Making Silicon Sing

Stanford CS224N Custom Project

Kadija Ismail
Department of Computer Science
Stanford University
kismail@stanford.edu

Imen Kedir
Department of Computer Science
Stanford University
imenked@stanford.edu

Abstract

This paper investigates recent advances in audio and music generation conditioned on natural language, focusing on exploring limitations inherent in existing audio modeling approaches. Previous approaches such as Jukebox(Dhariwal et al. (2020)), AudioLM(Borsos et al. (2023)) and MusicLM(Agostinelli et al. (2023)) employ multiple streams of encoded audio—semantic tokens for high-level content and acoustic tokens for detailed information—and integrate textual conditioning, which increases complexity and computation due to need for multiple encoders/decoders and autoregressive transformers. Meta's MusicGen(Copet et al. (2024)) and AudioGen(Kreuk et al. (2023)) simplify the architecture with a single audio encoder and single-stage transformer LM, using efficient token interleaving patterns and a pre-trained T5 text encoder for text conditioning, reducing complexity but offering less control. This simplification has led to challenges in handling acapella and music with vocals.

To enhance the performance of MusicGen, we curated a dataset of 50 studio-quality acapella tracks, their instrumental accompaniments, and the full songs with both acapella and instrumental elements combined. This collection amounts to roughly seven and a half hours of audio data in total. Using Essentia, an open-source library for audio and music analysis and description, we extract natural language descriptions of the songs to enrich our dataset. We propose fine-tuning the model specifically on datasets containing lyrics, instrumentals, and vocals, aiming to identify and address its limitations through targeted data augmentation and improved conditioning techniques. Our approach seeks to bridge the gap between simplicity in architecture and high-quality, conditionally accurate audio generation.

1 Key Information to include

- Mentor: Josh Singh

2 Introduction

Generating high-quality audio conditioned on natural language poses several significant challenges. Previous transformer-based autoregressive models for audio face difficulties handling the long-range sequences produced by many off-the-shelf audio encoders. Due to the high sampling rates of audio data, typically 24kHz or 48kHz, even a few seconds of audio can consist of hundreds of thousands of samples. This immense volume of data presents significant challenges for processing and generating audio, especially in the context of modeling long-range sequences and maintaining high fidelity in the generated output. While audio encoders with large downsampling factors can mitigate this issue, they often result in reduced audio reconstruction quality, making it difficult to balance between efficiency and audio fidelity.

Recent models like AudioGen and MusicGen have introduced simplifications in architecture by leveraging EnCodec, a model trained specifically to compress audio and reconstruct the original signal with high fidelity. EnCodec consists of an autoencoder with a residual vector quantization bottleneck that produces several parallel streams of audio tokens with a fixed vocabulary. This approach significantly reduces the dimensionality of the audio representation while still allowing for high-quality reconstructions when decoded. This simplification offers an efficient solution for audio compression and reconstruction, but it also raises questions about its ability to handle more complex audio tasks, such as generating coherent acapella music.

Previous research utilizing multiple streams of encoded audio, such as AudioLM, has demonstrated that different types of tokens capture distinct aspects of the audio. For instance, in modeling speech, it was found that semantic tokens primarily capture linguistic content, while acoustic tokens capture speaker identity and recording conditions. This division of labor among different token types allows for more detailed and accurate audio modeling. However, the model we chose, MusicGen, does not use semantic tokens and relies solely on audio tokens. This absence of semantic tokens in MusicGen raises concerns about its ability to produce coherent music with acapella components.

To explore these limitations and potentially enhance the performance of MusicGen, we curated a dataset of 50 studio-quality acapella tracks, their instrumental accompaniments, and the full songs with both acapella and instrumental elements combined. This collection amounts to roughly seven and a half hours of audio data in total. By fine-tuning the model specifically on datasets containing lyrics, instrumentals, and vocals, we aim to identify and address its limitations through targeted data augmentation and improved conditioning techniques. Our approach seeks to bridge the gap between simplicity in architecture and high-quality, conditionally accurate audio generation.

3 Related Work

3.1 Advances in Generative Music Models

Generative music models have evolved significantly over the years, moving from symbolic representations like MIDI to more complex audio representations such as mel-spectrograms. Early works in symbolic music generation, like those using generative adversarial neural networks (GANs) or recurrent neural networks (RNNs), focused on generating music in a structured, symbolic format. While successful in some aspects, these approaches often lacked the ability to capture the nuanced details of audio signals.

Recent advances have shifted towards encoding audio into a discrete space using techniques like residual vector quantization (RVQ). Models such as VQ-VAE (van den Oord et al. (2018)) demonstrated impressive reconstruction quality at low bitrates across various domains. Building on this, EnCodec (Défossez et al. (2022)) further extended these capabilities, offering high-fidelity audio compression and reconstruction by leveraging RVQ to create a hierarchical structure of quantizers. This hierarchical approach significantly reduces the dimensionality of the audio representation while maintaining high reconstruction quality, making it suitable for generative tasks.

3.2 Autoregressive Transformers in Audio Generation

AudioLM marked a notable shift from diffusion-based approaches to autoregressive transformers for audio generation. By leveraging tokens produced by a neural audio codec like SoundStream (Zeghidour et al. (2021)), AudioLM could generate high-fidelity audio without relying on text conditioning. Instead, it used semantic and acoustic tokens to capture high-level content and fine acoustic details. This allowed AudioLM to generate coherent and high-quality speech and music continuations, although it lacked the ability to directly incorporate text conditioning into the generative process.

3.3 Integrating Text Conditioning in Music Generation

MusicLM addressed the need for text conditioning by introducing a two-tower joint embedding mode which maps both audio and text captions into the same embedding space, enabling the generation of music from textual descriptions. MusicLM achieved this by creating a vast dataset of audio and weakly associated text captions, primarily sourced from YouTube video titles and comments. While

this approach demonstrated the potential for high-quality, text-conditioned music generation, the reliance on large-scale, weakly labeled data makes it infeasible for many researchers to replicate.

3.4 Simplifying Text Conditioning with Pre-trained Models

MusicGen offers a more accessible approach by utilizing an off-the-shelf T5 text embedding model for text conditioning. This simplification shows that decent results in text-conditioned music generation can be achieved without the need for extensive and customized datasets. By leveraging pre-trained models and efficient token interleaving patterns, MusicGen reduces the complexity of the architecture while still delivering satisfactory performance. This approach highlights certain challenges, such as handling acapella and music with vocals, which we aim to address through targeted fine-tuning and dataset augmentation.

4 Approach

We utilize the MusicGen model, which employs an autoregressive transformer architecture. This model uses the efficient acoustic token interleaving pattern described in the MusicGen paper, specifically the delay pattern. This pattern helps manage the complexity of audio sequences and ensures high-quality generation by effectively interleaving audio tokens.

4.1 Residual Vector Quantization and EnCodec

A critical component of our approach is the use of EnCodec, a model designed to compress audio into discrete codes and reconstruct the original signal with high fidelity. EnCodec uses residual vector quantization (RVQ), which maps each segment of audio to multiple codebooks. The encoder takes an audio extract and outputs a latent representation. The encoder model consists of a series of 1D convolutional layers, with C channels and a kernel size of 7, followed by B convolution blocks. Each block contains a residual unit and a down-sampling layer, doubling the number of channels whenever down-sampling occurs. The encoder outputs 75 latent steps per second for 24 kHz audio and 150 for 48 kHz audio. The nearest codebook to this embedding is selected as the first codebook, also known as the top-level prior. This codebook is the most important as it captures the primary features of the audio segment. The next $K - 1$ codebooks are selected based on the residual of the embedding and the first codebook. Each subsequent codebook captures the finer details not captured by the previous codebooks.This hierarchical approach to encoding audio allows us to decode multiple codebooks in parallel with our autoregressive transformer. We can decode the top-level prior of the next time step before decoding all the codebooks of the previous time step, as they are independent. This parallel decoding significantly enhances the efficiency and speed of the model, and reduces the distance of long range dependencies.

4.2 Pretrained Model

Our experiments begin with the pre-trained 1.5 billion parameter MusicGen melody model, which has been trained for audio continuation given natural language text descriptions. This model serves as the baseline for our fine-tuning efforts.

5 Experiments

5.1 Data

To investigate whether there are any differences in MusicGen's ability to model acapella versus instrumentals, we created the following:

- **Acapella Dataset:** This dataset consists of 50 studio-quality acapella tracks. Each track is a high-fidelity recording of vocal performances without any instrumental accompaniment. The dataset provides a rich source of vocal-only audio data for fine-tuning, allowing us to assess the model's capability to handle and generate vocal content.
- **Instrumental Dataset:** This dataset includes the instrumental accompaniments corresponding to the acapella tracks. It serves as a control to compare the quality of generated

instrumental music against the generated acapella. By evaluating the performance on this dataset, we can gain insights into how well the model generates instrumental music, considering that MusicGen has been exposed to more instrumental music in its pre-training phase.

- **Combined Dataset:** This dataset combines the acapella and instrumental tracks to form complete songs. It consists of the same 50 tracks, but each track includes both the vocal and instrumental elements. This comprehensive dataset allows us to train the model to generate full musical pieces that include both vocals and instrumentals, reflecting a real-world music generation scenario.

To enhance text conditioning, we use Essentia, an open-source audio analysis library, to extract detailed descriptions from our instrumental and full music datasets. Essentia analyzes each track to determine its mood, genre, and BPM, which we then use to create natural language captions for the audio, serving as input for the text conditioning component of MusicGen. To ensure consistency during evaluation and to prevent the model from relying on pre-existing patterns from its pre-training, we prefix each natural language description with a UUID hash. This approach allows us to use the same UUID hash during evaluation, ensuring that the model's generated output is influenced by the fine-tuning process rather than its original pre-training data.

5.2 Evaluation method

We employ multiple evaluation metrics to objectively and subjectively assess the performance of our models. Specifically, we use KL-Divergence and Fréchet Audio Distance (FAD) for objective evaluation and conduct subjective evaluations to capture qualitative aspects.

5.2.1 Kullback-Leibler (KL) Divergence

KL-Divergence measures the divergence between two probability distributions. We use a state-of-the-art audio classifier trained on AudioSet(Koutini et al. (2022); Gemmeke et al. (2017)) to compute the KL divergence between the label distributions of the original and generated audio. The KL divergence between two distributions P and Q is given by:

$$D_{KL}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{1}$$

In the context of audio classification, P represents the probability distribution of labels for the original audio, and Q represents the probability distribution for the generated audio.

5.2.2 Fréchet Audio Distance (FAD)

FAD measures the distance between the distributions of real and generated audio features. It assesses the quality of the generated audio by comparing the means and covariances of embeddings extracted from a pre-trained audio classifier (e.g., VGGish). The FAD between two distributions with means μ_r, μ_g and covariances Σ_r, Σ_g:

$$FAD = \|\mu_r - \mu_g\|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}) \tag{2}$$

where:

- μ_r and Σ_r are the mean and covariance of the real audio embeddings.
- μ_g and Σ_g are the mean and covariance of the generated audio embeddings.

5.3 Experimental details

The pre-trained MusicGen model is fine-tuned on each dataset for 3 epochs with a learning rate of 1.0. Fine-tuning is performed on 8 A40 GPUs to handle the computational demands of our large model parameters.

Table 1: Objective Metrics

Category	Comparison	KL-Divergence	Fréchet Audio Distance
Baseline	**Instrumental**		
	Juice WRLD	0.75	486.36
	Kanye	0.09	144.81
	Rihanna	0.01	105.19
	Acapella		
	Juice WRLD	0.03	410.39
	Kanye	0.23	248.51
	Rihanna	0.03	228.13
	Complete		
	Juice WRLD	0.17	304.84
	Kanye	0.22	158.66
	Rihanna	0.45	184.13
Fine-Tune	**Instrumental**		
	Juice WRLD	0.02	482.56
	Kanye	0.02	139.45
	Rihanna	0.11	127.64
	Acapella		
	Juice WRLD	0.03	407.02
	Kanye	0.15	229.13
	Rihanna	0.17	275.63
	Full Song		
	Juice WRLD	0.04	264.76
	Kanye	0.16	104.98
	Rihanna	0.02	146.86

5.4 Results

The table presents the objective metrics for evaluating the performance of audio generation models by comparing conditioning versus baseline and conditioning versus fine-tune across three categories: Instrumental, Acapella, and Complete, for three artists: Juice WRLD, Kanye, and Rihanna. The metrics used are KL-Divergence and Fréchet Audio Distance (FAD).

5.4.1 Instrumental

For the baseline comparison, Juice WRLD shows a high KL-Divergence of 0.75 and a high FAD of 486.36, indicating significant divergence and feature distance between the original and generated audio. Kanye and Rihanna display lower KL-Divergence values (0.09 and 0.01, respectively) and moderate FAD scores (144.81 and 105.19), suggesting closer alignment with the original audio compared to Juice WRLD. When comparing with the fine-tune results, fine-tuning reduces both the KL-Divergence and FAD for all artists. Juice WRLD shows substantial improvements (KL: 0.02, FAD: 482.56), Kanye exhibits a significant decrease in both metrics (KL: 0.02, FAD: 139.45), and Rihanna shows an increased KL-Divergence but a decrease in FAD (KL: 0.11, FAD: 127.64).

5.4.2 Acapella

For the baseline comparison, Juice WRLD and Rihanna have similar KL-Divergence values (0.03), while Kanye's KL-Divergence is higher (0.23). FAD is highest for Juice WRLD (410.39) and lower for Kanye and Rihanna (248.51 and 228.13, respectively). In the fine-tune comparison, fine-tuning leads to improved KL-Divergence and FAD for Juice WRLD (KL: 0.03, FAD: 407.02) and Kanye (KL: 0.15, FAD: 229.13). Rihanna's metrics show an increase in both KL-Divergence (0.17) and FAD (275.63), indicating less improvement compared to the baseline.

5.4.3 Full Song

In the baseline comparison, Juice WRLD has a KL-Divergence of 0.17 and FAD of 304.84, Kanye has a similar KL-Divergence (0.22) but a lower FAD (158.66), and Rihanna has the highest KL-Divergence (0.45) and a moderate FAD (184.13). Fine-tuning generally improves the metrics; Juice WRLD shows reduced KL-Divergence (0.04) and FAD (264.76), Kanye shows a slight increase in KL-Divergence (0.16) but a significant reduction in FAD (104.98), and Rihanna shows a significant reduction in both KL-Divergence (0.02) and FAD (146.86), indicating the most substantial improvement among the three artists.

Overall, fine-tuning appears to improve the alignment between generated and original audio across most metrics and categories, with varying degrees of effectiveness across different artists and conditions.

6 Analysis

When operating on acapella tracks, MusicGen demonstrates a strong ability to capture the tone and style of the singer or rapper's voice. The model effectively replicates the vocal timbre and general expressive qualities of the performer. However, it struggles with producing coherent singing or rapping. The generated vocals often lack the natural flow and intelligibility of real lyrics, resulting in vocal outputs that sound fragmented and nonsensical.

MusicGen performs exceptionally well on instrumental continuations. The model is adept at maintaining the rhythm and beat of the original instrumental tracks. In many cases, it even adds plausible musical elements that enhance the original composition.

When tasked with continuing full songs that include both vocals and instrumentals, MusicGen shows mixed results. The model does a commendable job of continuing the instrumental parts, preserving the musical structure and adding coherent musical elements. However, it struggles with the vocal components. While it can maintain the tonal quality of the vocals, the lyrics are often unintelligible and lack coherence, similar to the issues observed in acapella continuation.

7 Conclusion

Our findings demonstrate that the MusicGen model is highly effective in generating instrumental music, maintaining rhythm and beat while adding plausible and coherent musical elements. The model also shows a strong ability to capture the tone and sound of an artist's voice, accurately replicating vocal timbre and style. However, it struggles with producing coherent singing or intelligible lyrics, resulting in vocal outputs that often sound fragmented and nonsensical.

Despite these challenges, the improved KL-Divergence and Fréchet Audio Distance scores after fine-tuning indicate that it is possible to model acapella music without the need for semantic tokens. This suggests that with more extensive training resources, it may be feasible to achieve coherent vocal generation. The primary limitation of our work lies in the lack of training resources, which prevented us from fully exploring the model's potential to produce coherent singing. However, our results provide a promising signal that, given sufficient data and computational power, generative models like MusicGen can effectively model both instrumental and vocal music.

8 Ethics Statement

Generative models like MusicGen present several ethical challenges and societal risks that must be addressed. One of the most pressing concerns is the potential for these models to create unfair competition for human artists. With the ability to produce high-quality music quickly and inexpensively, these models could devalue the work of professional musicians and reduce their opportunities for income and recognition. This poses a significant risk to the livelihoods of artists who rely on their creative outputs for financial stability and career growth.

To mitigate these risks, we emphasize the development of generative models as tools that enhance the creative processes of artists rather than replace them. By positioning MusicGen as an aid in

the creative workflow, artists can use the technology to experiment with new ideas, streamline their production processes, and ultimately create higher-quality work. This approach aims to empower artists, allowing them to leverage the strengths of generative models to augment their own creative capabilities.

Another critical aspect of our mitigation strategy is the commitment to open research and transparency. By publishing our code, training data, and research findings, we ensure that all stakeholders, including independent and less-resourced artists, have equal access to these advanced technologies. This democratization of access helps to level the playing field, allowing a broader range of individuals and organizations to benefit from the advancements in generative modeling.

Beyond the immediate impacts on the music industry, the ability of MusicGen to synthesize high-quality audio with long-term coherent structure unlocks numerous positive applications. These include aiding individuals with speech impediments by providing realistic and natural-sounding speech synthesis, as well as assisting musicians in composing new pieces by generating inspirational audio segments. Such applications can significantly enhance the quality of life for individuals with specific needs and open new creative possibilities for musicians.

References

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. 2023. Musiclm: Generating music from text.

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

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

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A generative model for music.

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

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA.

Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer. 2022. Efficient training of audio transformers with patchout. In *Interspeech 2022*, interspeech$_2$022.$ISCA$.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. 2023. Audiogen: Textually guided audio generation.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2018. Neural discrete representation learning.

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

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.

Zalan Borsos, Rapha ´ el Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Shar- ¨ ifi, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. Audiolm: a language modeling approach to audio generation. arXiv preprint arXiv:2209.03143, 2022.