split`, which discards low-amplitude parts below a specified decibel threshold. This step reduces unnecessary data, focusing the model on the informative parts of the audio.

  – *Data Augmentation* (optional): Depending on configuration, data augmentation techniques such as time-stretching, pitch-shifting, or noise injection are applied. These augmentations provide varied input representations, enhancing the model's robustness and generalization.

- **Text Prompt Generation**: Each audio sample is paired with a corresponding textual prompt, which is constructed from the note and composition metadata. This prompt serves as conditioning information, guiding the model to generate audio that aligns with the textual description.
- **Fixed-Length Processing**: Following augmentation, each waveform is either padded or trimmed to a fixed length of 163,840 samples. This fixed length ensures that all inputs are of uniform size, facilitating consistent batch processing and stable training.

3.3.3 Model Initialization

- The model was initialized using pre-trained parameters from the original AudioLDM2 configuration, leveraging the foundational knowledge embedded in the pre-trained model for audio generation. This pre-trained base provides the model with a head start in understanding fundamental audio characteristics, facilitating adaptation to the MusicNet dataset with minimal adjustments.
- To enhance the model's compatibility with the MusicNet dataset, additional layers were introduced in the fully connected (FC) layer of the feature extractor. Specifically, we added extra layers with ReLU activations to deepen the network and improve its capacity for fine-tuning. This extended FC layer is designed to capture finer details in the Mel-spectrogram representations of the dataset, which exhibit unique characteristics in terms of genre, instrument variety, and temporal patterns.
- A fixed length of 163,840 samples was maintained across the generated and target audio waveforms. This length ensures consistency in input size, which is critical for stable batch processing and helps avoid errors during training. During model initialization, both generated audio and target labels undergo a padding or trimming process to meet this fixed length requirement.
- By starting with pre-trained parameters, the model could retain valuable features from the original AudioLDM2, which promotes faster convergence and reduces the risk of overfitting. This initialization process allows the model to preserve general audio characteristics learned from its pre-trained state while focusing on the specific attributes of the MusicNet dataset, thus enabling effective adaptation to the new domain.
- Additionally, the `AudioFeatureExtractor` module, which uses Mel-spectrogram transformations, was fine-tuned to match the specific audio profiles in the MusicNet dataset. This module ensures that each audio waveform is consistently represented in a high-dimensional Mel-spectrogram space, allowing the model to exploit these rich features during the generation process.

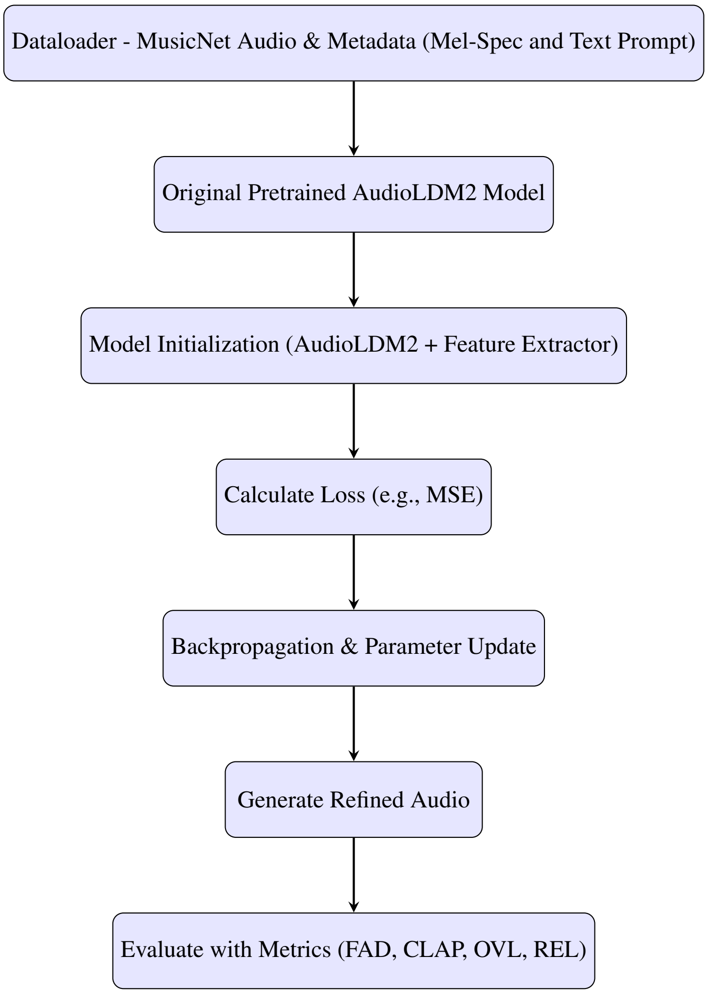

Dataloader - MusicNet Audio & Metadata (Mel-Spec and Text Prompt)
Original Pretrained AudioLDM2 Model
Model Initialization (AudioLDM2 + Feature Extractor)
Calculate Loss (e.g., MSE)
Backpropagation & Parameter Update
Generate Refined Audio
Evaluate with Metrics (FAD, CLAP, OVL, REL)

Figure 3: Fine-Tuning Workflow for AudioLDM2 Model: This flowchart illustrates the fine-tuning process, starting with loading the MusicNet audio files and associated metadata, including Mel-Spectrograms and text prompts. The original pretrained AudioLDM2 model is initialized with a feature extractor. During training, the model iteratively calculates loss (e.g., MSE), performs backpropagation, and updates parameters to improve audio generation quality. The refined audio output is then evaluated using metrics such as FAD, CLAP, OVL, and REL, assessing its alignment with the input prompt and overall quality.

3.3.4 Training Setup

- The training setup was configured with parameters tailored for effective fine-tuning on the MusicNet dataset. The batch size was set to 16 for both training and evaluation, balancing memory efficiency and model convergence speed.
- A learning rate of 5e-5 was chosen to allow for gradual adjustments to the model's weights, ensuring it could adapt to the specific characteristics of the MusicNet dataset without overfitting. Additionally, weight decay was applied at a rate of 0.01 to help regularize the model and prevent overfitting.
- The Adam optimizer was selected for its efficient gradient updates, which support smooth optimization across iterations. Mixed precision (FP16) training was enabled to reduce memory usage and accelerate training on compatible hardware.
- Evaluation and model checkpointing were conducted at the end of each epoch. The best model was determined based on the lowest evaluation loss (eval_loss), with the model configured to reload the best checkpoint at the end of training. This setup ensured that the final model reflected the optimal performance across all epochs.
- To monitor progress, logging was set to occur every 10 steps, providing detailed insights into training dynamics. Evaluation metrics, particularly Mean Squared Error (MSE), were tracked to assess the model's alignment with target audio features, ensuring a steady performance improvement across epochs.

3.3.5 Fine-Tuning the Model

- The fine-tuning process was conducted over 30 epochs on the MusicNet dataset, iteratively updating the model's weights to minimize the difference between the generated outputs and target audio features. This prolonged training allowed the model to refine its parameters effectively, gradually enhancing the alignment of generated audio with the conditioning text prompts.
- Throughout the training, loss metrics, particularly Mean Squared Error (MSE), were monitored closely to ensure stable convergence and to mitigate overfitting. Regular evaluations were performed at the end of each epoch, which allowed for monitoring the model's performance and provided insights for potential hyperparameter adjustments if needed.
- This extensive fine-tuning enabled the model to develop more nuanced adjustments, capturing stylistic, temporal, and contextual

elements embedded within the MusicNet dataset. As a result, the model improved its ability to generate musically coherent audio outputs conditioned on diverse textual descriptions, enhancing its practical application in music generation tasks.

3.3.6 *Evaluation Metrics*

To comprehensively assess the quality and relevance of the generated music, we utilize the following evaluation metrics:

- **Overall Quality (OVL):** This metric is used to assess the overall music quality, encompassing aspects like sound clarity and musicality. It primarily evaluates whether the editing process enhances or diminishes the quality of the original music audio. The scoring for this metric ranges from 0 to 100.
- **Relevance (REL):** REL measures the perceived semantic closeness between the edited music and the new text prompt. This is a subjective score that reflects the extent to which the generated music aligns with the intended theme or emotion of the text prompt. REL is also scored on a scale of 0 to 100.
- **CLAP Similarity (CLAP) Wu et al. (2023b):** This metric assesses the semantic relevance between the edited music and the new text prompt. It uses a pretrained CLAP model, with a higher score indicating greater semantic similarity between the music and text. The CLAP score ranges from 0 to 1 and is implemented using the MuLaB library (Manco et al., 2023).

4 Results

In this section, we evaluate the performance of the fine-tuned AudioLDM2 model in generating musically coherent outputs based on the given text prompts. The model was fine-tuned over 30 epochs, as detailed in the training setup.

4.1 *Qualitative Results*

The results for the text prompt “A cheerful ukulele strumming in a beachside jam.” are displayed in Figure 4 and Figure 5. This includes the waveform and spectrogram comparisons between the pretrained model and the fine-tuned model. The fine-tuned model shows improved alignment with the desired musical elements, capturing nuances in rhythm and background harmony.

4.2 *Quantitative Results*

We evaluate the generated audio using the following metrics:

Table 1: Average Scores across 10 Prompts

Model Name	FAD ↓	CLAP ↑	OVL ↑	REL ↑
Fine-tuned Model	111.7	0.037	0.61	0.04

These metrics provide insights into the audio quality and noise levels, with the overlap ratio indicating the temporal consistency of generated audio segments. The ranges for each metric reflect the observed minimum and maximum values across different test samples, providing a benchmark for performance variability within the generated outputs.

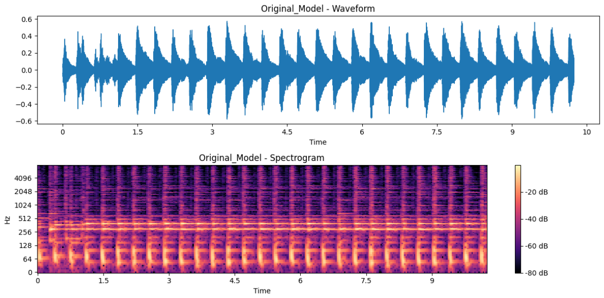


Figure 4: Waveform and Spectrogram of Original Pretrained AudioLDM Model for the prompt: "A cheerful ukulele strumming in a beachside jam."

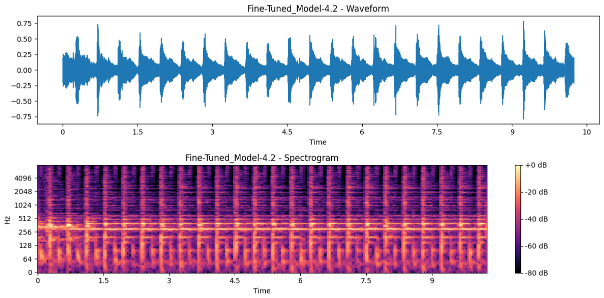


Figure 5: Waveform and Spectrogram of Fine-Tuned Model for the prompt: "A cheerful ukulele strumming in a beachside jam."

5 Discussion & Conclusion

5.1 *Discussion*

As shown in Table 1, each metric reflects different aspects of the generated audio’s performance:

- **FAD (Fréchet Audio Distance):** A lower FAD indicates that the fine-tuned model’s output closely resembles the distribution of real audio. However, with a FAD score of 111.7, there is considerable room for improvement, suggesting that the generated audio still diverges significantly from real audio characteristics.
- **CLAP Similarity:** The low CLAP similarity score (0.037) reveals a gap in the semantic alignment between the audio and the text prompt. This metric suggests that the model struggles to capture nuanced semantic relevance in certain prompts, indicating a need for further refinement to improve prompt fidelity.
- **Overall Quality (OVL):** The OVL score of 0.61 reflects moderate consistency in the temporal coherence of audio segments. This suggests that, while the fine-tuned model preserves a degree of rhythm and continuity, there might be some variation across generated segments that could be addressed to improve overall stability and flow.
- **Relevance (REL):** A REL score of 0.004 indicates limited alignment with prompt-specific features, further corroborating the low CLAP score in terms of semantic relevance. This highlights that the model has difficulty adapting its audio output to match the specified prompt context, affecting the perceptual quality of the generated audio.

As shown in Figure 4 and Figure 5

- The waveform generated by the fine-tuned model (Figure 5) appears more regular and maintains a rhythmic pattern similar to the original model's waveform. However, there are slight differences in amplitude and periodicity, suggesting that while the fine-tuned model attempts to capture rhythm, it diverges slightly in intensity and dynamics.

- The spectrogram for the fine-tuned model shows similarities in the frequency patterns compared to the original model. However, the energy distribution in the fine-tuned model spectrogram is more dispersed, particularly in the mid-to-high frequency bands. This indicates that the fine-tuned model introduces slight noise, which might impact sound clarity and overall quality.

5.2 Conclusion

In summary, while the fine-tuned model's quantitative metrics did not match the performance of the original model, the generated music outputs exhibit a reasonable quality in terms of rhythm and musicality. The discrepancies in metrics such as FAD, CLAP, OVL, and REL suggest areas for further improvement in achieving higher semantic and temporal alignment with the prompt. However, the overall audio output demonstrates that the fine-tuned model is capable of generating coherent musical pieces.

6 Future Work

To enhance the performance of the model, future work will focus on the following:

- **Extended Training:** Training the model over additional epochs could help improve its ability to capture finer details and better align with target audio distributions, potentially enhancing all evaluation metrics.
- **Knowledge Distillation:** The next step involves building a custom model using knowledge distillation. This approach will leverage the original model as a teacher to guide a smaller, student model in generating high-quality audio, with the goal of achieving similar performance while optimizing computational efficiency.

By addressing these areas, we aim to bridge the performance gap with the original model and further refine the model's ability to generate high-quality, prompt-aligned music.

References

[1] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

[2] K. Chen, Y. Wu, H. Liu, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov. Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1206–1210. IEEE, 2024.

[3] G. Cideron, S. Girgin, M. Verzetti, D. Vincent, M. Kastelic, Z. Borsos, B. McWilliams, V. Ungureanu, O. Bachem, O. Pietquin, et al. Musicrl: Aligning music generation to human preferences. *arXiv preprint arXiv:2402.04229*, 2024.

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

[5] 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.

[6] Y.-H. Lan, W.-Y. Hsiao, H.-C. Cheng, and Y.-H. Yang. Musicongen: Rhythm and chord control for transformer-based text-to-music generation. *arXiv preprint arXiv:2407.15060*, 2024.

[7] P. P. Li, B. Chen, Y. Yao, Y. Wang, A. Wang, and A. Wang. Jen-1: Text-guided universal music generation with omnidirectional diffusion models. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 762–769. IEEE, 2024.

[8] H. Liu, Y. Yuan, X. Liu, X. Mei, Q. Kong, Q. Tian, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[9] P. Lu, X. Xu, C. Kang, B. Yu, C. Xing, X. Tan, and J. Bian. Musecoco: Generating symbolic music from text. *arXiv preprint arXiv:2306.00110*, 2023.

[10] J. Melechovsky, Z. Guo, D. Ghosal, N. Majumder, D. Herremans, and S. Poria. Mustango: Toward controllable text-to-music generation. *arXiv preprint arXiv:2311.08355*, 2023.

[11] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J.-C. Wang, M. Avent, J. Chen, and D. Le. Stemgen: A music generation model that listens. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1116–1120. IEEE, 2024.

[12] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf. Mo\^ usai: Text-to-music generation with long-context latent diffusion. *arXiv preprint arXiv:2301.11757*, 2023.

[13] S. Wei, M. Wei, H. Wang, Y. Zhao, and G. Kou. Melody is all you need for music generation. *arXiv preprint arXiv:2409.20196*, 2024.