A REVIEW OF GENERATIVE MUSIC MODELS

TECHNICAL REPORT

Shizhuo Li
Carnegie Mellon University
Pittsburgh, PA 15213
shizhuol@andrew.cmu.edu

Sunny Tang
Carnegie Mellon University
Pittsburgh, PA 15213
jiahuit@andrew.cmu.edu

ABSTRACT

This report provides a review of generative music models, highlighting recent advancements and identifying persistent challenges within the field. As artificial intelligence continues to develop, generative music models have employed sophisticated machine learning techniques such as VQ-VAE, transformers, and sequence-to-sequence models to create music. This report also reviews some recent outstanding models such as Jukebox, MusicLM, and Musicgen, examining their architectures, methodologies, applications, and limitations. We also assess some models' performance, focusing on their ability, particularly their performance of text-to-music generation and consistency of the model. The results of the experiments show varied performance and consistency in the music generated, followed by a comprehensive analysis. The findings suggest the potential of generative model developments in the future and the need for further refinement and advancement within the field of generative music models.

1 Introduction

As technology advances in recent decades, artificial intelligence has opened up unprecedented opportunities for creative innovations in different realms. Within the field of art and music, there have long been enormous efforts put into investigating automatic music-generative models. The introduction of deep neural network architecture has contributed immensely to the early development of models to process musical inputs, extract features, and compose short sequences of melodies that resemble the original piece [Zhu et al., 2023].

While most of the prior work focuses on simulating the human composition of music through sampling and generating fragments of melody symbolically [Moorer and Anderson, 1972], the models fall short of capturing and identifying voices, lyrics, and most subtle dynamics that are essential to music.

To address such limitations, some later research has introduced revolutionary approaches with more sophisticated algorithms and machine-learning techniques such as VQ-VAE(Vector Quantized Variational Autoencoder), which not only enables unseen lyrics generation and the re-rendering of songs but also allows for conditioning on a variety of artists and genres [Dhariwal et al., 2020].

In addition to symbolic music generation, recent research took an innovative step to investigate music generation conditioned on text descriptions. Specifically, [Agostinelli et al., 2023] introduced MusicLM, which leverages prior works such as AudioLM, a framework for audio generation, and MuLan, a joint music-text model that pairs music and corresponding test descriptions in an embedding, to train and generate based on text-conditioning signals. [Copet et al., 2024] presents a single model that allows for both text and melody inputs to generate music at 32 kHz, a higher frequency than the output of MusicLM. These recent efforts further challenge the limitations of music generation and extend the possibility of carrying out different music creation tasks.

Nevertheless, many existing models require extensive human guidance for consistent style and patterns when producing long pieces of melodies [Zhu et al., 2023]. Most previous works do not provide a comprehensive analysis of algorithms and models and present conservative future trajectories on the development of music generation systems [Wang et al., 2023]. The lack of creativity and originality of AI-composed music due to existing models' overreliance on training data has also been discussed [Zhu et al., 2023].

From creating compositions reminiscent of classical masterpieces to pushing the boundaries of traditional music with experimental soundscapes, despite contemporary limitations, these models present abundant creative possibilities for human musicians while also making music production and customization more accessible to the general public. Currently, generative music models encompass a wide range of methodologies and approaches, each with its own strengths, limitations, and applications.

With the growing need to delve into the capabilities and explore diverse applications of generative music models, this paper will first introduce relevant musical terminologies used in developing music generation models and provide a comprehensive overview of common machine learning architecture such as VQ-VAE, transformers, and sequence-to-sequence models. This paper also aims to review relevant literature on existing models and provide a holistic analysis that draws connections among diverse algorithms and results from past research. Given that only some of the models presented are open-sourced, we will only compare and contrast selected generative music models against a set of evaluation metrics to suggest practical integration or shed light on future research directions that can address and overcome limitations identified in the literature review.

2 Concept Overview

Before delving deeper into complex machine-learning models that realize automated music generation, we will first introduce various key musical terminologies that are essential for understanding the fundamental concepts that these models are built upon and the intricacies of music compositions. [Zhu et al., 2023]

i). **Pitch** is directly related to the frequency of a sound, which describes the number of cycles of vibration per unit of time. Higher frequencies correspond to higher pitches and vice versa.

ii). **Tone** refers to a sound characterized by a specific pitch or frequency. It contributes to the overall quality or timbre of a music composition.

iii). **Timbre** describes the quality and characteristics of a sound, which helps to distinguish different sounds even when they have the same pitch or volume.

iv). **Harmony** represents the combination of different simultaneous sounds and pitches that convey a pleasing auditory experience to the audience.

v). **Melody**, or tune, refers to a sequence of notes arranged in a meaningful and expressive manner. It's an essential element in conveying emotions and providing structure and coherence to music compositions.

3 Music Generative Models: Technique, Applications, Limitations

In this section, we will introduce some popular generative music models and explore various machine-learning techniques to analyze and evaluate the different methodologies and applications of automated music composition. We conducted a thorough literature review of the following models and their architectures

3.1 Jukebox: Vector Quantized Variational Autoencoder (VQ-VAE) [Dhariwal et al., 2020]

The quantization-based approach VQ-VAE is a common structure utilized by early audio and music generative techniques. Introduced by [van den Oord et al., 2017], this model relies on vector quantization (VQ) and combines the variational autoencoder (VAE) framework with discrete latent representations. Compared to other VAE models, VQ-VAE allows for more effective use of the latent space and the flexibility of discrete distributions. In specific, it can model features that span across multiple dimensions and reconstruct at low bitrates in various domains. Given its distinctive nature, VQ-VAE introduces opportunities for generating raw high-fidelity audio while ensuring long-term musical coherence.

Traditional VQ-VAE models are composed of three parts: an encoder, a bottleneck, and a decoder. The encoder creates embeddings by condensing the original audio input. The length of these latent representations compared to the original audio length dictates the level of compression, which influences the balance between maintaining accuracy and ensuring logical flow. The bottleneck will transform the embeddings from the encoder into code vectors with codebook lookup. Finally, the decoder will reconstruct raw audio from the latent representations [Dhariwal et al., 2020].

Aiming to generate high-quality music in raw audio, Jukebox is built upon a hierarchical VQ-VAE, which can compress the music into discrete codes, removing the audio but retaining the essential details of pitch, timbre, and volume. In order to eliminate issues of codebook collapse from using successive encoders with this model, Jukebox adopts a simplified modification with only feedforward encoders and decoders. Figure 1 provides an overview of the architecture

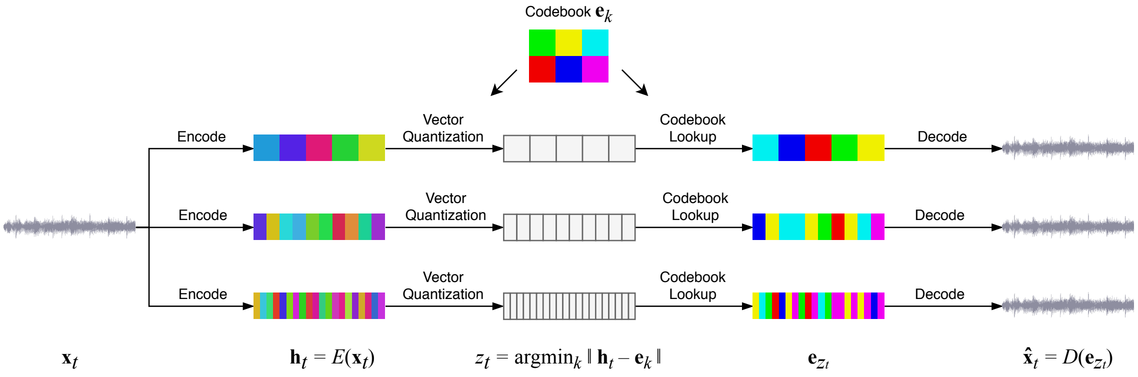

Figure 1: An overview of Jukebox VQ-VAE model for music generation. The bottom level focuses on quality reconstruction while the top level encoding retains only the essential musical information such as pitch, timbre, and volume [Dhariwal et al., 2020]

utilized by Jukebox. Three separate prior models are trained with different temporal resolutions and autoregressive transformers are incorporated to model sequences of discrete codes which enables Jukebox to condition on artists, genre, and lyrics while also generating music that maintains long-range coherence [Dhariwal et al., 2020].

Provided with artist, genre, and lyrics as input, the model can carry out a variety of different tasks, ranging from the generation of unseen lyrics and completion of songs to re-renditions of music. However, even though the model introduces techniques to enhance coherence in creating long pieces of music, it fails to establish a musical and emotional structure to maintain quality and consistency across a single piece without any unwanted noise. In specific, the generated output does not follow repeating choruses or a common theme over time. Furthermore, the upsampling process is implemented sequentially, it's extremely inefficient for the model to render a single piece of audio and thus cannot yet be applied under interactive contexts [Dhariwal et al., 2020].

3.2 MusicLM: Hierarchical Autoregressive Modeling [Agostinelli et al., 2023]

One of the most recent music generation models is MusicLM, which allows for high-fidelity music generation from text descriptions. This model leverages a hierarchical sequence-to-sequence modeling approach, enabling the synthesis of music at 24 kHz that is coherent over extended periods, overcoming previous limitations to generating longer and more complex audio sequences. Specifically, MusicLM made efforts to address the challenge proposed by Jukebox, attempting to eliminate any noticeable artifacts introduced in the produced output [Agostinelli et al., 2023].

MusicLM extends the capabilities of AudioLM, a model designed to generate high-quality audio with long-term consistency [Borsos et al., 2023], through three key enhancements: (i) integrating descriptive text into the generation process, (ii) demonstrating the extension of this conditioning to other signals like melody, and (iii) modeling a wide range of lengthy music sequences spanning various genres beyond just piano music, including drum'n'bass, jazz, and classical compositions [Agostinelli et al., 2023]. To accomplish this, the architecture of MusicLM, as shown in figure 2, mainly consists of three models: SoundStream, w2v-BERT, and MuLan. In specific, the SoundStream model utilizes a structure that combines a convolutional encoder/decoder network with a residual vector quantizer, allowing for efficient compression of speech, music, and general audio at bitrates typically targeted codecs tailored for speech [Zeghidour et al., 2021]. Additionally, similar to AudioLM, aiming to produce music with long-term coherence, MusicLM incorporates "an intermediate layer of the masked-language-modeling (MLM) module of a w2v-BERT model" [Agostinelli et al., 2023], which introduces a combination of MLM and contrastive learning for self-supervised speech representation [Chung et al., 2021]. The final component of MusicLM is MuLan, which provides direct linkages between audio and unconstrained natural language descriptions. MusicLM utilizes MuLan's music embeddings for training and its text embeddings during inference. The two-tower model MuLan is trained on 44 million music recordings and associated free-form text annotations, producing embeddings that encompass existing ontologies while also allowing for true zero-shot functionalities [Huang et al., 2022]. When training MusicLM, continuous representations of the target audio sequence are extracted from MuLan's audio-embedding network, which can be directly used as conditioning signals in the autoregressive transformer model. Conditioning on MuLan's audio embeddings at training brings several benefits. Firstly, it enables scalability of the training data as text captions are not required. Secondly, leveraging MuLan's robustness to noisy text descriptions, acquired through its contrastive loss training, enhances musicLM's resilience to such noise during inference [Agostinelli et al., 2023].

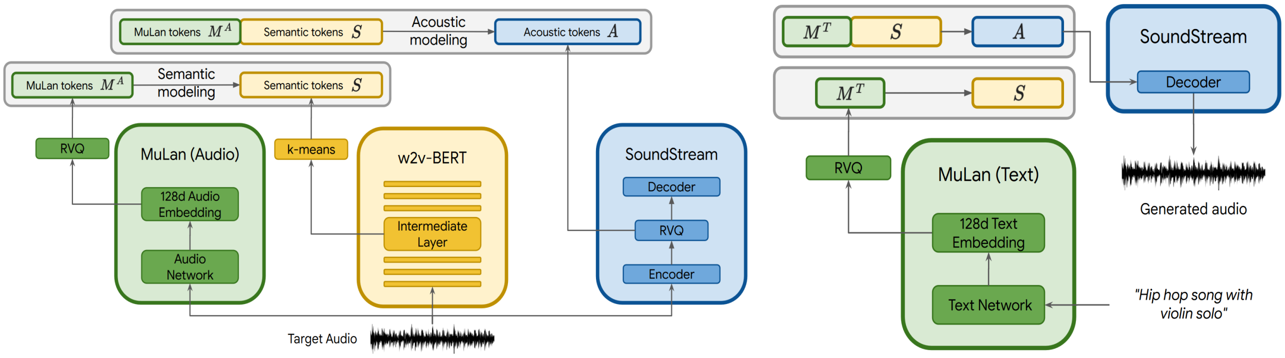

Figure 2: The figure on the left shows how the model is trained. The three models are pre-trained independently and every modeling stage is sequence-to-sequence using decoder. The figure on the right shows the inference phase. Mulan text tokens obtained from the text input are processed as conditioning signals and the decoder in SoundStream transforms the resulting audio tokens into waveforms [Agostinelli et al., 2023]

MusicLM is evaluated with MusicCaps, a dataset consisting of text pairs with rich text descriptions released by Agostinelli et al. [2023]. This dataset is crucial for evaluating the performance of MusicLM and future models in the text-to-music generation domain. The performance of MusicLM is assessed according to two criteria: the quality of output audio and the adherence of music to text descriptions. Evaluating against metrics like KL divergence(KLD) and MuLan Cycle Consistency(MCC) developed given the above two aspects, MusicLM is compared with two proposed baseline models, Mubert [Mubert-Inc, 2022] and Riffusion [Forsgren and Martiros, 2022], that yield above-average performances among models introduced prior to MusicLM. Based on the results, MusicLM produces music audio comparable to Mubert in terms of quality and yields the best performance among the three models in terms of adherence and the ability to extract information from texts. In specific, MusicLM achieves the lowest KL divergence score, indicating that the music generated has the most similarities in acoustic characteristics as the reference audio provided, and the highest MCC score, which quantifies the similarity between music and text descriptions [Agostinelli et al., 2023].

Extending the capabilities of prior music models like Jukebox, MusicLM introduces conditioning on melody by providing pieces of audio "in the form of humming, singing, whistling, or playing an instrument" in addition to text descriptions such as artists and genres. Furthermore, since MusicLM performs autoregressive generation in the temporal dimension, it can create longer sequences that are coherent for up to several minutes and allow for generation under story mode, which refers to composing music while changing text descriptions [Agostinelli et al., 2023].

Despite its robust performance, there are risks and limitations associated with MusicLM. Since the model utilizes MuLan to represent conditioning, it inherits some of its limitations. In particular, MusicLM does not demonstrate a comprehensive understanding of negation and adherence to precise temporal ordering in the text input [Agostinelli et al., 2023]. In addition, MusicLM is incapable of lyrics generation, which is a functionality of Jukebox, and modeling high-level musical concepts such as verse and chorus, suggesting directions for future related research.

3.3 MusicGen: Single-Stage Transformer with Encodec Tokenizer [Copet et al., 2024]

Another model introduced by recent research is musicGen. Compared to Jukebox, MusicGen's architecture simplifies the generation process by training a single-stage autoregressive transformer language model over a 32kHz Encodec tokenizer with 4 codebooks sampled at 50 Hz. The model also utilizes efficient token interleaving patterns, eliminating the need for hierarchical or upsampling models. This addresses issues of low efficiency introduced by Jukebox's sequential sampling process and significantly improves the performance of music generation. Moreover, MusicGen is trained using autoregressive transformers of different sizes: 300M, 1.5B, and 3.3B, along with memory-efficient attention to improving memory usage and processing speed when generating long sequences of outputs [Copet et al., 2024].

Building upon works proposed by MusicLM, MusicGen utilizes an autoregressive transformer-based decoder, conditioned on a text or melody representation by providing a conditioning tensor as a prefix to transformer inputs. In addition, many frameworks of previous compression models produce parallel streams comprised of tokens originating from different learned codebooks. To address this issue, MusicGen operates over quantized units from EnCodec, a neural audio compression model that's able to produce high-fidelity audio samples across a range of sample rates and bandwidths [Défossez et al., 2022]. This convolutional auto-encoder is quantized using Residual Vector Quantization(RVQ), which produces non-independent quantized values for different codebooks as the quantization error of the previous quantizer is used to encode the next one [Copet et al., 2024].

Furthermore, MusicGen introduces codebook interleaving strategies to enhance the efficiency of music generation. Specifically, [Copet et al., 2024] generalizes the framework to various codebook interleaving patterns: flattening pattern, parallel pattern, coarse first pattern, and delay pattern. When performing autoregressive modeling, each pattern has its own benefits and drawbacks. In specific, although flattening improves the generation of music, it comes at a significantly higher computational cost while a simple delay approach yields similar performance for a fraction of the cost. Hence, MusicGen adopts delay interleaving patterns to boost the model's overall performance and the quality of resulting outputs.

Similar to MusicLM, MusicGen conditions on text and melody. To represent text for conditional audio generation, the model adopts a pre-trained text encoder T5, which has proven to yield the best performance when evaluating various metrics. In addition to text-conditioning, the model can also compose high-quality samples while guided by melodies. To achieve this, the model conditions on the input's chromagram and incorporates an information bottleneck by choosing the dominant time-frequency pin at each time step to avoid possible overfitting in the reconstruction process [Copet et al., 2024].

To evaluate MusicGen, [Copet et al., 2024] conducted experimental studies to compare its performance with MusicLM and the two baseline models used to assess MusicLM. Specifically, the models are mainly assessed according to the Frechet Audio Distance (FAD). Among the baseline models, MusicGen demonstrates a relatively low FAD score, which indicates the plausibility of the model. Moreover, based on the results from evaluation by human listeners, MusicGen produces music audio that is both of the highest quality and adherence to text descriptions compared to the proposed baseline models. To evaluate MusicGen's performance when conditioned on melodic representations, a new metric chroma cosine-similarity is used, which measures the similarity between the generated music and the provided melody. The results indicate that the composed output by MusicGen successfully follows and incorporates the given melody [Copet et al., 2024].

Although MusicGen is proven to be more efficient and produces higher quality and more adherent music audio given text descriptions, there's not much fine-grained control over adherence of the output as MusicGen is constructed with a simple generation method. Furthermore, melody conditioning presented by this model requires more future research on data segmentation, types, and the amount of guidance in order to yield generations with outstanding performance and quality [Copet et al., 2024].

4 Experiment

We reviewed and compared the models above and have designed experiment to test their reliability and validity. Questions of the following are asked:

1. What is the performance of each model and how do they compare?
2. How consistent are the performance of models, are they reliable in performance?
3. How consistent are the models when generating, after a few seconds of music, will the quality get lower for the following continued generation?

In this section, we design experiments to answer the above question and test the ability of text to music generations of some music generation models.

We subsampled a dataset from a subset of MusicCaps [Agostinelli et al., 2023], a collection designed to aid research in the area of music understanding and generation. It includes rich textual descriptions of music tracks provided by human annotators. MusicCaps contains pieces of music in 10 seconds, and a description of music in natural language. These descriptions are not merely technical but capture the emotional tone, themes, and other abstract aspects of music that are typically perceived by human listeners. Here is an example of a natural language description of music in MusicCaps[Agostinelli et al., 2023]:

> *This song features an electric guitar as the main instrument. The guitar plays a descending run in the beginning then plays an arpeggiated chord followed by a double-stop hammer onto a higher note and a descending slide followed by a descending chord run. The percussion plays a simple beat using rim shots. The percussion plays in common time. The bass plays only one note on the first count of each bar. The piano plays backing chords. There are no voices in this song. The mood of this song is relaxing. This song can be played in a coffee shop.*

We sampled a subset of 30 music-text pairs, which will be used in following evaluation and is designed in three parts. We are to compare and test models from MusicLM [Agostinelli et al., 2023] and MusicGen [Copet et al., 2024] series to test which includes total of four models including: `MusicLM` [1], `MusicGen-large` [2] , `MusicGen-small` [3], and `MusicGen-medium` [4] Since MusicLM contains 3 components including Mulan, which has no pre-train open source release, we decided to include a pre-trained model and adapt code from this Github repository [5].

4.1 Evaluation of performance

We run a test on the four models chosen with the dataset we subsampled from MusicCaps, that is, let the models generate music from natural language descriptions in the dataset. After we received the music pieces generated, we compared them with the reference music from the dataset, that is, for a valid model, the music generated should have similar instruments, pace, emotion, genre, and other dimensions described, that is sharing similarity with the reference music in the dataset. We generated 30s of music for each MusicGen model and 10s for MusicLM.

To evaluate similarity, we choose the metric Frechet Audio Distance [Gui et al., 2024] (FAD). FAD score compares real and generated embedded audio by extracting features, modeling them as Gaussian distributions, and calculating the Fréchet distance between these distributions to gauge similarity.

To encode the audios into tensors for FAD, we chose the following embedding models for FAD score computation: `larger_clap_music` [Wu et al., 2022][6] , and `Vggish` [Koh and Dubnov, 2021][7]. The scores are computed and shown in table 11.

Table 1: Average FAD score of models between reference and generated music

Model Name	**FAD** `Vggish`	**FAD** `larger_clap_music`
MusicLM	7.55	6.93
MusicGen-small	6.32	4.5
MusicGen-medium	6.21	4.39
MusicGen-large	5.99	4.24

Observed that the performance of the MusicGen series for both embedding models is notably superior to that of the MusicLM. A lower FAD score indicates a greater degree of similarity between the reference and the generated music, suggesting a higher fidelity of the generated output in relation to the target or reference audio. This finding is significant as it highlights the ability of the MusicGen models to closely mimic the desired musical attributes encoded within the embeddings.

Furthermore, within the MusicGen series, the larger models demonstrate enhanced performance compared to their smaller counterparts. The reason is obvious since larger models have more parameters than smaller ones. This trend suggests a positive correlation between model size and the quality of music generation, with larger models possessing an increased capacity to capture the complex features in music composition and replication.

4.2 Evaluation of Performance Consistency Across Multiple Generations

To rigorously assess the model's consistency, hence the reliability, of the model, for each selected prompt, we instigated the generative process of the model to produce a total of 20 distinct musical generations. This approach is designed to evaluate the model's capacity to maintain a stable generative behavior across multiple iterations of a single prompt. For the metric, we also use FAD scores to quantitize the similarity between each pair of generated music pieces. The result of test is shown in table 2.

[1] https://github.com/lucidrains/musiclm-pytorch

[2] https://huggingface.co/facebook/musicgen-large

[3] https://huggingface.co/facebook/musicgen-small

[4] https://huggingface.co/facebook/musicgen-medium

[5] https://github.com/BarbosaRT/open_musiclm_colab

[6] https://huggingface.co/laion/larger_clap_music

[7] https://www.kaggle.com/models/google/vggish

Table 2: Average FAD score of models with 20 generations from one prompt

Model Name	**FAD** `Vggish`	**Variance of FAD** `Vggish`	**FAD** `Larger_clap_music`
MusicLM	4.32	2.44	4.05
MusicGen-small	1.52	0.36	1.44
MusicGen-medium	0.93	0.13	1.37
MusicGen-large	1.52	0.48	1.39

Observe that the MusicGen series models outperform MusicLM in terms of the Fréchet Audio Distance (FAD) scores, indicating a more consistent and reliable performance. This superiority is further underscored by the smaller variance in the FAD scores for the MusicGen series compared to MusicLM.

Surprisingly, the MusicGen-Medium model stands out for its exceptional consistency, achieving the lowest FAD scores and exhibiting the smallest variance among the scores. This suggests that it's particularly stable across different evaluations.

4.3 Evaluation of Consistency during Music Generation

In this section, our objective is to evaluate the model's ability to sustain the feature of music across the entire span of a musical piece. That is, for a given piece of music, the attributes characterizing the initial segment should be in close similarity to those defining the concluding segment. To conduct this assessment, we initially generated musical outputs from our dataset. After that, each musical composition was split at its midpoint to yield two segments, that are, the first and second half of music pieces.

We measured the extent of similarity between the two halves again by FAD scores as the evaluation metric. The results are shown below in table 3.

Table 3: Avg FAD score between the first and second half of music pieces

Model Name	**FAD** `Vggish`	**FAD** `larger_clap_music`
MusicLM	3.78	3.81
MusicGen-small	2.31	2.26
MusicGen-medium	1.78	1.80
MusicGen-large	1.73	1.75

We can infer that across the board, all models exhibit a degree of internal consistency, with the MusicGen-large model showing the most consistency. The MusicGen-small model shows less consistency than the medium and large models, as indicated by its higher FAD scores. The MusicLM model, however, has higher FAD scores than all the MusicGen models, suggesting it has less internal consistency. The models that show internal consistency within a generation of music may preserve the consistency of music features longer.

5 Discussion

Based on the results from the above experiment, MusicGen achieves better performance from all three aspects. In particular, out of the three model parameters we experiment with MusicGen, the model trained with the largest-sized autoregressive transformers yields the best performance in terms of quality and consistency. The effect of model size reflected in this experiment is consistent with conclusions drawn by [Copet et al., 2024] since larger models can better process and understand the input text descriptions, providing outputs of higher quality and with better adherence to given prompts.

From the results, we observe the results from the comparison between MusicGen and MusicLM are expected since MusicGen made structural improvements on the basis of MusicLM. The differences in their ML architecture underlie the differences in their performances. In specific, MusicLM utilizes a hierarchical sequence-to-sequence approach whereas MusicGen adopts a single-stage transformer along with token interleaving techniques, which significantly simplifies the process with greater efficiency and better performance. We also note that since the open-sourced code for MusicLM did not release the trained MuLan model or the dataset used when training, we conducted the experiment with an unofficial pre-trained MuLan, which could influence the performance of our MusicLM model, contributing to the slightly higher FAD scores.

In addition, we also note that since MusicGen and MusicLM are trained with different datasets, the models may overly rely on the training data. The potential lack of diversity or focus on specific types of music (genre, artists, etc) may introduce bias and affect the models' performance when assessed with datasets given its prior exposure to similar data. Therefore, we observe that the results of this experiment may be influenced by external factors besides those we are investigating. Future research can shed light on approaches to training and testing models in ways that address and mitigate biases introduced during the development process, suggesting a generative music architecture that yields superior performance regardless of input characteristics.

References

Yueyue Zhu, Jared Baca, Banafsheh Rekabdar, and Reza Rawassizadeh. A survey of ai music generation tools and models. 2023.

Moorer and James Anderson. Music and computer composition. *Communications of the ACM*, 15(2):104–110, 1972.

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.

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. Musiclm: Generating music from text. 2023.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. 2024.

Lei Wang, Ziyi Zhao, Hanwei Liu, Junwei Pang, Yi Qin, and Qidi Wu. A review of intelligent music generation systems. 2023.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. 2017.

Zalan Borsos, Raphael Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. Audiolm: a language modeling approach to audio generation. 2023.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. 2021.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. 2021.

Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. Mulan: A joint embedding of music audio and natural language, 2022.

Mubert-Inc. Mubert. `https://mubert.com/`, 2022.

S. Forsgren and H. Martiros. Riffusion. `https://www.riffusion.com/#`, 2022.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression, 2022.

Azalea Gui, Hannes Gamper, Sebastian Braun, and Dimitra Emmanouilidou. Adapting frechet audio distance for generative music evaluation, 2024.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation, 2022. URL `https://arxiv.org/abs/2211.06687`.

Eunjeong Koh and Shlomo Dubnov. Comparison and analysis of deep audio embeddings for music emotion recognition. In *Proceedings of the Conference Name*, pages FirstPage–LastPage, City, Country, April 2021. Organization or Publisher. Available: `LinkToThePaperIfAvailable`.