



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE  
CORSO DI LAUREA MAGISTRALE IN  
ICT FOR INTERNET AND MULTIMEDIA

Riffusion Meets Emotions: Deep Learning with Stable Diffusion  
for Emotionally Expressive Music Composition

Relatore: Prof. Antonio Roda

Laureando: Mohammad Mehdi Zare

ANNO ACCADEMICO 2023-2024

Data di laurea 25/11/2024

# Abstract

In this work, I present the fine-tuning of the Riffusion model using Dreambooth, guided by the DEAM (Database for Emotional Analysis of Music) dataset, to enhance emotion-based music generation. Using the advanced software frameworks and computational resources accessible via Google Colab, three distinct experiments were executed with several key hyperparameters such as spectrogram resolution, batch size, learning rate schedules and regularization methods being varied. The goal was to condition the model to synthesize spectrograms accurately corresponding to localized desired emotions, but retaining an overall high musical quality. This was validated by the experimental results which showed incremental gain in the stability of loss function and clarity of spectrograms after each configuration. In particular, I noted more stable convergence and less overfitting in the final experiment thanks to using a cosine learning rate scheduler and introducing weight decay. Nevertheless, several issues such as prominent noise artifacts, unstable loss curves and relatively small and unbalanced dataset prevented us from achieving purely high quality outputs. These results demonstrate the promise and challenges of using diffusion models for emotion-driven music composition. I conclude the study by highlighting several components that should be investigated in future work, such as larger data sets, increased computational resources, denoising capabilities and model architecture design for diffusion models to explore the true potential of creating emotionally convincing music.

Keywords:

*Emotion-Based Music Generation, Riffusion, Dreambooth, Diffusion Models, DEAM Dataset, Spectrograms, Machine Learning, Transfer Learning*

## Contents

1. Introduction.....	4
1.1. Background and Content.....	4
1.1.1 Generative Models and the Emergence of Stable Diffusion .....	4
1.1.2 Riffusion: Revolutionizing Music Generation via Spectrograms .....	5
1.1.3 Music and Emotional Resonance.....	5
1.1.4 Leveraging the DEAM Dataset for Emotion-Based Music Generation .....	6
1.1.5 Fine-Tuning Riffusion: Music Generation by Emotion .....	6
1.2. Motivations of Study .....	7
1.3. Problem Statement.....	7
1.4. Research Objectives.....	<b>Error! Bookmark not defined.</b>
2. Related Work.....	8
2.1. Music Transformer: Generating Music with Long-Term Structure [6] .....	8
2.2. Jukebox: A Generative Model for Music by Dhariwal et al.[7] .....	9
2.3. MuseGAN by Hao-Wen Dong et al.[8] .....	11
2.4. Pop Music Transformer by Yu-Siang Huang and Yi-Hsuan Yang [9] .....	12
2.5. From Artificial Neural Networks to Deep Learning for Music Generation by Jean-Pierre Briot[10] .....	14
2.6. The challenge of realistic music generation: modelling raw audio at scale by Sander Dieleman et al.[11] .....	15
2.7. WaveNet Autoencoders by Jesse Engel et al.[12] .....	16
2.8. MidiNet by Li-Chia Yang et al.[13].....	20
2.9. MusicLM: Generating Music From Text by Andrea Agostinelli et al. [14] .....	22
2.10. Latent Diffusion Models by Robin Rombach et al. [15].....	25
2.11. DreamBooth by Nataniel Ruiz et al. [18].....	28
3. Methodology .....	32
3.1. Introduction to methodology .....	32
3.2. Dataset Description.....	33
3.2.1 DEAM Dataset Overview .....	33

3.2.2	Data Preprocessing.....	34
3.3.	Model Architecture .....	37
3.3.1	Riffusion Model Overview .....	37
3.3.2	Fine-Tuning with Dreambooth .....	41
4.	Experiments .....	46
4.1.	Introduction to Experiments.....	46
4.2.	Software Frameworks and Libraries.....	46
4.3.	Hardware Specifications .....	47
4.4.	Experiment 1: Baseline Fine-Tuning .....	47
4.5.	Experiment 2: Enhanced Fine-Tuning with Increased Resolution and Batch Size .....	52
4.6.	Experiment 3: Advanced Fine-Tuning with Optimized Learning Rate Schedule and Weight Decay .....	57
4.7.	Summary of Experimental Findings .....	62
5.	Results .....	64
5.1.	Summary of Result .....	64
5.2.	Challenges and Limitations.....	66
6.	Conclusion .....	70
6.1.	Summary of Findings.....	70
6.2.	Future Work.....	70

Figure 1. Relative global attention .....	8
Figure 2. Multiscale VQ-VAE Encoding and Reconstruction Process .....	10
Figure 3: Three GAN models for generating multi-track data .....	12
Figure 4: Beat-Based Music Modeling and Generation Framework .....	13
Figure 5. Evolution of Neural Networks to Modern Deep Learning Models .....	15
Figure 6. Schematic overview of an autoregressive discrete autoencoder.....	16
Figure 7. Overview of Spectral and WaveNet Autoencoder Models .....	18
Figure 8. Note Reconstructions Across Instruments Using CQT Spectra.....	19
Figure 9. Comparison between recent neural network based music generation models. ....	20
Figure 10. System diagram of the MidiNet .....	21
Figure 11. User Study Results Comparing MelodyRNN and MidiNet Models by Musical Background.....	22
Figure 12. Training and Inference Workflow for MuLan-Based Audio Generation.....	24
Figure 13. Evaluation Metrics for Music Generation Models Using MusicCaps Dataset. ....	25
Figure 14. Conditioning Latent Diffusion Models via Concatenation or Cross-Attention. ....	27
Figure 15. Text-to-Image Synthesis Samples Generated by LDM-8 (KL) Model. ....	28
Figure 16. Fine-Tuning Diffusion Models with Subject-Specific Prompts.....	29
Figure 17. Applications of Novel View Synthesis and Property Modifications. ....	30
Figure 18. Preprocessing Pipeline.....	37
Figure 19. Training Workflow .....	45
Figure 20. Training parameters of experiment 1.....	47
Figure 21. Training loss curve of experiment 1. ....	51
Figure 22. Generated spectrogram with “Happy” prompt .....	51
Figure 23. Generated spectrogram with “Sad” prompt.....	52
Figure 24. Training parameters of experiment 2.....	53
Figure 25. Training loss curve of experiment 2. ....	56
Figure 26. Generated spectrogram with “Calm” prompt. ....	56
Figure 27. Generated spectrogram with “Sad” prompt.....	57
Figure 28. Training parameters of experiment 3.....	58
Figure 29. Training loss curve of experiment 3. ....	61
Figure 30. Generated spectrogram with “Happy” prompt .....	62
Figure 31. Generated spectrogram with “Tense” prompt.....	62

# Chapter 1

## 1. Introduction

### 1.1. Background and Content

#### 1.1.1 Generative Models and the Emergence of Stable Diffusion

The past few years have seen generative models drastically alter many areas and fields, but perhaps most of all, the more creative domains concerned with the generation of art, music, and language. Chief among them, until now, is Stable Diffusion, owing to the fact that it achieves highly detailed, high-quality outputs through progressive refinement of noise. Whereas the previous models, such as Generative Adversarial Networks and Variational Autoencoders, mostly suffered from problems including instability, an inability to capture fine detail, Stable Diffusion relies on an iterative process of denoising, where noisy data is refined through successive steps into coherent and realistic results.

The core of Stable Diffusion is to learn a series of probabilistic transitions that progressively denoise a latent representation. It would typically be initialized by a noise vector, which in each step is gradually transformed into a data sample, say an image or spectrogram. The model shall learn a reverse of the forward diffusion process—a process where noise is progressively added to the data samples. Through this reverse process, structured and detailed outputs are created:

$$p_{\theta}(x_{t-1}|x_t) \approx \mathcal{N}(\mu_{\theta}(x_t, t), \Sigma_{\theta}(x_t, t))$$

where  $p_{\theta}(x_{t-1}|x_t)$  is the probability distribution that transforms a noisy state  $x_t$  towards a less noisy state  $x_{t-1}$  at each iteration, parameterized by  $\theta$ . Iterative processes allow it to model both large structures and fine details with high accuracy.

Applications of Stable Diffusion range from generating images, synthesizing text-to-image, up to creating artistic content. Its flexibility has also opened ways for innovations in music and audio generation, beyond traditional image-based domains.

### 1.1.2 Riffusion: Revolutionizing Music Generation via Spectrograms

Riffusion is one of these newer applications built on Stable Diffusion principles for music generation. In contrast to prior music generation models, which operate either on raw audio waveforms or symbolic formats like MIDI, Riffusion takes a different approach: it represents music as mel-scaled spectrograms—a sort of visual representation of sound frequencies across time.

Mel-Scaled Spectrograms as a Medium for Music Generation:

A spectrogram displays graphically the intensity of a host of different frequency components of an audio signal that vary over time. The mel scale is tuned to reflect human auditory perception, therefore it is especially suitable in musical representation.

Consider that transformation of raw audio  $x(t)$  to a mel-scaled spectrogram mainly consists of two steps: a Short-Time Fourier Transform which changes signals from the time domain to the frequency domain, followed by the transformation of a mel-filterbank that perceptually scales the frequencies. This can be represented as:

$$S_{mel}(t, f) = M \cdot |STFT(x(t))|$$

where  $S_{mel}(t, f)$  is the mel-scaled spectrogram,  $M$  is the mel-filterbank matrix, and  $|STFT(x(t))|$  denotes the magnitude of the Short-Time Fourier Transform of the audio signal. Riffusion works on the premise that mel-scaled spectrograms are images, hence positioning it to best take advantage of Stable Diffusion's capabilities in image synthesis. It synthesizes new spectrograms by an iterative process of denoising, where the output converts into sound with complex harmonies and rhythm. This bridges the gap between generative methods centered around images and those of music synthesis, presenting a new paradigm for music creation.

### 1.1.3 Music and Emotional Resonance

Everyone knows that music evokes feelings and is a forceful means for expressing emotions. Whether a part of an ongoing tradition or a new composition, music is based on communication; it is used in ritual and prayer; words and sounds are sung to express personal. Indeed, it is hard to pin down the nature of the relationship between music and emotion, given that music can play a part in both experiencing emotions and demonstrating feelings. This interplay is often observed through the circumplex model of affect, which illustrates emotions along two major axes [4]–[5]: valence (from positive to negative) and arousal (from low to high intensity).

- Valence describes the positive or negative feeling of an affective experience.
- Arousal looks at the strength or level of energy (or intensity) that comes with the emotion.

As a two-dimensional model, it offers an organized schema of the emotional experience, just perfect when designing AI systems for composing music capable of sounding emotionally weighed. Thus, higher arousal (e.g., excitement) could hypothetically match an energetic, upbeat tempo composition while lower arousal (e.g., sadness) might reflect a slower, more melancholic piece [3].

#### 1.1.4 Leveraging the DEAM Dataset for Emotion-Based Music Generation

DEAM dataset (Dataset for Emotional Analysis of Music) is an important resource to capture emotional modeling for generation music written by emotion. The DEAM dataset contains a variety of annotated songs with continuous valence and arousal values, which creates a strong basis for training AI models to generate music that corresponds to different emotional states. While capturing transitions and mixed states in human emotions may be challenging given only discrete labels from categorical datasets (where the label space inherently lacks the emotional granularity needed for fine-tuned emotion modeling), the DEAM continuous annotations provide sufficient continuous signal that captures this complexity well [4].

Fine-tuning Riffusion with DEAM dataset would enable the model to learn mappings between mel-scaled spectrograms and their related sentiments/emotions states. It's using transfer learning, which means it uses some of the parameters of a pre-trained model adapted to a specific task. Transfer learning is useful as it allows us to learn faster and demands lesser data than building a model from scratch which is very significant for projects like emotion-based music generation [2][4][5].

#### 1.1.5 Fine-Tuning Riffusion: Music Generation by Emotion

For emotion-based music generation, it is needed to fine-tune a deep learning model like Riffusion over a specific dataset, i.e. DEAM[3] dataset with continuous values of valence (Positivity/Negativity) & arousal (Energy/Intensity). Transfer learning is the basis of the fine-tuning process, allowing Riffusion to keep its generative music abilities while simply learning how to tie created pieces of music with specific emotional contexts. This trick saves the training time and resource needed and improves model specialization.

##### Fine-Tuning Process on DEAM Data

Fine-tuning aims to teach Riffusion to generate spectrograms corresponding to specific emotions as defined by DEAM dataset annotations. Specifically, this means taking inputs (random noise passed through the diffusion) and mapping them to outputs that have certain emotional characteristics. This is an optimization problem in which the model's parameters are iteratively updated to reduce the distance between predicted emotional states and ground-truth annotations from DEAM. The optimization is directed by a loss that quantifies the discrepancy between generated and intended emotional expressions [4].

The mel-scaled spectrograms the Riffusion model operates over embody complex acoustic features including rhythm, harmony and dynamics, features that convey much of music's emotional essence. Through fine-tuning the model on these spectrogram representations and their respective valence-arousal values, Riffusion learns to differ its outputs not just based on structural and stylistic normalities but also through emotional currents. Ability for such is of assistance to applications like music therapy, custom music playlist generation and adaptive soundscapes in gaming and immersive media [3][5].

##### Advantages of Music Generation by Emotions

This AI music generation based on emotions can transform multiple sectors by offering experiences that are truly personal to every user. For example:



- Music Therapy: Songs can also be individualized based upon patient's mood to assist with mental health or well-being interventions.
- Adaptable Gaming : Emotionally responsive soundtracks respond not only to in-game events, but also to player emotion and engagement.
- Personalized Playlist: Mood-based playlists which adapt according to the emotional condition of a listener provides a great personalization and engagement experience [5].

Generating music based on particular emotions provides new frontiers for creating content that can be both contextually and emotionally relevant

## 1.2. Motivations of Study

AI models still struggle to produce music that has a precise relation with or triggers certain emotions. Although previous systems such as Riffusion are good at generating musically coherent and stylistically diverse pieces, they do not always generate music that consistently elicits emotion. This constraint is a telling one, as music's strength is not just limited to its play on structure and groove — but rather in evoking an emotional connection with the listener. This would translate to AI generating music where emotion is the long-term catalyst making a difference between applications including entertainment, therapy and much more [2][3]. DEAM, with its continuous valence-arousal scores is a candidate dataset for creating such boundaries. This dataset allows Riffusion to be fine-tuned, and provides greater specificity in the affect of generated music. This ability enables to facilitate user interaction, provide individualized experiences and explore new creative possibilities in AI-based music composition.

## 1.3. Problem Statement

While state-of-the-art generative models like Riffusion are pretty good at generating musically coherent music, they have little ability to effectively condition their output on emotional contexts. This restriction makes them less well-suited for context-sensitive tasks that involve emotional nuance, including therapy applications, adaptive entertainment, and tailored music experiences. Fine-tuning Riffusion on the DEAM dataset to boost its ability to generate music that corresponds with desired emotional states is an important step in bridging this gap

# Chapter 2

## 2. Related Work

### 2.1. Music Transformer: Generating Music with Long-Term Structure [6]

The primary objective of the study to use knowledge about other epochs of music to achieve far longer coherent stretches than had previously been possible. This suggests the authors hoped to use the Transformer model — established in natural language processing for its effectiveness — to manage musical composition complexity. They wanted to modify it so that it would better recognize the cyclicity and hierarchical structure which are central features of music.

In this work, the authors added a mechanism to the Transformer model as means of tackling long-term coherence in music generation, which essentially is done by relative positional self-attention. It is different from the vanilla self-attention which you use absolute positional encodings. It instead think about the relative positions or distances between notes in a sequence, which is more suitable for how some music compositions are composed—for example, those that deal with repeating motifs or themes in different temporal scales.

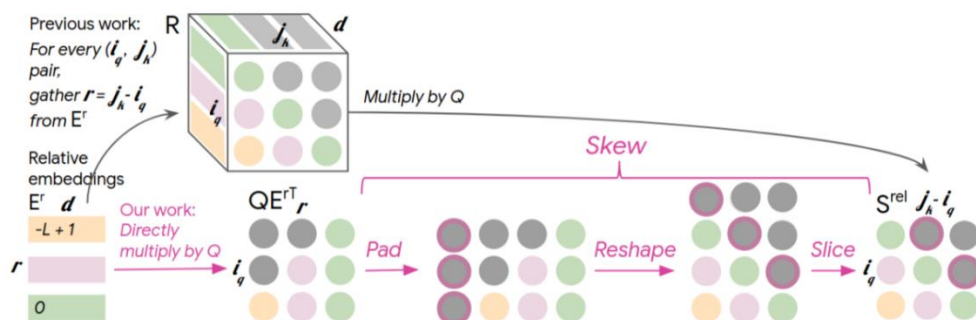


Figure 1. Relative global attention

The bottom row describes our memory-efficient “skewing” algorithm, which does not require instantiating  $R$  (top row, which is  $O(L^2D)$ ). Gray indicates masked or padded positions. Each color corresponds to a different relative distance. Image taken from [6].

One of the main technical contributions of their work was to implement relative attention in a more memory-efficient manner. The quadratic memory complexity in input sequence length of traditional self-attention become prohibitive for longer sequences typically found in music. Huang et al. suggested a solution which reduces memory usage from quadratic to linear against the length of sequence and thus enable longer sequences in musical compositions

The proposed model was evaluated on two datasets: 1) JSB Chorales and 2) Piano-e-Competition dataset. The music of these datasets are intricate, spanning both melody and accompaniment across long timescales which require coordination between the two parts.

The Music Transformer has outperformed previous models in their result, especially in the case of generating an important section of music with coherent and extended structures. The Transformer achieved state-of-the-art results on the Piano-e-Competition dataset, proving that it can indeed process very expressive and challenging piano performances. It produced minute-long pieces with convincing internal structure and could have continued given motifs in a way that was deemed more musical in listening tests.

This attention mechanism proved critical in improving the performance of the model, allowing it to preserve a regular timing grid characteristic of music, generating pieces more consistent both harmonically and rhythmically.

Though, the Music Transformer was a big step forward, it still had its limitations:

- The computational complexity and resource demands (although improved in terms of memory requirements) are really high and can limit its actual use on less powerful systems or for real-time applications.
- Generalizability to Other Types of Music: The majority of the literature reviewed relates to classical and chorale music. However, the generalizability of the model on other kinds of music in terms of structural properties between genres, namely jazz or pop was not systematically tested as well.
- Musical Constraints: The truth is, the model could have written music that works on a technical level; however, can it compare to what an emotional human composer brings to musical composition?

## 2.2. Jukebox: A Generative Model for Music by Dhariwal et al.[7]

I present here a new generative model that enables the creation of high-fidelity and diverse music pieces containing singing, all in raw audio format. The other possible bottleneck is the long context of raw audio, which makes it extremely difficult to generate coherent music for minutes. However, they wanted to create music specifically conditioned on artist, genre and even lyric data that would provide more control over the desired style and subject matter of the music being generated.

They compress raw audio into discrete codes with a hierarchical Vector Quantized Variational AutoEncoder (VQ-VAE) and model these codes using autoregressive Transformers. By doing so, they could tackle the ultra-high dimensionality of raw audio. Stacked layers are used in the system, with a bottom layer that captures primitive audio features and top layers encoding higher level musical representations.

VQ-VAE Architecture: The model employs a hierarchical approach with several layers of VQ-VAEs to embed audio into progressively higher-level representations. Enabling efficient compress and synthesis of audios.

Sparse Transformers, which model the conditional distributions of these codes to create new music, are then fed the discrete codes from encoding in an autoregressive fashion.

Conditioning Mechanisms: They performed artist and genre conditioning to control music generation, and lyric conditioning to make the generated singing sync with the provided lyric text.

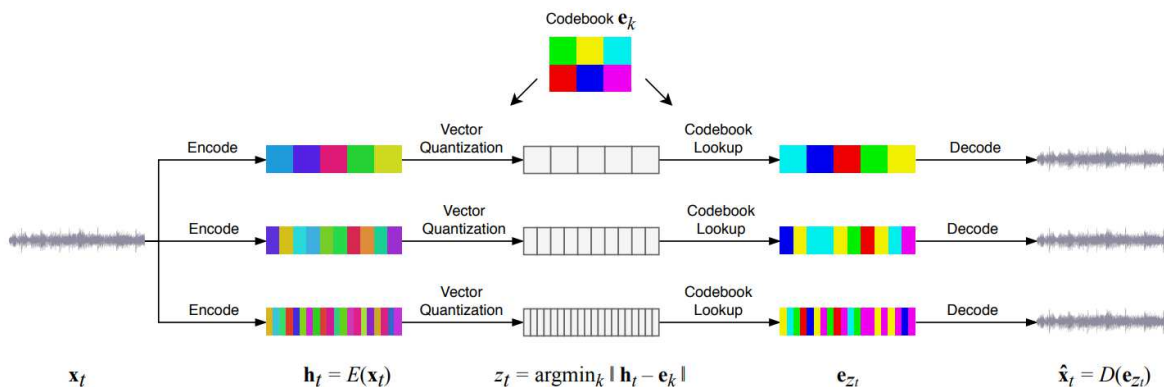


Figure 2. Multiscale VQ-VAE Encoding and Reconstruction Process

Training three different VQ-VAE accounts with different time-scale resolutions, as shown in Figure The input audio is divided into segments at each level, and the segments are encoded to latent vectors ( $h_t$ ), which are then quantized to the nearest codebook vectors ( $e_{z_t}$ ). The quantized vectors,  $z_t$  are a compact discrete representation of the audio and will be used for training the autoregressive priors. The decoder recomposes the audio, using this sequence of codebook vectors. The model reaches the highest abstraction in the top level, which encodes longer audio per token while keeping a fixed codebook size. With this setup, you can reconstruct audio at any level of abstraction and the least abstract bottom-level codes create the best quality audio. Image taken from [7].

Jukebox with which the model was able to produce musically reasonably coherent songs with understandable singing in different music genres (e.g., rock, hip-hop, jazz). It produced music that retained melody, rhythm and timbre quite successfully, with a notable increase in the diversity and fidelity when compared to prior models. The model was also able to generate completions of previously-existing songs, allowing for a degree of control over the generation process through the provided conditions.

- Quality: Diversity of the songs produced remained high, and they were sufficiently realistic over multiple minutes. It was also observed how well the model could accommodate various instrumentation including vocals.
- Impact of Conditioning: As it had already done in on lyrics and MIDI to have a substantial control of the musical attributes, such conditioning seemed to work here too leading to a more controlled singing voice that matched well with the target music style category.

Although the model is a major step forward in music generation, some limitations are observed:

- Computational Intensity: It is a computationally intensive process, especially because audio generation has long-range dependencies. This can be limiting in practical usage of the model in real-time applications.
- Model Complexity: The complexity of the model and requirement of large data for training makes it difficult to have adequate resources available in order to make them efficient and scalable.
- Creative Constraints: While it does have potential, there still a gap in creating human emotional depth and variation across music creation.

### 2.3. MuseGAN by Hao-Wen Dong et al.[8]

The paper introduces MuseGAN, a model that can sustain generating multi-track polyphonic music (whereas most prior work limited their attention to symbolic music generation within one single track). This model would help in generating music with harmonic and rhythmic structures along with inter-track dependencies and a global temporal structure while dealing with the complexities of polyphonic textures and the interactions between multiple musical instruments.

MuseGAN is built around Generative Adversarial Networks (GANs) composed of 3 primary models to solve for different music-gen situations:

- Jamming Model: Multiple independent generators produce their own track without being scored into a single prearranged piece, implementing a concept of musical improvisation.
- Composer Model: One generator generates all tracks at once, simulating a single composer composing an entire orchestration piece.
- Hybrid Model: Exactly what it sounds like — combines the two above models to mix consistent generated features shared across tracks while also allowing diversity on a track-by-track basis.

They give fundamental thinking as them bars rather than an individual note. They think there is a good reason to believe that transposed convolutional neural networks can be used to discover these local, translation invariant patterns in music.

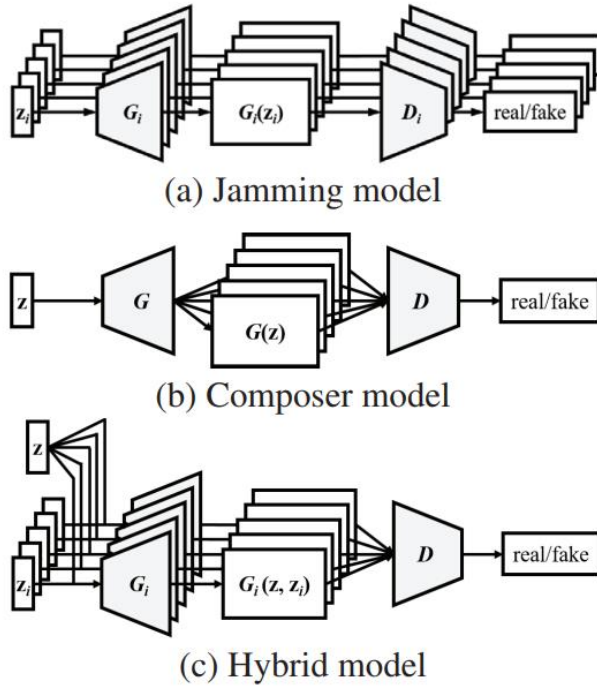


Figure 3: Three GAN models for generating multi-track data

Note that they do not show the real data  $x$ , which will also be fed to the discriminator(s). Image taken from [8].

Using MuseGAN, they generated coherent four-bar music sequences in five different tracks: strings, drums, bass, guitar and piano. This also allowed for human-AI collaboration in composing music, where the AI could finish a piece started by a human. Our evaluation of MuseGAN consists of objective metrics for intra-track and inter-track qualities and a user study with 144 participants, which together demonstrate its capacity to generate harmonically and rhythmically satisfying music.

However, MuseGAN is not without its limitations;

- Computational Demand: GANs are quite complex and computationally extensive to train, which may hinder larger experiments or applications in real-time.
- Musical Creation: MuseGAN can create competent music, but the creativity and emotions that a human composer presents remain unfilled.
- Evaluation of Cross-Genre Generalization: The generalization ability of the model across many genres and other styles not so prevalent in the training data has yet to be evaluated.

## 2.4. Pop Music Transformer by Yu-Siang Huang and Yi-Hsuan Yang [9]

The model present a pop music transformer in the study to improve their expressivity, and propose REMI (REvamped MIDI-derived events), a new data representation that utilizes beat

templates as binary inputs/outputs to generate polyphonic pop piano music. It focuses on enhancing the rhythmic and harmonic structure of generated music so that it corresponds more closely to the beat-bar-phrase hierarchical composition present in human-generated music.

The researchers used REMI to adapt the standard Transformer model, i.e., they started from a traditional Transformer structure and were able to feed metrical information directly into the data being passed to the Transformer. Contains explicit bars and bar-position-in-bar information, allowing the model to better grasp the metrical structure of music. The methodology involves:

- Beat-level Describing Data: Introducing a beat-bar level hierarchy directly in the music representation on which it works, so that a better model can learn and regenerate the music with accurate rhythmic features.
- Transformer-XL: Using the Transformer-XL model, which add continuous learning induction to a standard Transformer — attempting to learn longer sequence context provides, thus critical for maintaining musical coherence over larger pieces.

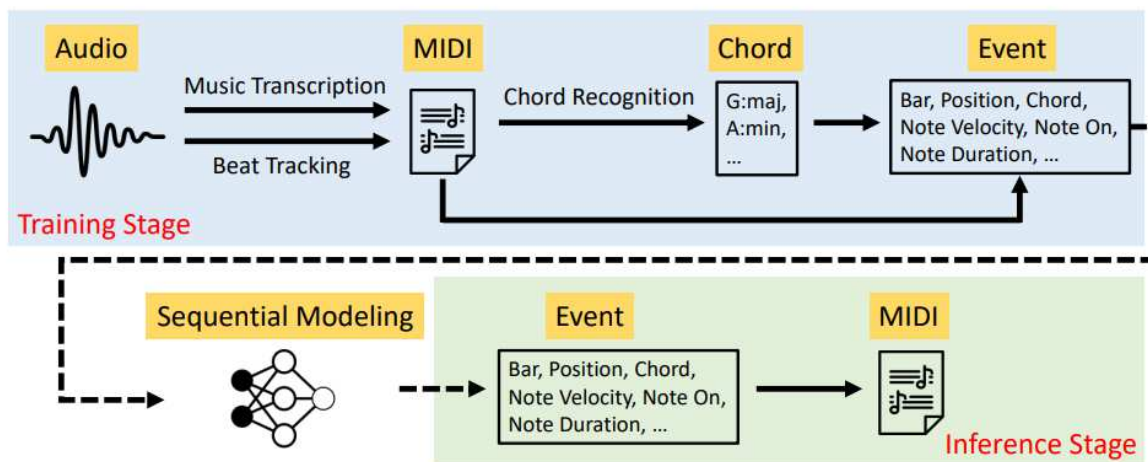


Figure 4: Beat-Based Music Modeling and Generation Framework

The diagram of the proposed beat-based music modeling and generation framework. The training stage entails the use of music information retrieval (MIR) techniques, such as automatic transcription, to convert an audio signal into an event sequence, which is then fed to a sequence model for training. At inference time, the model generates an original event sequence, which can then be converted into a MIDI file for playback. Image taken from [9].

Pop Music Transformer generated pop piano music that shows improved rhythmic structure and musical context over previous state-of-the-art models for Transformers. Through both objective metrics and user studies, it was shown to be able to synthesize human-like metrical and rhythmic structure into music:

- Objective Evaluation: The model improved on keeping the beat and bar structure throughout generated pieces more consistent.
- Subjective Evaluation: The Pop Music Transformer was preferred by listeners in a perceptual study compared to music generated from baseline models, with improved timing accuracy and musicality.

Though there has been a great deal of improvement over music generation models from the past, there are some constraints with the model as well:

- Complexity and Computation: High Complexity and Computational Demand of the model can limit its use in real-time.
- Genre Transitioning: The model in its current form is more fine-tuned to pop music and whether or not those same learned features can be transitioned effectively to other genres or styles of music has yet-to-be seen.
- Diversity in creativity: Although models have evolved, the creative diversity and emotional experience of music created by an artist is still something that is missing

## 2.5. From Artificial Neural Networks to Deep Learning for Music Generation by Jean-Pierre Briot[10]

The paper provides an extensive review and tutorial of the history, development and use of deep learning based techniques for music generation. This article investigates how neural networks evolved from individual nodes that simply make decisions to the layering of multiple levels such that deep learning models are able to compose music, specifically tracing the historical progression of this technology and thereby illustrating its metamorphosis from systems designed for perceptive pattern recognition into systems that produce art.

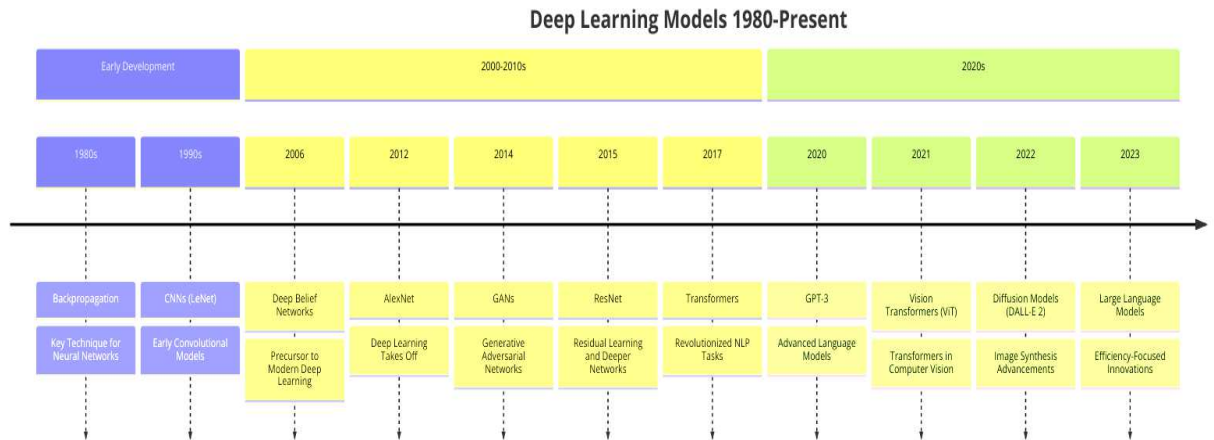
Methodology The paper draws on historical research in combination with a review of recent literature. It delves into:

- Early Experiments: Featuring works from the 1980s that utilized neural networks to compose music.
- Conceptual Frameworks: Outlining a way to view and categorize the various types of deep learning models employed in music making.
- Case Studies: As this section contains many case studies, It describes several contemporary systems that have exemplified the different approaches taken in deep learning based music generation since they diverge from simpler neural network architectures to more sophisticated frameworks based on GANs (Generative Adversarial Networks) and VAEs (Variational Autoencoders).



Figure 5. Evolution of Neural Networks to Modern Deep Learning Models

Diagram illustrating the transition from early neural networks to advanced deep learning models today. Image generated using AI



The paper synthesizes a large body of research to provide an in-depth understanding of the impact of deep learning on music generation. Key findings include:

- **Successful Applications:** Showing that deep learning models learned music styles from the data and were able to create new music that is consistent with the style.
- **Evolving Methods:** Emphasizing the progression from rudimentary models that could create simple melodies to advanced systems that were able to generate complete pieces with intricate forms.
- **Interactivity and Control:** Talking about current capabilities of deep learning models to interact with users dynamically, where you can provide controls/input signals for the model to generate music according to your needs, allowing new ways of creative musical expression.

## 2.6. The challenge of realistic music generation: modelling raw audio at scale by Sander Dieleman et al.[11]

This paper addresses the problem of generating realistic music in raw audio, as opposed to high-level symbolic representations such as MIDI. The goal is to generate generative models that can output music which incorporates the quirks of individual performances, as given these are necessary for realism. To solve these issues, in this work, the researchers intend to develop a model that not only can represent long-range correlations between audio signals but also works on raw audio data instead of spectrograms like previous models which were mainly short-term.

To better leverage the long-range correlations present in musical waveforms, the authors investigate autoregressive discrete autoencoders (ADAs). The methodology includes:

- Autoregressive Models: Using complex autoregressive models that directly model audio waveforms, enabling the generation of high-fidelity music.
- Autoregressive Discrete Autoencoders (ADAs): This further advances the modeling of music structure across a range of timescales, which is one of the main challenges in generating raw audio music.

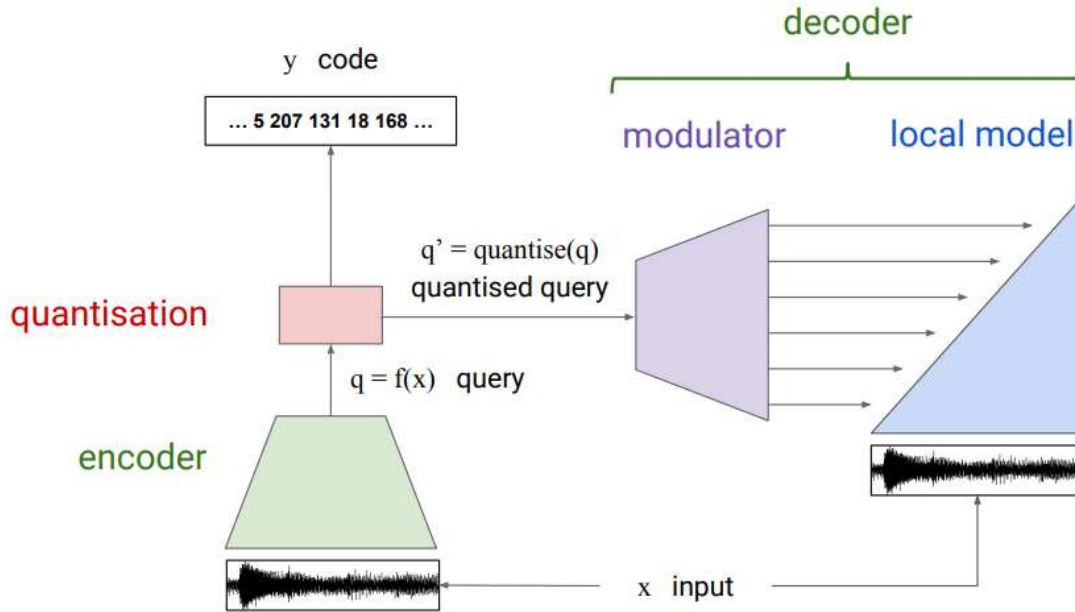


Figure 6. Schematic overview of an autoregressive discrete autoencoder

The encoder, modulator and local model are neural networks. Image taken from [11].

#### Limitations:

- Computational Demand: Due to the inherent complexity of modeling raw audio signals on a large scale, these models come with a high computational cost.
- Bias Toward Short-term Dependencies: the models have a strong inclination to consider only the local signal structure. This is not too much of an issue for most types — but a limit it creates on longer shaped musical style
- Balance of Musicality and Fidelity: I might lose some fidelity in the signal (which is not our primary goal) while gaining not only musical but also sonic signature enough to know by ear when something comes instead of other.

## 2.7. WaveNet Autoencoders by Jesse Engel et al.[12]

This paper presents a new method of audio synthesis by leveraging a WaveNet-style autoencoder to alleviate some of the limitations inherent in direct modelling musical notes from raw audio (eg. loudness-related distortion or lack of expression). The objectives are twofold:

- Create a neural network audio codec, also known as autoencoder to learn temporal embeddings of audio data for synthesizing new high-fidelity musical sounds that does not require conditioning signals
- Introducing NSynth, a large scale public dataset of musical notes, purpose-built for training and evaluating neural audio synthesis models.

With this motivation, Engel and his colleagues then started implementing a sophisticated neural network architecture inspired by the big progress brought to speech generation with WaveNet. The methodology involves:

WaveNet Autoencoder Architecture: The autoencoder is implemented using a deep convolutional neural network with dilated convolutions, which progressively increases the receptive field of the model to learn audio over longer temporal horizons. The architecture consists of mainly two components:

- Encoder: It takes the raw audio and compresses it a learned temporal embedding that contains sinusoidal features of the input.
- Decoder: Uses the embeddings to reconstruct the audio, trying to be as similar to the original input as possible but also allows for generating new sounds based on the manipulated embeddings.

The resource is a dataset called NSynth, which contains around 300,000 musical notes based on over 1,000 unique sounds to help developers easily train the model. The dataset provides an annotation for each note (pitch, velocity etc.) and hence serves a uniform range of systematic evaluation and comparison benchmarks of various audio synthesis models.

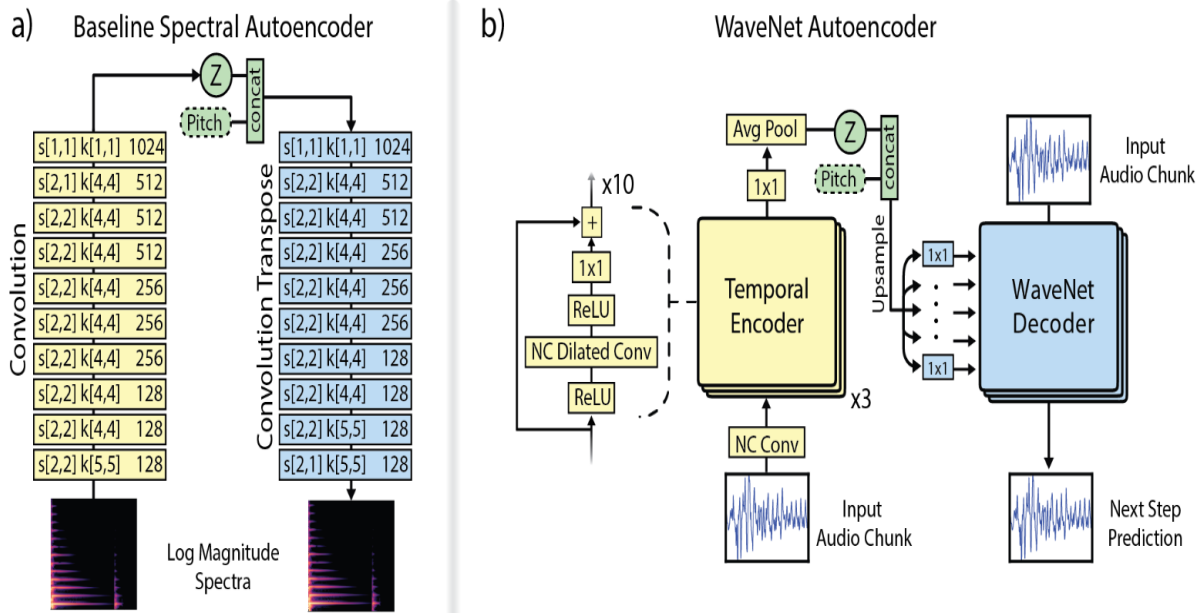


Figure 7. Overview of Spectral and WaveNet Autoencoder Models

Models considered in this paper. For both models, we optionally condition on pitch by concatenating the hidden embedding with a one-hot pitch representation. 1a. Baseline spectral autoencoder: Each block represents a nonlinear 2-D convolution with stride ( $s$ ), kernel size ( $k$ ), and channels ( $\#$ ). 1b. The WaveNet autoencoder: Downsampling in the encoder occurs only in the average pooling layer. The embeddings are distributed in time and upsampled with nearest neighbor interpolation to the original resolution before biasing each layer of the decoder. ‘NC’ indicates non-causal convolution. ‘1x1’ indicates a 1-D convolution with kernel size 1. Image taken from [12].

Several critical results were revealed through the implementation of the WaveNet autoencoder:

- **Enhanced Fidelity:** The audio produced by the WaveNet autoencoder had higher fidelity as well as better representation of musical nuances (i.e. timbre and dynamics) compared to baseline models.
- **Expressive Interpolations:** The model was also able to interpolate between sounds and generate new notes that shared properties of two distinct instruments in a coherent and musically plausible way.
- **Analysis of the embeddings:** Embedding analysis generated by the encoder showed them to carry a rich and meaningful representation of the audio, aligned with human perception of musical characteristics.

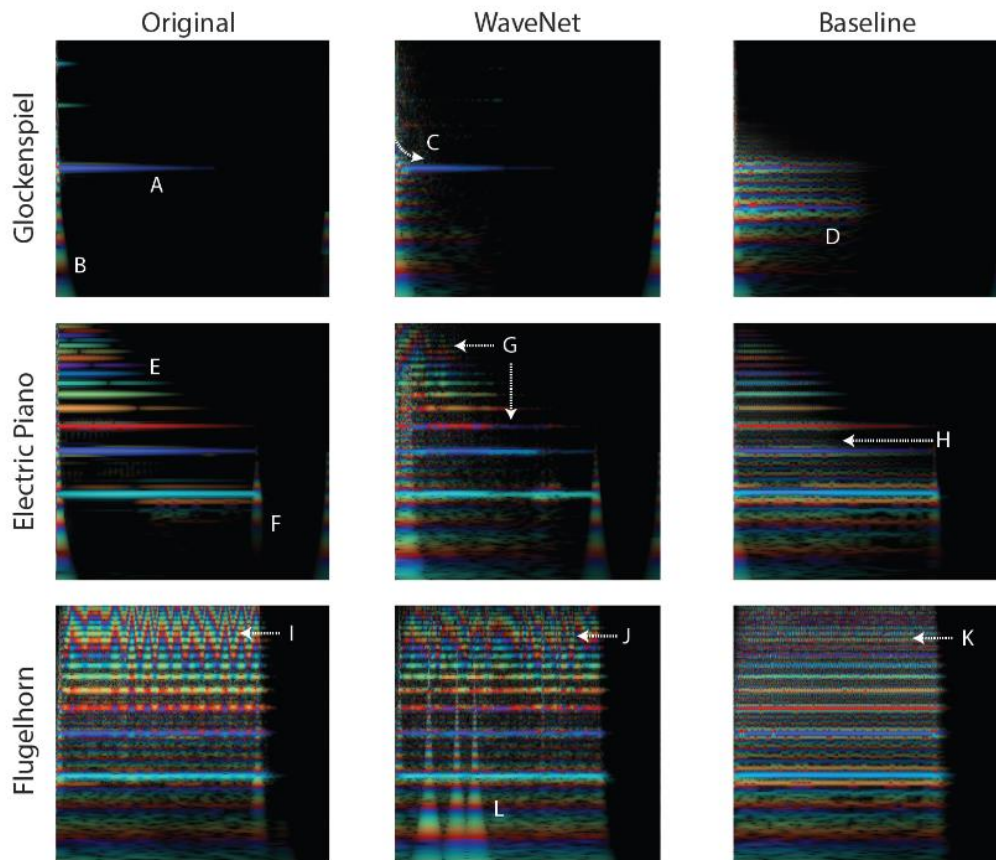


Figure 8. Note Reconstructions Across Instruments Using CQT Spectra

Reconstructions of notes from three different instruments. Each note is displayed as a CQT spectrum with time on the horizontal axis and pitch on the vertical axis. Intensity of lines is proportional to the log magnitude of the power spectrum and the color is given by the instantaneous frequency. See Section 4.1 from the original paper for details. Image taken from [12].

There are few Limitaions too:

The study represents a considerable step forward in neural audio synthesis, but it also comes with a number of limitations:

- **High Computational Requirements:** The WaveNet model is computationally intensive due to its architecture, making it less feasible for real-time applications or deployment on low-power devices.
- **Training Challenges:** The model's training is highly sensitive to the choice of hyperparameters, and the autoencoder can be prone to overfitting, especially given the high variability in musical content across the NSynth dataset.
- **Generalization Concerns:** The fact that the model generalizes well to other audio types (that are not wellrepresented in NSynth such as non-Western musical instruments, environmental sounds etc.) is an open question.

## 2.8. MidiNet by Li-Chia Yang et al.[13]

This paper presents MidiNet: A new model of generative adversarial network (GAN) that uses convolutional neural networks (CNNs) to produce melodies in the symbolic-domain. While many models use Recurrent Neural Networks (RNNs) to build melodies one note at a time, MidiNet uses CNNs and builds the melody bar by bar, which takes less time to train and can be parallelized with relative ease. The goal was to develop a model that could:

- Creating melodic content either from scratch or conditioned on chord sequences or prior melody.
- Increasing the diversity and creativity of rendered music due to the model's ability of generating new melodies based on existing bars or external music data.

	MelodyRNN [33]	Song from PI [7]	DeepBach [15]	C-RNN-GAN [21]	MidiNet (this paper)	WaveNet [31]
core model	RNN	RNN	RNN	RNN	CNN	CNN
data type	symbolic	symbolic	symbolic	symbolic	symbolic	audio
genre specificity	—	—	Bach chorale	—	—	—
mandatory prior knowledge	priming melody	music scale & melody profile	—	—	—	priming wave
follow a priming melody	✓	✓			✓	✓
follow a chord sequence					✓	
generate multi-track music		✓	✓		✓	✓
use GAN				✓	✓	
use versatile conditions					✓	
open source code	✓			✓	✓	

Figure 9. Comparison between recent neural network based music generation models.

Table taken from [13].

MidiNet is a GAN for symbolic music that consists of two CNN-based networks, the generator and discriminators. Most notably the model represents melodies in 2-D matrices, much like a musical score and utilizes several new approaches:

- Generator and Discriminator CNNs: The generator transforms random noise into music bars while the discriminator assesses their realism. Transposed convolutions are great to allow the generator to upscale from low-dimensional noise to the full matrix representation of a music bar.
- Conditional Mechanism: It can add some knowledge, e.g., information about the chords or a melody crafted before. This is done by conditioning on additional CNN to better the logical musical progression over bars in a way.

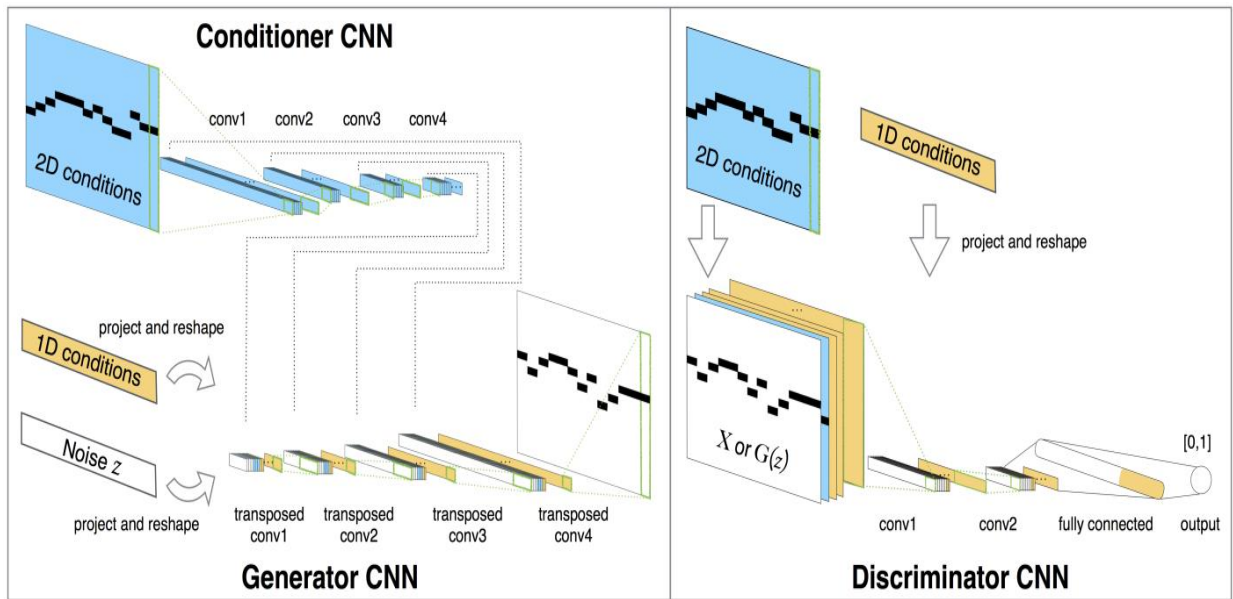


Figure 10. System diagram of the MidiNet

model for symbolic-domain music generation. Image taken from [13].

The results of MidiNet were evaluated through a user study comparing its output with that of Google’s MelodyRNN. In the study they evaluated the produced music on its realism, pleasantness and interest. Key findings include:

- Performance: MidiNet was comparable with MelodyRNN when it came to producing realistic and enjoyable music, but significantly better at generating melodies that were more interesting or diverse.
- Creativity and Flexibility: The model showcased some extent of creativity, as it produced different melodies by using the same priming melody by simply changing random noise input.

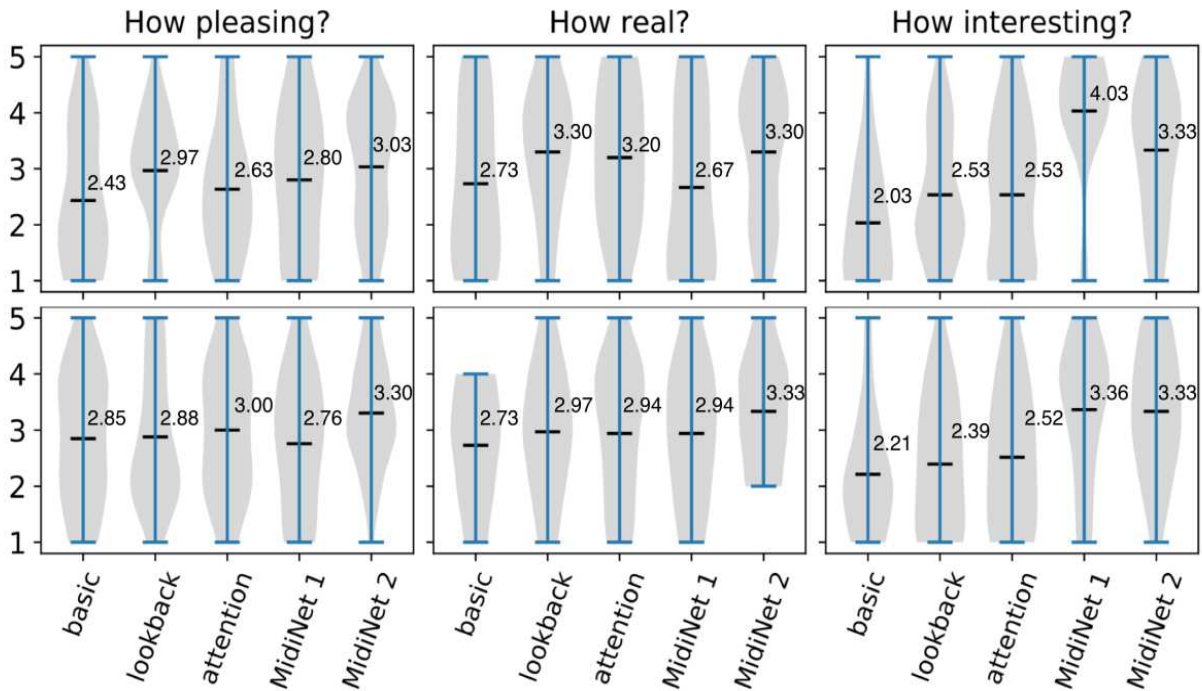


Figure 11. User Study Results Comparing MelodyRNN and MidiNet Models by Musical Background.

Result of a user study comparing MelodyRNN and MidiNet models, for people (top row) with musical backgrounds and (bottom) without musical backgrounds. The middle bars indicate the mean values. Please note that MidiNet Model 2 takes the chord condition as additional information. Image taken from [13].

Although MidiNet is a major step for the use of CNNs in symbolic music generation, the authors mentioned some limitations:

- **Model Complexity:** Although CNNs help avoid this hassle, setting up and running the model becomes complex especially for GAN frameworks.
- **Reliance on Conditioning:** The quality of the generated output heavily relies on the conditionings. It may not allow novel compositions merely based on no conditioning at all.
- **Generalization:** The ability of the model to generalize was not well examined, as it was limited to one type of music, thus limiting its use case for different kinds of musical tasks.

## 2.9. MusicLM: Generating Music From Text by Andrea Agostinelli et al. [14]

This paper presents MusicLM, a model which is able to synthesize high-fidelity music from text descriptions. The main goal is to solve the problem of converting text into high level linguistic structures like audio with long term consistency. The goal of MusicLM is to create music that



closely follows the given descriptions, paving the way towards conditional neural audio generation that goes beyond the current state of the art.

MusicLM employs a hierarchical sequence-to-sequence framework that extends on AudioLM, a previous audio synthesis model. They use multi-stage autoregressive modeling approach:

Quantization and Tokenization:

MusicLM uses a method known as quantization which essentially takes raw audio and converts it into a form that can be handled by a neural network. To do this, continuous audio signal now is turned into a sequence of discrete tokens that can be processed by neural architectures. These tokens denote snippets of audio and their discretized form reduces the cost of processing continuous streams of audio.

- **SoundStream:** A unique element of MusicLM's quantization is SoundStream, an audio codec for high-fidelity compressed audio generation. As tokens are generated, they are consumed by the model which synthesises the waveform back from those tokens. SoundStream plays a key role to keep the audio fidelity as high as possible between tokenization and synthesis process which is important for good quality music outputs.

Hierarchical Modeling:

In order to handle the complexity of producing long outputs of music that make sense, MusicLM uses a two-level modeling approach. It processes audio at various time scales, capable of covering everything from local note transitions to global musical phrasing and structure.

- **Multi-Scale Representation:** The model produces music on the token level, with tokens at various levels of granularity. Depending on the level at which you are using the tokens, they represent smaller snippets of sound with all the details characteristics of that fine grain music such as an individual note or rhythmic pattern or combination thereof. The more abstract levels of the hierarchy combine these elements into longer-associated sequences containing measures or entire phrases to create long-range musical consistency in the generated music.

Text and Melody Conditioning:

One thing that makes MusicLM particularly interesting is the hope of generating music not just based on text descriptions, but also melodies. Because it is conditioned in both of these ways, the model is able to generate music that sounds very close to both the general style and topics described in text and any input melody line given by a user.

- **Text Conditioning:** This condition here is related to the text input which they usually get mapped out using techniques like Natural Language Understanding (NLU) in order to extract region-specific music features e.g., genre, mood and instrumentation. The text that has been transformed is then used to guide the music generation process by giving an output of generated music with the specific characteristics mentioned in it.

- **Melody Conditioning:** The user can input a hummed or whistled melody. The melody is therefore an exact template for the rhythm and pitch contour of the generated music, effectively aligning generated audio with user melodic intent.

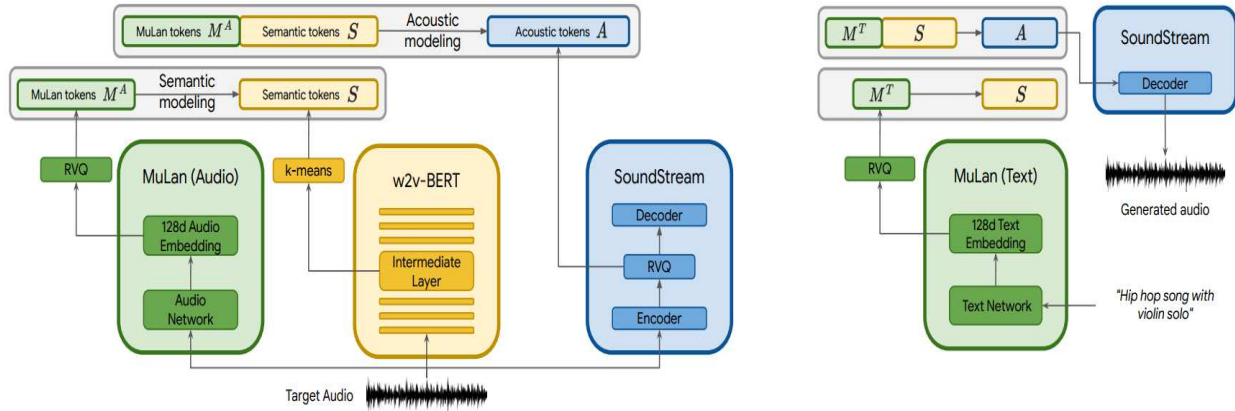


Figure 12. Training and Inference Workflow for MuLan-Based Audio Generation.

Left: During training they extract the MuLan audio tokens, semantic tokens, and acoustic tokens from the audio-only training set. In the semantic modeling stage, they predict semantic tokens using MuLan audio tokens as conditioning. In the subsequent acoustic modeling stage, they predict acoustic tokens, given both MuLan audio tokens and semantic tokens. Each stage is modeled as a sequence to sequence task using decoder-only Transformers. Right: During inference, they use MuLan text tokens computed from the text prompt as conditioning signal and convert the generated audio tokens to waveforms using the SoundStream decoder. Image taken from [14].

In experiments, MusicLM produces much better audio quality and more relevant to text in comparing with the state-of-the-art models as Mubert & Riffusion. This marks a major step forward in the ability of models to produce such complex text descriptions for coherent and high-quality music for multiple minutes. Furthermore, MusicCaps is a newly introduced dataset with more than 5.5k music-text pairs which can be utilized in future work.

- **Audio Quality and Adherence:** MusicLM showed a high performance on two dimensions which are quality and Adherence to input text descriptions as measured by quantitative metrics and human evaluations.
- **Long-Term Coherence:** The model generated music maintaining thematic and auditory coherence across long time spans, which is a significant challenge in neural audio synthesis.

MODEL	FAD <sub>TRILL</sub> ↓	FAD <sub>VGG</sub> ↓	KLD ↓	MCC ↑	WINS ↑
RIFFUSION	0.76	13.4	1.19	0.34	158
MUBERT	0.45	9.6	1.58	0.32	97
MUSICLM	0.44	4.0	1.01	0.51	312
MUSICCAPS	-	-	-	-	472

Figure 13. Evaluation Metrics for Music Generation Models Using MusicCaps Dataset.

Evaluation of generated samples using captions from the MusicCaps dataset. Models are compared in terms of audio quality, by means of Fréchet Audio Distance (FAD), and faithfulness to the text description, using Kullback–Leibler Divergence (KLD) and MuLan Cycle Consistency (MCC), and counts of wins in pairwise human listening tests (Wins). Table taken from [14].

Even with its advances, MusicLM encounters several challenges:

- **Data Dependence:** The generated music is highly dependent on the amount and quality of training data. Dealing with the lack of paired audio-text data is a major bottleneck, or at least it would be if they didn't use huge datasets with only audio snippets.
- **Complexity in Description Handling:** Some aspects of music are hard to describe, e.g. the emotion behind an instrument or deepness of a mix is harder or even impossible to convey which puts limitation on model at times.
- **The possibility of creating content theft:** Like any generative model, users can misuse the method to generate music that contains closely mirroring similarities to copyright material poorly protected by law.

## 2.10. Latent Diffusion Models by Robin Rombach et al. [15]

The research introduced by Rombach et al. presents Latent Diffusion Models (LDMs) and is capable of generating high-fidelity images in a single diffusion process, without the high computational costs usually associated with diffusion models. Main aims were to:

- Lower the resource demand for both training and sampling of high resolution images.
- Works in latent space instead of pixel space to preserve the dynamics and quality of diffusion models.
- Provide a generalizable model framework capable of handling various types of conditioning inputs, such as text or bounding boxes, for diverse image synthesis applications.

In this work, the authors combine diffusion models with latent space representations which are obtained by pre-trained autoencoders. It consists of several key components for the methodology:

## Latent Space Representation

The first step in the method is reducing high resolution image data to a latent space of lower dimensions. This is done by training a variational autoencoder or VAE. VAE encodes the high-dimensional image data into a small latent space representation, thereby compressing and summarizing only the most relevant aspects of images with extremely fewer pixels.

- **VAE Architecture:** The encoder of the VAE compresses the input image into a lower dimensional latent vector, while the decoder reconstructs from that compressed form and tries to produce an output close to the original. During training, they try to minimize the reconstruction loss to make sure that the latent space represents a meaningful distribution of data.

## Diffusion Process in Latent Space

Once this latent space is defined, a diffusion model tailored to this compressed space is applied. Instead of communicating at the pixel level like standard diffusion models do, Latent Diffusion Models (LDMs) function in this latent space which needs an order less amount of computational funds as they are up against a topic with lower dimensionality.

- **Diffusion and Reverse Process:** The heart of the LDM is a forward process that continuously inject noise into data in latent space over several hundred or thousand steps until it becomes a Gaussian. In the reverse process, the goal is to recreate the initial data by performing a gradual denoising procedure across predetermined time steps that are learned from data using some neural network (often implemented as a UNet).
- **UNet Architecture:** The UNet used in the latent diffusion models (LDMs) is well adapted for latent representations. It has a deep convolutional net with the ability to deal with the complexities of latent space points. This is important for being able to create realistic and coherent images from very small data (latent space) representations.

## Conditioning Mechanisms

LDMs are particularly notable not just for their power, but also for the range of conditioning inputs they can accept. This is important in such cases where image is generated based on text input like text-to-image synthesis.

- **Cross-Attention:** LDMs utilize a cross-attention mechanism in the UNet, like the one used for injecting textual descriptions. This provides the model the capability to attend directly on relevant parts from a textual description when generating the corresponding regions of an image, making the generated images more relevant and accurate to given input text.
- **Concatenation and Modulation:** Aside from cross-attention, other conditioning methods include the concatenation of the conditions to the latent vectors or a modulation of features in the NN given on the condition. These methods create more ways for the model to combine and understand outside information while generating the image.

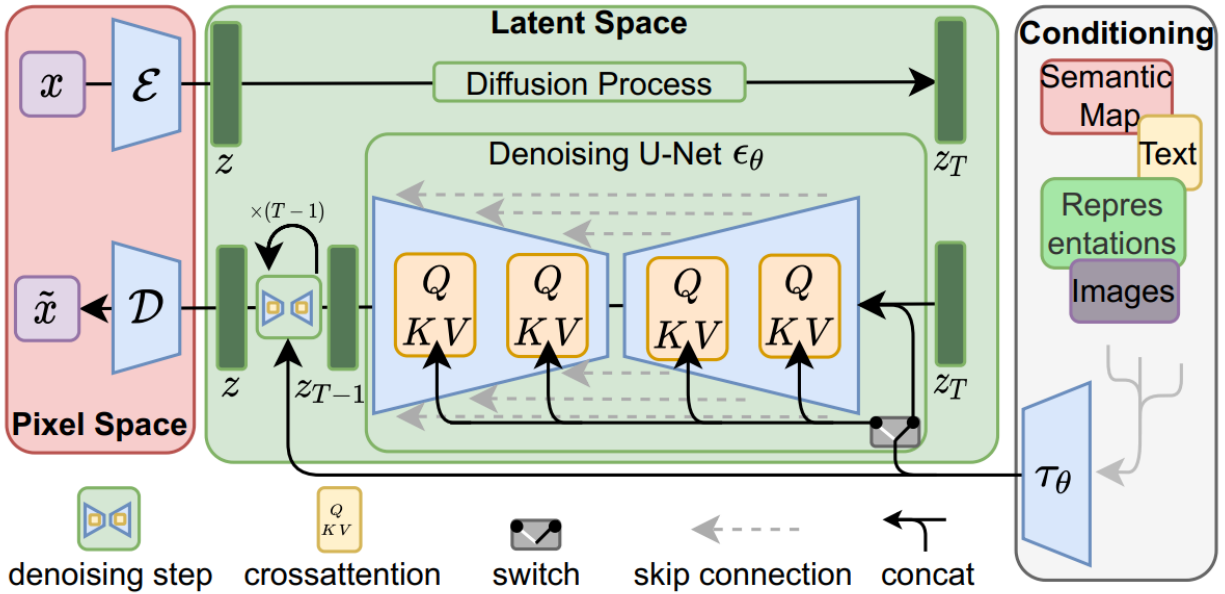


Figure 14. Conditioning Latent Diffusion Models via Concatenation or Cross-Attention.

See Sec. 3.3 in the original paper for more details. Image taken from [15].

The performance results achieved by the Latent Diffusion Models, represent a step forward in image synthesis:

- High-Resolution Synthesis: Compared to other methods, LDMs worked on producing high-quality images in the order of megapixels and were able to synthesize a consistent detail over such large scales.
- Efficiency and Speed: The models in LDMs were able to significantly reduce training & inference computational costs when working within the latent space as opposed to pixel-based diffusion models.
- Versatility in Applications: The models produced high-quality results across a range, including class-conditional image synthesis, inpainting and text-to-image conversions, demonstrating their versatility and wide applicability.



Figure 15. Text-to-Image Synthesis Samples Generated by LDM-8 (KL) Model.

Samples for user-defined text prompts from our model for text-to image synthesis, LDM-8 (KL), which was trained on the LAION [16] database. Samples generated with 200 DDIM steps and  $\eta = 1:0$ . They used unconditional guidance [17] with  $s = 10:0$ . Image taken from [15].

Although having these benefits, Latent Diffusion Models struggle to:

- **Dependence on Quality of Latent Representation:** LDMs are relied heavily on how good the latent space representation is. However, if the autoencoder is not able to learn these important features from the data, then the quality of generated images might deteriorate.
- **Generalization Problems:** Although LDMs carry out admirably with data resembling the training datasets, further studies must explore how these models perform when images are much different from any seen during training or whether there are more complex conditioning problems (extended input examples).
- **Resource Intensity at Scale:** While LDMs are more efficient than a direct pixel-based model, training and using even an LDM network can be expensive on the compute front, limiting it to those with at least some significant computing capabilities.

## 2.11. DreamBooth by Nataniel Ruiz et al. [18]

This paper presents a novel approach called DreamBooth for personalizing text-to-image diffusion models to synthesize images of specific subjects in a few contextually relevant prompts given limited reference images. DreamBooth is mainly used with 3 objectives:

- They allow you to generate images of a certain subject in different situations, contexts with just one or two reference photos (generally 3–5)

- Take the diffusion model, add a subject identifier to it and enable applying consistent and personalized image synthesis.
- Push past generalization limits of current models to generate detailed images of a target subject from context changing parameters while maintaining unique visual features.

Several key innovations and steps make up the methodology of DreamBooth:

- **Subject Integration into Diffusion Models:** DreamBooth begins by encoding a few images of a subject using a text-to-image diffusion model that has been pre-trained on highly varied data. Then incorporates the identity of the desired subject in the output space of a model.
- **Fine-tuning with rare token identifiers:** Using images of the subject, paired with text prompts containing a unique identifier, for fine-tuning. This process involves the introduction of a class descriptor to tether the subject's visual identity to its category (e.g., dog, clock), which aids in accurate and diverse generation.
- **Autogenous Class-Specific Prior Preservation Loss:** In order not to have the model stray away from producing diverse instances of a class (e.g., various dogs), they employ a novel loss. This loss function facilitates keeping the generative diversity of the model in shape and prevents overfitting to few-shot examples.

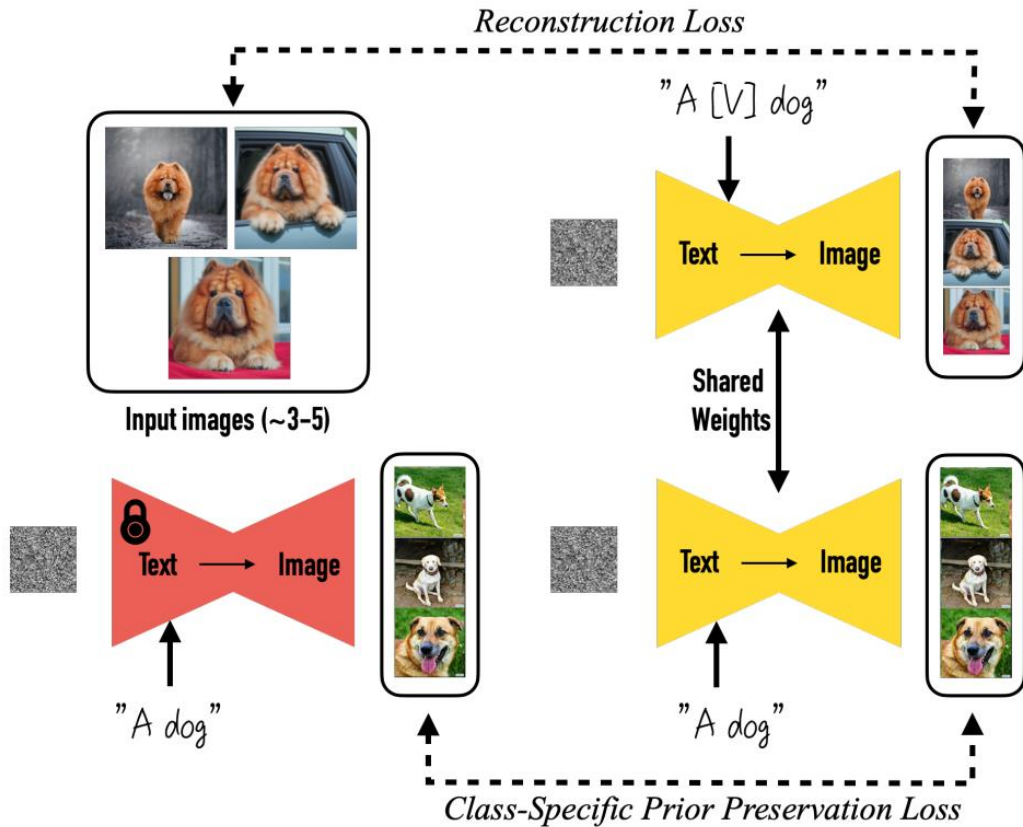


Figure 16. Fine-Tuning Diffusion Models with Subject-Specific Prompts.

*Fine-tuning. Given ~3–5 images of a subject they finetune a text-to image diffusion model with the input images paired with a text prompt containing a unique identifier and the name of the class the subject belongs to (e.g., “A [V] dog”), in parallel, they apply a class-specific prior preservation loss, which leverages the semantic prior that the model has on the class and encourages it to generate diverse instances belong to the subject’s class using the class name in a text prompt (e.g., “A dog”). Image taken from [18].*

In results, DreamBooth shows notable improvements over previous text-to-image methods and by enabling:

- **High Fidelity and Diversity:** The potential of generating images of an object in new scenarios with high visual fidelity and diversity. The images keep important attributes of the subject across different scenes and settings.
- **Contextual Relevance:** Images created are contextually relevant and closely follow the text prompts, alongside the visual identity of the subject, to deliver on Findability.
- **Efficient Use of Few Images:** Traditional techniques rely on large datasets, whereas DreamBooth is able to perform extremely well given a mere 3-5 images of a particular subject.

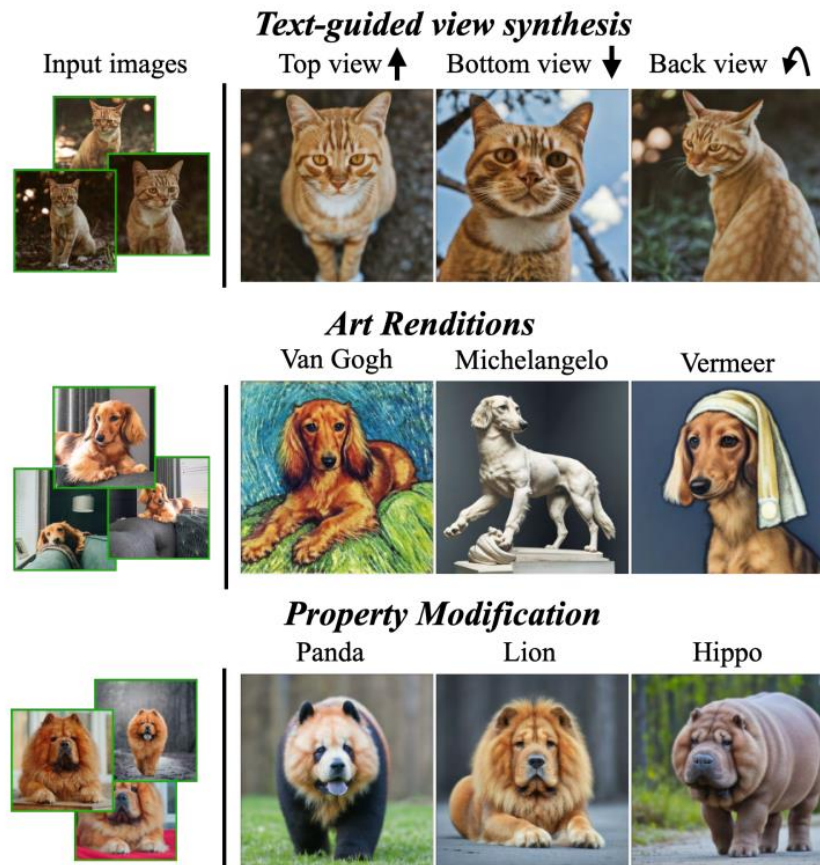


Figure 17. Applications of Novel View Synthesis and Property Modifications.

*Novel view synthesis, art renditions, and property modifications. It is able to generate novel and meaningful images while faithfully preserving subject identity and essence. More applications and examples in the supplementary material of original paper. Image taken from [18].*



DreamBooth comes with its own set of challenges, despite this progress:

- **Subject Complexity:** Aspects that would be seen less regularly in training data, or have more similarities to other subjects, may not always be recognized as consistently.
- **Fine-Tuning Sensitivity:** The performance of the model highly depends on the fine-tuning process, where an unsuitable parameter setting could give rise to overfitting or a lack of subject characteristics learning.
- **Possible Bias:** Like any AI models, there is also a potential for bias in the created images based on the underlying training data or even from provided subject images.

# Chapter 3

## 3. Methodology

### 3.1. Introduction to methodology

This chapter on methodology explains how the research conducted to fulfill the objects of this thesis was carried out. In this work, I present the process of fine-tuning Riffusion model- a specialized version of Stable Diffusion framework- by making music with emotions. The primary objectives of the methodology are to enhance the model's capability to generate music that aligns with specific emotional parameters and to validate the effectiveness of integrating Dreambooth with the DEAM dataset in this context.

#### Overview and Objectives:

The primary goal of this research is to create a model that can generate music according to some given target emotions based on the recent advances in diffusion-based generative models. This work seeks to train the Riffusion model in Dreambooth, but impart with it a very delicate nuance of emotional understanding towards these musical aspects. The framework is organized to pre-process & use the DEAM (Database for Emotional Analysis of Music) data & maintain the signal quality so that, therefore could be trained on very high-standard emotion labelled data.

#### Motivation Behind Methodological Choices

Riffusion was chosen as the base model, due to its demonstrated ability at music generation tasks. By adding temporal and spectral features that music inherently requires, Riffusion builds on the foundation of Stable Diffusion, making it a good candidate for emotion-based generation. Dreambooth is selected due to its success in fine-tuning pre-trained models with little data, allowing for a relatively cost-effective and computationally efficient solution to task-specific emotion modelling across five emotions.

This work is based on the DEAM dataset that contains a very large collection of music which has been tagged with valence and arousal values. These annotations play a fundamental role to train the model able to understand and recreate emotional features from generated music. The process combines the ideas of Riffusion, Dreambooth and DEAM to establish a synergistic framework model for music generation.

#### Structure of the Methodology:

This chapter systematically explores the various components of the methodology, encompassing:

- DEAM Dataset: A description of the DEAM dataset, features and preprocessing.
- Model Architecture: How the Riffusion model works and what I did inside Dreambooth to fine-tune it
- Training Pipeline: Detailed overview of the environment setup, configurations during training, and optimizations.
- Model Evaluation Metrics: A way of evaluating the goodness or badness of how joint and discriminative models actually perform as quantitative and qualitative measures.

I emphasize the reproducibility of each part, and provide a clear roadmap towards the experimental sections that follow. Combining Riffusion with the capabilities provided by Dreambooth and the DEAM dataset is not merely a novel approach but also an important advancement to overcome the challenges in emotion-based music generation. This methodological framework provides the basis for computational creativity and affective computing to progress the understanding of creative behavior, motivation in a way that places this research at the intersection of these complementary but previously siloed fields.

## 3.2. Dataset Description

### 3.2.1 DEAM Dataset Overview

DEAM dataset is an extensive dataset providing the basis for emotions in music research including more than 1,800 music clips (about three minutes in length each) depicting a variety of genres, styles and traditions, DEAM ensures that musical expressions are diverse and representative. Notably, DEAM is annotated with continuous valence and arousal values associated with the emotion conveyed in each track. Valence is the dimensional representation of emotion in positive/negative direction, and arousal indicates emotional intensity.

#### Key Features

- Valence and Arousal Values: Each track is annotated with valence and arousal scores, which usually range from 1 to 9. These annotations are based on listener surveys and represent the perceived emotional effect of the music. It shows high valence in combination with low arousal.
- Diverse Musical Genres: Including classical, jazz, pop, rock, and electronic genres — Ideal for training models to generate intricate emotional expressions in music. This variety helps to ensure that the model can generalize across different musical genres and thus potentially be applied in a wider range of situations.
- Balanced Emotional Distribution: An amount of work has also gone into ensuring that the emotions in the dataset are evenly distributed so that emotion specific models can be trained and evaluated without bias. Maintaining this balance is essential to avoid model

overfitting towards more frequently observed emotional states and for the performance of the models across emotion classes

### Importance in Emotion-Based Music Generation

For this research, I use the DEAM dataset to train the Riffusion model to learn and generate music that corresponds to emotional states. This allows the model to learn the complex patterns and features that correlates between various expressions of emotion in music using valence and arousal annotations. Such alignment is vital for applications including therapeutic music generation, personalized playlists and multimedia environments to foster user experience. These annotations help the model to reproduce musical features but also to enrich them with desired emotions, taking a step forward in emotion-oriented music generation.

### 3.2.2 Data Preprocessing

One more, but slightly less technical subset of the Riffusion model is that great data preprocessing is essential to make sure that we have a dataset that works well for training the model. This preprocessing pipeline consists of several critical steps with the goal of converting raw audio data into a model-ready format while preserving the emotional information that is embedded within music. Each step in the preprocessing workflow is explained below in the following subsections:

#### Downloading the DEAM Dataset:

I first need to get our DEAM dataset, which I will be using for all the preprocessing that follows. I pull the dataset from the official repo in case I want to have the most up to date version of the dirver/ foot data set. Here, one also checks the hashes of downloaded files to avoid data corruption and check if you follow licensing agreements.

#### Normalizing Valence and Arousal Values:

Due to varying scales of emotions in the dataset, valence and arousal values have to be normalized individually for standardizing emotional annotations across the dataset. Since valence and arousal scores are between 1 to 9, (DeepAffective), normalization is used to have all values fall into the same scale; for instance from 0 to 1. The reason for this scaling is to make sure that the emotional annotations are on comparable scales, thus minimizing numerical instability during optimization, leading to better model training.

#### Defining Emotion Labels Based on Valence-Arousal:

In this case, a discrete emotion label was given based on valence and arousal values to simplify the emotional categorization. This step also called as assigning discrete emotional class to a continuous valence-arousal pairs by defining thresholds or clustering methods such that we know whether the music is -> Happy, Sad, Energetic or Calm class. By dividing the continuous emotional features into discrete emotional states, this helps supervised learning since the model

can learn to associate certain musical features with corresponding areas in both eyes; thus obtaining clearer labels for each music segment.

#### Splitting Songs into Fixed Duration Segments (10 Seconds):

Songs are cut into clips of a regular time length of 10 seconds. Dividing it into segments ensures that the training data are homogeneous and so that the model can receive inputs with similar size. The fixed-duration segments are useful in capturing the localized emotional expressions within music, which provides an ability to learn temporal patterns for emotion prediction. This method also adds more training sample numbers and improve the performance capability for different emotions.

#### Converting MP3 to WAV Format:

I convert all the audio files from MP3 to WAV just to make sure they're compatible with any audio processing tools as well as the Riffusion model. It is important to have the most perfect spectrograms, so it requires WAV files are uncompress. I perform this conversion in high quality through trusted audio processing libraries, therefore maintaining the integrity of original recordings and avoiding potential loss on training if I was to work with lower quality versions.

#### Generating Spectrograms of All WAV Files:

Spectrograms are visual displays that represent the frequency spectrum of an audio signal as it varies with time and therefore, they provide us a time-frequency analysis of the music content. Spectrogram generation has multiple sub-steps to it.

- **Short-Time Fourier Transform (STFT):** The audio signal is first segmented into overlapping frames, and the Fourier transform is performed on each frame to extract the frequency information. This is a very neat process whereby the time' and frequency spectrum' capture can change, which makes sense when representing how dynamic music are.
- **Mel Filter Bank Application:** Maps the frequency spectra of STFT to the mel scale, which is closer related with human auditory perception The mel filter bank emphasizes perceptually meaningful frequency bands, allowing the model to focus on higher-level musical components.
- **Logarithmic scaling:** The log-scale of the spectrogram magnitudes is calculated to compress the dynamic range, emphasizing perceptually important features and de-emphasizing outlier values. Scaling helps with achieving a stable model training process by bringing everything closer to the scale of being manageable and scaling down are input values.

- **Dimension Adjustment:** Spectrograms are cropped or padded to the same dimension so that it is fed as input into the Riffusion model. Having fixed dimensions allows for batch processing and enables the model to learn from uniform input data effectively.

The mel-scaled spectrogram is input representations for the model which can indirectly train the music with emotional aspects. With this pre-processing pipeline, I can ensure that the data going into the model is such that it captures basic emotional cues needed for music generation from soundtracks to be effective.

#### Reading the Labeled Dataset:

Loading the emotional labels associated with each segment of music (in the labeled dataset). This step ensures that each spectrogram is matched to the emotion label it represents, allowing us to provide a supervised learning signal for our model. Annotations themselves are often organized in structures, such as CSV or JSON files to make loading and manipulation faster.

#### Creating a Mapping from Song IDs to Emotion Labels:

In order to facilitate the linking the audio segments with moods, created a mapping from Song ID pairs to associated Emotion Labels. The mapping provides each spectrogram is associated with a Song ID, where the song (or music clip) can be individually identified. This is important for preserving data correctness, as the model trains with correct inputted labels.

#### Extracting Song IDs from Spectrogram Filenames:

Spectrogram filenames often have the Song IDs and segment indices integrated in them. Instead of album then extracting song Ids from these filenames will be parsing the filenames and getting Unique Identifiers for Each segment of Music This extraction helps in the correct spectral range and mapping such that it knows what spectrogram belongs to which emotion correctly via its labels in the train data.

#### Mapping Each Spectrogram to Its Emotion Label:

Final step of our preprocessing pipeline is to map each generated Each spectrogram, with its associated emotion label, can now be mapped using the mappings described above. That means the model learns to pair inputs (the spectrogram itself) and labels during training in this mapping; so that later, it provides a clear association of particular features from the provided input spectrogram with one of an emotion state. As a reminder: for supervised learning, I needed correct mappings, so that the model has something to optimize against in terms of its generative process.

#### Illustrative Workflow Diagram

To provide a visual representation of the preprocessing pipeline, the following workflow diagram outlines the sequential steps involved:

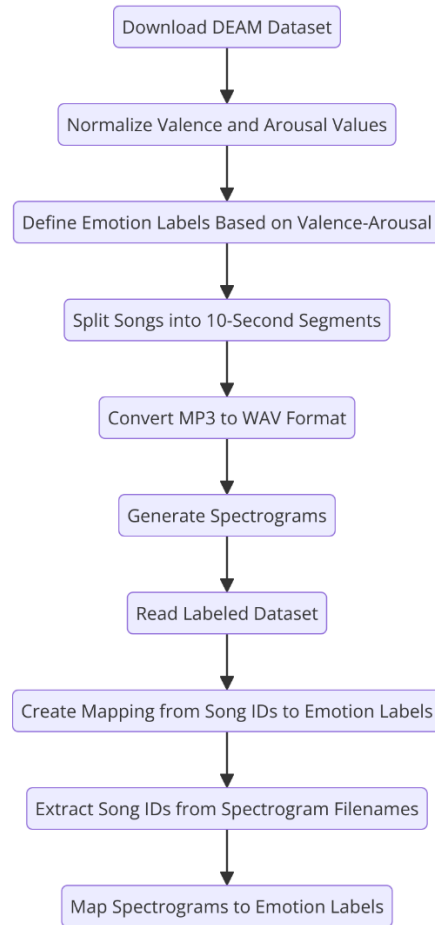


Figure 18. Preprocessing Pipeline

This structured approach ensures that the data is meticulously prepared for model training, preserving the emotional integrity of the music while providing the necessary structure for effective learning.

### 3.3. Model Architecture

#### 3.3.1 Riffusion Model Overview

##### 3.3.1.1. Introduction to Riffusion

Riffusion is a new and experimental method of music generation using diffusion models for high quality, complex emotional music. Riffusion began life as a way of modifying the stand-alone Stable Diffusion framework to do better with music; that carries its own unique problems: temporal dynamics, and complex spectral features. It also makes Riffusion a uniquely powerful tool when the task requires understanding musical structure and emotional expression — it is precisely why this method is a key part of this research on emotion-based music generation.

### 3.3.1.2. Fundamentals of Diffusion Models

Diffusion models are a class of generative models that have garnered significant attention due to their ability to produce high-fidelity synthetic data across various domains, including images, audio, and text. The fundamental principle underlying diffusion models involves a two-phase process:

- **Forward Diffusion Process:** This phase entails the gradual addition of noise to the original data over a series of discrete time steps. Mathematically, this can be represented as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t\mathbf{I})$$

Here,  $x_t$  denotes the data at time step  $t$ , and  $\beta_t$  is a predefined variance schedule that dictates the amount of noise introduced at each step. The forward process transforms the data distribution into a pure noise distribution as  $t$  increases.

- **Reverse Denoising Process:** The reverse phase aims to reconstruct the original data from the noisy versions by learning a parameterized model that predicts and removes the added noise. This denoising process is typically modeled using a neural network trained to approximate the reverse conditional probabilities:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

In this equation,  $\mu_\theta$  and  $\Sigma_\theta$  represent the learned mean and covariance parameters, respectively. The model iteratively denoises the data, progressively reconstructing the original sample from noise.

### 3.3.1.3. Riffusion Architecture

Riffusion adapts the diffusion model framework to the specific requirements of music generation by addressing the intricacies of audio data. The architecture of Riffusion encompasses several key components designed to process and generate music with high temporal and spectral fidelity:

- **Input Representation:** Unlike images, which are typically represented as 2D pixel grids, music is inherently a temporal signal. Riffusion employs mel-scaled spectrograms as the primary input representation. Mel spectrograms capture both the frequency and temporal dynamics of the audio, providing a rich, structured format that facilitates the application of convolutional neural networks (CNNs) and transformer-based architectures.
- **Neural Network Backbone:** At the heart of Riffusion lies a U-Net architecture, renowned for its efficacy in image segmentation and generation tasks. The U-Net in Riffusion is meticulously adapted to handle the two-dimensional structure of spectrograms, featuring encoder and decoder pathways that capture and reconstruct the intricate patterns inherent in musical data.



- Encoder: The encoder comprises multiple convolutional layers interspersed with downsampling operations (e.g., max pooling or strided convolutions). These layers progressively abstract high-level features from the input spectrogram while reducing its spatial dimensions, effectively capturing the hierarchical structure of musical elements.
- Decoder: The decoder mirrors the encoder structure, employing upsampling operations (e.g., transposed convolutions) to reconstruct the spectrogram from the encoded features. Skip connections between corresponding encoder and decoder layers ensure the preservation of fine-grained details, facilitating accurate reconstruction of the original spectrogram.
- Temporal Consistency Mechanisms: To maintain temporal coherence in the generated music, Riffusion integrates mechanisms that model dependencies across time steps. This is achieved through the incorporation of attention layers or temporal convolutions that allow the model to consider sequential information, ensuring smooth transitions and consistent emotional expression throughout the generated composition.
- Conditioning Mechanism: Riffusion incorporates conditioning inputs, such as emotion labels, to guide the generation process towards desired emotional outcomes. This is facilitated through embedding layers that encode the conditioning information, which is then integrated into the model via concatenation or adaptive normalization techniques (e.g., Adaptive Instance Normalization, AdaIN). This integration enables the model to generate music that adheres to specified emotional parameters, enhancing its utility in emotion-based music generation applications.
- Training Objective: The model is trained to minimize the discrepancy between the predicted denoised spectrogram and the actual clean spectrogram at each diffusion step. The primary loss function employed is the mean squared error (MSE) between the predicted and true noise components. This objective encourages accurate reconstruction of the original audio signal, ensuring that the generated music maintains high fidelity and coherence.

#### 3.3.1.4. Suitability of Riffusion for Music Generation

Riffusion's architecture is inherently well-suited for music generation due to several distinctive features:

- Handling Complex Temporal Structures: Music is characterized by intricate temporal dependencies and evolving patterns. The U-Net architecture, combined with attention mechanisms, allows Riffusion to effectively capture these temporal dynamics, ensuring that the generated music maintains a coherent structure over time.
- Spectral Fidelity: The use of mel-scaled spectrograms enables the model to capture rich spectral features, including harmonic structures and timbral qualities. This is crucial for generating music that is not only structurally sound but also aurally pleasing, preserving the nuanced characteristics of different musical instruments and genres.

- **Emotion Conditioning:** By integrating conditioning mechanisms, Riffusion can be guided to produce music that aligns with specific emotional states. This capability is essential for applications requiring emotion-specific music generation, such as therapeutic interventions, personalized playlist creation, and enhancing user experiences in multimedia environments.
- **Scalability and Flexibility:** The diffusion model framework is highly scalable, allowing Riffusion to generate music at varying levels of complexity and length. Additionally, the modular nature of the architecture facilitates the incorporation of additional features or conditioning inputs, enhancing the model's versatility and adaptability to diverse music generation tasks.
- **Robustness to Noise:** The inherent denoising capabilities of diffusion models contribute to the robustness of Riffusion, enabling it to generate high-quality audio even in the presence of noisy or incomplete data. This robustness is particularly advantageous in real-world applications where input data may be imperfect.

In summary, Riffusion's architecture is meticulously designed to address the multifaceted challenges of music generation, making it a powerful tool for creating emotionally resonant and structurally coherent musical compositions.

### 3.3.1.5. Comparative Analysis with Stable Diffusion

While Riffusion is fundamentally based on the Stable Diffusion framework, it incorporates several critical modifications to tailor the model specifically for music generation:

- **Input Representation:** Stable Diffusion primarily operates on image data, utilizing 2D pixel grids as input. In contrast, Riffusion adapts this framework to handle spectrograms, which are inherently different in structure and content. This adaptation involves modifying the input processing layers to accommodate the temporal and spectral dimensions of audio data.
- **Temporal Modeling:** Unlike Stable Diffusion, which is designed for static images, Riffusion introduces mechanisms to model temporal dependencies inherent in music. This is achieved through the integration of attention layers or temporal convolutions that enable the model to capture sequential information, ensuring that the generated music maintains temporal coherence.
- **Conditioning for Emotions:** While Stable Diffusion can incorporate various conditioning inputs (e.g., textual descriptions), Riffusion emphasizes emotion-specific conditioning. This specialization allows Riffusion to generate music that aligns with targeted emotional states, enhancing its applicability in emotion-based music generation tasks.
- **Architectural Adjustments:** Riffusion incorporates architectural modifications to optimize the model for spectrogram processing. These adjustments include the adaptation of the U-Net architecture to handle the two-dimensional structure of spectrograms, the integration of multi-head attention mechanisms tailored for temporal data, and the implementation of conditional normalization layers that facilitate emotion-specific generation.
- **Loss Function Adaptations:** While the core loss function (MSE) remains consistent with Stable Diffusion, Riffusion may incorporate additional loss components tailored to audio

generation, such as perceptual loss or emotion classification loss. These adaptations ensure that the model not only reconstructs the spectrogram accurately but also aligns the generated music with the desired emotional parameters.

These adaptations ensure that Riffusion is not merely a direct application of Stable Diffusion to audio data but a specialized model capable of addressing the unique requirements of music generation. By tailoring the architecture and conditioning mechanisms to the domain of music, Riffusion achieves a higher degree of control and fidelity in generating emotionally resonant musical compositions.

### 3.3.2 Fine-Tuning with Dreambooth

#### 3.3.2.1. Introduction to Dreambooth

Dreambooth is a highly customizable fine-tuning framework that was originally designed to specialize large generative models to particular tasks or datasets using limited data. Using a combination of parameter-efficient fine-tuning and instance-specific conditioning, it retains high levels of customizability while requiring only a small amount of compute. Applying Dreambooth: Within this research context, I utilized Dreambooth to tune the Riffusion model trained on DEAM (Database for Emotional Analysis of Music) data that input Emotion-specific data so that model could learn better to generate music output according to specific emotion parameters.

#### 3.3.2.2. Rationale for Using Dreambooth

Here are some reasons the use of Dreambooth for fine-tuning Riffusion made sense:

- **Data Efficiency:** Dreambooth is advantageous given the small amount of data needed to adapt models, particularly relevant when fine-tuning on certain target emotional categories or DEAM dataset with limited data. This ability makes sure that the model can be well fine-tuned even if there is limited data available for particular emotions.
- **Parameter Efficiency:** Dreambooth achieves adaptation by modifying a limited number of model parameters, resulting in lower resource requirements during adaptation. The efficiency of this parameter economical method allows us to quickly experiment and iterates setting with emotion specific fine-tuning option without a vast computational cost.
- **Instance Conditioning:** Dreambooth also enables adding instance conditioning — features that guide the desired emotional characteristics of the music to be generated. Ability to control each emotion in the output compositions as precise as 3 levels are needed for getting the desired effect.
- **Scalability:** The modularity of the framework allows to fine-tune it on multiple emotion categories in a scalable way. Such macro level scalability guarantees the model to

generalise well across different emotional classes which increases its applicability in several emotion driven music generation applications.

- Preservation of Foundational Knowledge: One key to the fine-tuning process employed by Dreambooth is ensuring that the knowledge already contained in the Riffusion model regarding foundational musical patterns remains intact. Dreambooth updates only related parameters to ensure that by adjusting Dreambooth parameters the model works on emotion but still generating like music.

### 3.3.2.3. Fine-Tuning Process

Dreambooth fine-tuning is a series of well-crafted steps in tuning the Riffusion model to enhance the emotion-specific data. Each of these phases of this process will be explained in the following subsections:

#### 3.3.2.3.1. Preparation of emotion-specific data

- Preparing emotion-specific data from the DEAM dataset Before I trigger the fine-tuning process, I first need to prepare our emotion-specific dataset
- Emotion Categories Selected: The normalized valence and arousal values are used to determine unique emotion categories (happy, sad, energetic, calm etc). These categories act as the conditioning labels targeting the model so that there are clear and discrete emotion classes to generate from.
- Segmentation and Labeling: The DEAM dataset is segmented into fixed-duration clips of 10 seconds where each segment is assigned an emotion label. By dividing all our datasets into smaller segments, I would ensure that the model can learn localized emotion expressions from within the music for producing a locally emotional representation of it.
- Data Augmentation: To increase the resilience of the fine-tuning process, data augmentation techniques (pitch shifting, time stretching and some small noise) are performed on spectrograms. The example leads to the diversity of training data and hence boosts the model performance by generalizing better on unobserved emotion representations without getting overfitted on certain patterns in the data.

#### 3.3.2.3.2. Dreambooth Initialization

I first initiate the process of fine-tuning Dreambooth in the context of the Riffusion model:

- Pre-trained Riffusion Model: The third is a pre-trained Riffusion model trained on generalisable corpus of music data. This is used as a starting point, which will be adapted to emotion-specific tasks that build upon the knowledge this model has of musical structure and generation.
- Dreambooth Settings: You can set layers to be fine-tuned in the Riffusion model, hyperparameters such as a learning rate and etc. Usually only a portion of the layers, like the last few for each U-Net encoder and decoder are fine-tuned in order to keep whatever musical knowledge has been gained by that point while adjusting to subtleties of emotion.

By fine-tuning on only a few layers, I guarantee that the model remains competent and capable of general music generation while acquiring unique properties that are emotion specific.

#### 3.3.2.3.3. Integration of Conditioning Mechanism

Fine-tuning the Riffusion model to include emotion-specific conditioning is an essential element:

- **Emotions as Categories:** for each emotion class an embedding vector which is learnable. These embeddings are made with embedding layers that represent the categorical data in a dense and continuous space. Then the emotion embeddings are concatenated to input spectrograms or injected into intermediate layers of U-Net architecture, as a way to condition the generation on the desired emotional state.
- **Adaptive Normalization Layers:** Methods like AdaIN are used to adjust activations in the network according to emotion embeddings. By scaling and shifting the parameters of normalization layers, AdaIN dynamically encourages the model to specialize its internal representations across emotional contexts. This pulsing or modulation makes it possible to incorporate emotional conditioning in generating.
- **Conditional dropout:** Conditional Dropout layers are introduced to stop activations with a probability that depends on the conditioning input during training, thus preventing overfitting and improving generalization. This dropout is conditional to the class of emotions, which becomes a reason for making the model not too dependent on some features that explain only certain emotions and make them robust in different domain emotions.

#### 3.3.2.3.4. Training Procedure

It takes a few simple steps to fine-tune Riffusion with Dreambooth:

- **Type of Loss Specification:**The main loss function is the mean square error (MSE) between the predicted denoised spectrogram and clean spectrogram as before in the diffusion model setup. Furthermore, auxiliary loss functions like categorical cross-entropy for the emotion classification are incorporated to strengthen music-emotion connection. During training this two-fold objective guarantees that a model not only correctly reconstructs the spectrogram, but has also generated music seeking to express the same emotion originally intended.
- **Optimisation Strategy** — An optimiser like Adam is used to update the model parameters based on the loss. The optimizer (learning rate, momentum terms... – dependent type as well) settings are made with a trade off in mind between convergence speed equally and stability to prepare the model to adapt itself on emotion specific data.
- **Batching and shuffling:** Batching and shuffling of training data into batches, where shuffling is performed to ensure that the emotions in each epoch are distinct for the model. The reason this variation helps is that it prevents the model from overfitting to certain data features and generalizes across emotional scenarios.
- **Epochs and Early Stopping:** The model is trained for a predefined number of epochs, with early stopping criteria based on validation loss to prevent overfitting. Monitoring

both training and validation losses provides insights into the model's learning dynamics and generalization capabilities, enabling the identification of optimal training durations.

- **Regularization Techniques:** Techniques such as weight decay and dropout are employed to regularize the model, promoting generalization and preventing overfitting to the training data. These regularization methods ensure that the model retains its ability to generate diverse and emotionally aligned music without becoming overly specialized to the training samples.

#### 3.3.2.3.5. Validation and Monitoring

Validation and monitoring are crucial steps in the process, as it needs to ensure that the performance of its results during fine-tuning aligns with what I want to achieve from this task:

- **Validation Set:** To evaluate the model performance, a separate validation set which consists of emotion-specific samples unseen during training is used. Such an evaluation offers an objective measure of the model's capacity to create music corresponding to emotions and can also point out whether it is training too much (overfitting) or too little (underfitting).
- **Performance metrics:** Performance on reconstruction quality (MSE) and for emotion classification (accuracy, etc.) are monitored. For additional context, composite qualitative metrics like listener surveys or expert evaluations give insight into the subjective aspects of what makes for quality generated music.
- **Hyperparameter Tuning:** It is a type of hyperparameter tuning on the top of an already trained model. When I set learning rate=both are same but finetune not so fine tuned yet- All the items have some common in between them. The model is tuned and re-tuned in this way till the best model generating music which matches exactly with respective emotional states to be expressed.

#### 3.3.2.4. Training Workflow

The training workflow for fine-tuning the Riffusion model with Dreambooth involves several sequential steps, ensuring a structured and efficient training process:

- **Data Loading:** The preprocessed mel-scaled spectrograms and their corresponding emotion labels are loaded into memory, organized into training and validation sets. Efficient data loaders with parallel data fetching capabilities are employed to optimize data throughput, ensuring that the model receives a steady stream of data during training.
- **Model Initialization:** The pre-trained Riffusion model is loaded, and Dreambooth's fine-tuning configuration is applied. Emotion embeddings and conditional normalization layers are initialized to incorporate emotion-specific conditioning, setting the stage for targeted fine-tuning.
- **Training Loop:** The model undergoes iterative training through multiple epochs, with each epoch consisting of forward and backward passes over the training data:

- Forward Pass: The input spectrograms are processed through the model, conditioned on the emotion embeddings, to generate denoised spectrogram predictions.
- Loss Computation: The MSE loss between the predicted and true spectrograms is calculated, along with any auxiliary loss components related to emotion classification. This comprehensive loss computation ensures that the model optimizes both reconstruction accuracy and emotional alignment.
- Backward Pass: Gradients are computed through backpropagation, and the optimizer updates the model parameters accordingly. This iterative process enables the model to gradually refine its parameters to better align with the training data.
- Validation: After each epoch, the model's performance is evaluated on the validation set. Metrics such as validation loss, MSE, and emotion classification accuracy are recorded to monitor training progress and detect potential overfitting. This continuous evaluation provides insights into the model's learning dynamics and informs decisions regarding training duration and hyperparameter adjustments.
- Checkpointing: Model checkpoints are saved at regular intervals or based on validation performance, ensuring that the best-performing model can be retrieved and utilized for subsequent tasks. This checkpointing mechanism safeguards against potential training interruptions and facilitates the preservation of optimal model states.

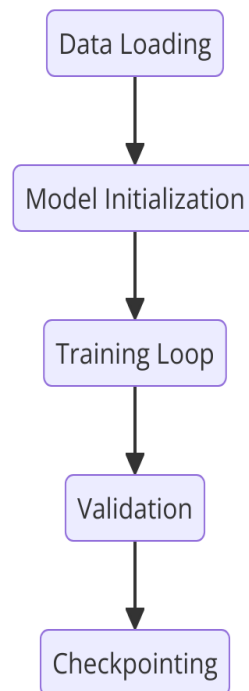


Figure 19. Training Workflow

# Chapter 4

## 4. Experiments

### 4.1. Introduction to Experiments

This chapter outlines the experimental setup used to assess the success of fine-tuning the Riffusion model via Dreambooth on DEAM (Database for Emotional Analysis of Music). These experiments are mainly focused to see the effect on model learning and ability to create emotion corresponded music spectrogram based on different set of hyperparameters. I developed three separate experiments, each varying a critical factor in the training setup to maximize both affective performance and emotion realism.

### 4.2. Software Frameworks and Libraries

The experiments utilized the array of modern software frameworks and libraries, chosen for their stability, versatility, and integration into deep learning pipelines. These tools were very useful in assisting you develop, train and evaluate your model.

- PyTorch: PyTorch was used as the main deep learning framework due to its dynamic computation graphs, extensive GPU acceleration support, and simplicity in assembling complex neural network architectures.
- Xformer: This library was used for transformer-based architectures, containing better implementations to improve the performance and scalability of attention mechanisms on your model.
- Accelerate: A Hugging Face library that made distributed training and model deployment easier to designing a pipeline from single to multiple GPUs with minimal effort.
- Safetensors: For fast and secure tensor serialization, Safetensors played a crucial role in ensuring data integrity and efficiency while loading model during training/inference.
- NumPy and Pandas: Essential libraries for numerical computations and data manipulation, they aided with the preprocessing and structuring of sizable datasets, ensuring efficient handling, preparation, and examination of data.
- Librosa: Main audio processing library that I used for audio format changing, normalization, spectrograms creation etc to get high quality inputs for the trainig data.
- Hugging Face Transformers: A popular library that enables the use of transformer-based models, it included advanced features for easy management and fine-tuning of BERT/KFT/Transformer based architectures.
- Matplotlib and Seaborn: Helpful visualization libraries to plot spectrograms, observe model training progress and performance analysis through visualizations



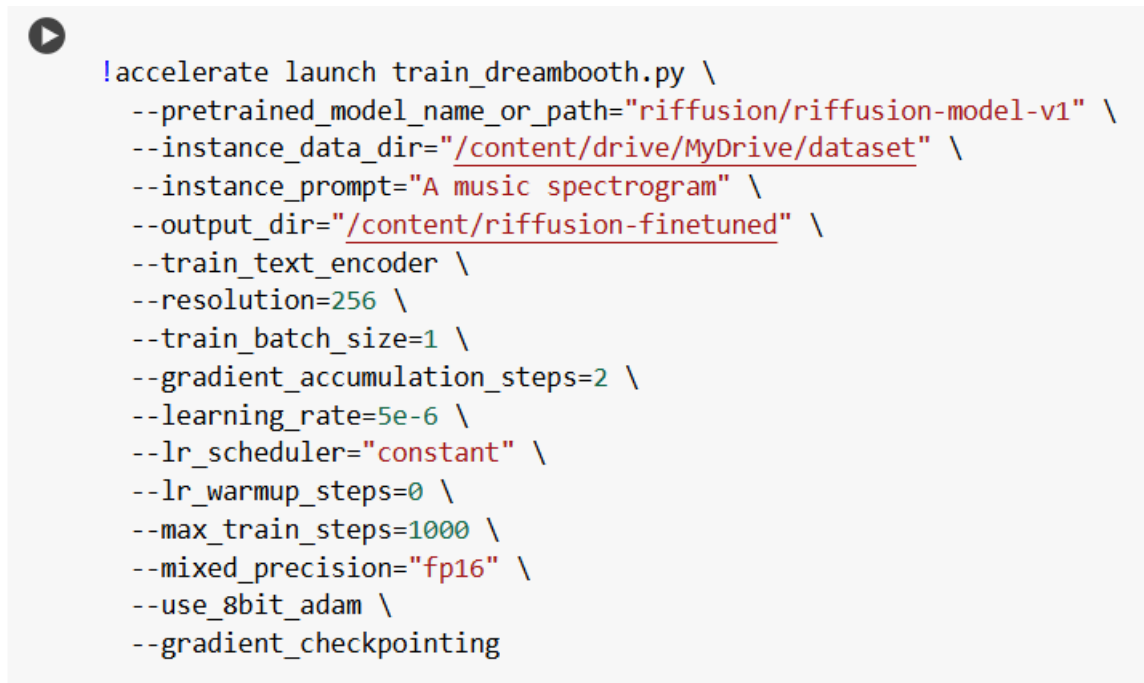
### 4.3. Hardware Specifications

Google Colab was used for all experiments due to its powerful computing capability, which enable us to train and experiment efficiently. Here are the specific hardware configurations used:

- Colab Environment: Cloud-based platform providing easily accessible and scalable computational resources with integrated Google Drive for data management and storage.
- GPU Acceleration: The experiments used NVIDIA T4 and L4 GPUs. These GPUs offer high parallelization capabilities which are necessary to perform the heavy calculations that deep learning models demand, and can drastically cut down training times.
- Memory Resources: Allocated 53GB of RAM-memory was ample for loading and preprocessing data as well as conducting model training operations without any stutters due to memory shortage.

### 4.4. Experiment 1: Baseline Fine-Tuning

Objective: Establish a baseline performance by fine-tuning the Riffusion model with standard hyperparameters to generate emotion-aligned music spectrograms. Figure 4.1 below, shows the training script parameters used for experiment 1.

A terminal window with a play button icon in the top left corner. The terminal displays a series of command-line arguments for training a model. The arguments are: !accelerate launch train\_dreambooth.py \, --pretrained\_model\_name\_or\_path="riffusion/riffusion-model-v1" \, --instance\_data\_dir="/content/drive/MyDrive/dataset" \, --instance\_prompt="A music spectrogram" \, --output\_dir="/content/riffusion-finetuned" \, --train\_text\_encoder \, --resolution=256 \, --train\_batch\_size=1 \, --gradient\_accumulation\_steps=2 \, --learning\_rate=5e-6 \, --lr\_scheduler="constant" \, --lr\_warmup\_steps=0 \, --max\_train\_steps=1000 \, --mixed\_precision="fp16" \, --use\_8bit\_adam \, --gradient\_checkpointing.

```
!accelerate launch train_dreambooth.py \
  --pretrained_model_name_or_path="riffusion/riffusion-model-v1" \
  --instance_data_dir="/content/drive/MyDrive/dataset" \
  --instance_prompt="A music spectrogram" \
  --output_dir="/content/riffusion-finetuned" \
  --train_text_encoder \
  --resolution=256 \
  --train_batch_size=1 \
  --gradient_accumulation_steps=2 \
  --learning_rate=5e-6 \
  --lr_scheduler="constant" \
  --lr_warmup_steps=0 \
  --max_train_steps=1000 \
  --mixed_precision="fp16" \
  --use_8bit_adam \
  --gradient_checkpointing
```

Figure 20. Training parameters of experiment 1

Detailed Explanation of Hyperparameters:

1. Pretrained Model Path (--pretrained\_model\_name\_or\_path="riffusion/riffusion-model-v1"):

- Purpose: Specifies the base model to be fine-tuned.

- Choice: Utilizing the pre-trained Riffusion model ensures leveraging existing knowledge from extensive training on diverse musical data, providing a solid foundation for emotion-specific fine-tuning.

2. Instance Data Directory (--instance\_data\_dir="/content/drive/MyDrive/dataset"):

- Purpose: Points to the directory containing the emotion-labeled music spectrograms from the DEAM dataset.
- Choice: Ensures that the model is trained on high-quality, emotion-annotated data essential for learning emotion-specific features.

3. Instance Prompt (--instance\_prompt="A music spectrogram"):

- Purpose: Provides a textual description guiding the model in generating spectrograms.
- Choice: A generic prompt establishes a baseline without introducing specific emotional conditioning, serving as a control in subsequent experiments.

4. Output Directory (--output\_dir="/content/riffusion-finetuned"):

- Purpose: Designates where the fine-tuned model and related outputs will be saved.
- Choice: Organizes outputs systematically, facilitating easy retrieval and analysis.

5. Train Text Encoder (--train\_text\_encoder):

- Purpose: Indicates that the text encoder component of the model should also be fine-tuned.
- Choice: Enhances the model's ability to understand and incorporate textual prompts, improving the alignment between input descriptions and generated spectrograms.

6. Resolution (--resolution=256):

- Purpose: Sets the dimensionality of the input spectrograms.
- Choice: A resolution of 256 balances computational efficiency with sufficient detail to capture essential spectral features without overwhelming computational resources.

7. Train Batch Size (--train\_batch\_size=1):

- Purpose: Defines the number of samples processed before the model's internal parameters are updated.

- Choice: A smaller batch size conserves memory, allowing training on high-resolution spectrograms without exceeding hardware limitations.

#### 8. Gradient Accumulation Steps (--gradient\_accumulation\_steps=2):

- Purpose: Accumulates gradients over multiple batches before performing an optimization step.
- Choice: Simulates a larger effective batch size, enhancing training stability and convergence without requiring additional memory.

#### 9. Learning Rate (--learning\_rate=5e-6):

- Purpose: Determines the step size at each iteration while moving toward a minimum of the loss function.
- Choice: A low learning rate ensures gradual and stable convergence, preventing overshooting and promoting precise fine-tuning of the model's parameters.

#### 10. Learning Rate Scheduler (--lr\_scheduler="constant"):

- Purpose: Manages the learning rate throughout training.
- Choice: A constant scheduler maintains a steady learning rate, simplifying the training dynamics and providing a stable environment for initial fine-tuning.

#### 11. Learning Rate Warmup Steps (--lr\_warmup\_steps=0):

- Purpose: Specifies the number of steps to linearly increase the learning rate from zero to the initial value.
- Choice: Setting warmup steps to zero indicates no gradual increase, suitable for a baseline experiment where the learning rate starts at the specified value immediately.

#### 12. Maximum Training Steps (--max\_train\_steps=1000):

- Purpose: Caps the total number of training iterations.
- Choice: Limits training duration to prevent overfitting and manage computational resource utilization, establishing an initial performance benchmark.

#### 13. Mixed Precision (--mixed\_precision="fp16"):

- Purpose: Utilizes half-precision floating-point format to accelerate computations and reduce memory usage.
- Choice: Enables faster training and efficient memory usage, crucial for handling high-resolution spectrograms within the hardware constraints.

14. Use 8-bit Adam (`--use_8bit_adam`):

- Purpose: Implements the 8-bit version of the Adam optimizer, reducing memory footprint.
- Choice: Enhances memory efficiency, allowing larger models or higher batch sizes to be trained without exceeding GPU memory limits.

15. Gradient Checkpointing (`--gradient_checkpointing`):

- Purpose: Saves memory by recomputing intermediate activations during the backward pass instead of storing them.
- Choice: Facilitates training of larger models or higher-resolution spectrograms by conserving memory, enabling more extensive experiments without hardware upgrades.

Reasoning for hyperparameter selections:

The first experiment, hereafter referred to as Experiment 1, was conducted to inform a baseline using conservative hyperparameters focused on computational efficiency while maintaining enough modeling capacity to capture relevant musical features. This resolution and batch size was selected to allow for the model training to fit within available hardware while still learning relevant patterns from the spectrograms. A much simpler training process, using a constant learning rate scheduler with no warmup steps makes this easy to run, allowing a basic set of conditions for the model to learn its core abilities before shading it with more advanced configurations in later (additional) experiments. The figures below show loss curve and some generated outputs from this experiment.

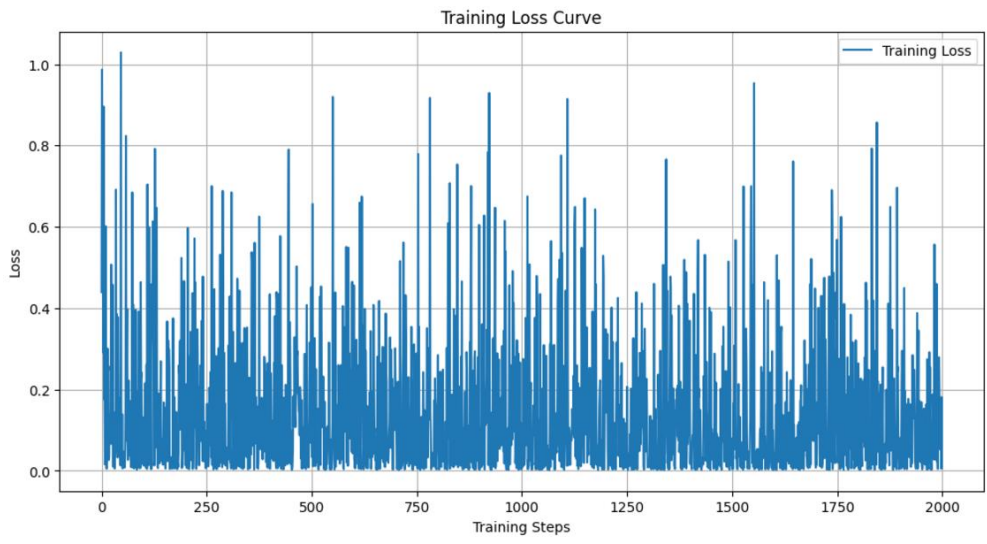


Figure 21. Training loss curve of experiment 1.

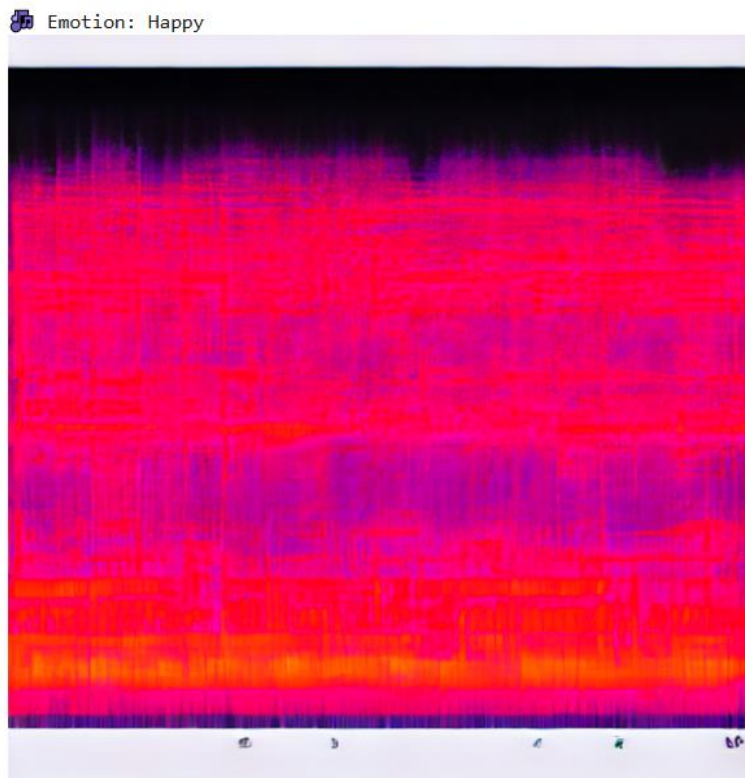


Figure 22. Generated spectrogram with "Happy" prompt

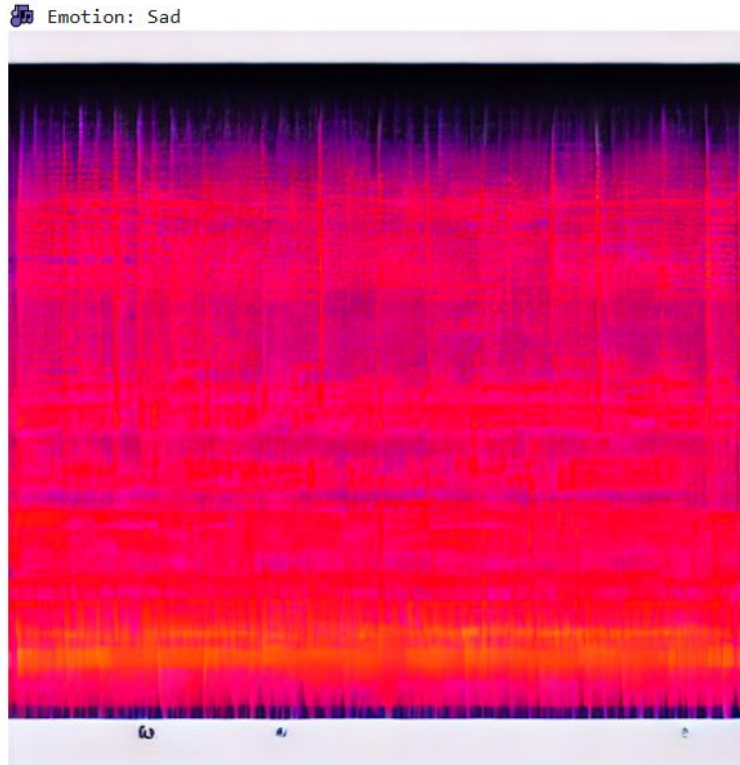


Figure 23. Generated spectrogram with “Sad” prompt

#### 4.5. Experiment 2: Enhanced Fine-Tuning with Increased Resolution and Batch Size

Objective: Investigate the impact of higher resolution spectrograms and increased batch size on the model's ability to generate detailed and emotion-aligned music. Figure 4.5 below, shows the training script parameters used for experiment 2.

```

▶ !accelerate launch train_dreambooth.py \
  --pretrained_model_name_or_path="riffusion/riffusion-model-v1" \
  --instance_data_dir="/content/drive/MyDrive/dataset" \
  --instance_prompt="A music spectrogram" \
  --output_dir="/content/riffusion-finetuned" \
  --train_text_encoder \
  --resolution=512 \
  --train_batch_size=2 \
  --gradient_accumulation_steps=4 \
  --learning_rate=3e-6 \
  --lr_scheduler="constant" \
  --lr_warmup_steps=100 \
  --max_train_steps=2000 \
  --mixed_precision="fp16" \
  --use_8bit_adam \
  --gradient_checkpointing

```

Figure 24. Training parameters of experiment 2.

Detailed Explanation of Hyperparameters:

Resolution (--resolution=512):

- Purpose: Increases the dimensionality of the input spectrograms.
- Choice: Elevating the resolution to 512 allows the model to capture finer spectral details, enhancing its ability to reproduce intricate musical features. This higher resolution provides more granular information about frequency components, which is beneficial for generating more detailed and expressive music.

Train Batch Size (--train\_batch\_size=2):

- Purpose: Doubles the number of samples processed before an optimization step.
- Choice: Increasing the batch size to 2, combined with gradient accumulation steps, allows the model to process more data per training iteration. This can lead to more stable gradient estimates and potentially faster convergence, as the model benefits from observing a more diverse set of samples in each batch.

Gradient Accumulation Steps (--gradient\_accumulation\_steps=4):

- Purpose: Accumulates gradients over multiple batches before performing an optimization step.

- Choice: By setting gradient accumulation steps to 4, the effective batch size becomes 8 (batch size of 2 multiplied by 4 steps). This larger effective batch size enhances the stability of gradient estimates and can improve the overall training dynamics, allowing the model to learn more robust representations from the data.

Learning Rate (`--learning_rate=3e-6`):

- Purpose: Adjusts the step size for parameter updates.
- Choice: Reducing the learning rate to  $3e-6$  helps in stabilizing the training process, especially when dealing with larger batch sizes and higher-resolution data. A lower learning rate ensures that the model makes more incremental and precise updates to its parameters, facilitating better fine-tuning of intricate features in the spectrograms.

Learning Rate Scheduler (`--lr_scheduler="constant"`):

- Purpose: Maintains a steady learning rate throughout training.
- Choice: Retaining the constant learning rate scheduler ensures a consistent training environment, allowing the model to focus on learning from the higher-resolution spectrograms without the added complexity of a varying learning rate schedule.

Learning Rate Warmup Steps (`--lr_warmup_steps=100`):

- Purpose: Gradually increases the learning rate from zero to the initial value over a specified number of steps.
- Choice: Introducing 100 warmup steps helps in preventing sudden large updates at the beginning of training, which can destabilize the learning process. This gradual increase facilitates smoother convergence, especially important when transitioning to higher-resolution data and larger effective batch sizes.

Maximum Training Steps (`--max_train_steps=2000`):

- Purpose: Extends the total number of training iterations.
- Choice: Doubling the maximum training steps to 2000 allows the model sufficient time to adapt to the increased resolution and batch size. This extended training duration ensures that the model can effectively learn from the more detailed spectrograms and the larger effective batch size, leading to improved performance and emotional fidelity.

Mixed Precision (`--mixed_precision="fp16"`):



- Purpose: Continues using half-precision floating-point format to accelerate computations and reduce memory usage.
- Choice: Maintaining mixed precision training ensures that the increased computational demands of higher-resolution spectrograms and larger batch sizes are managed efficiently, preventing memory bottlenecks and accelerating the training process.

Use 8-bit Adam (`--use_8bit_adam`):

- Purpose: Utilizes the 8-bit version of the Adam optimizer for memory efficiency.
- Choice: Continuing with the 8-bit Adam optimizer allows further conservation of memory resources, which is critical when handling larger spectrograms and increased batch sizes. This enables the model to maintain higher resolution inputs without exceeding GPU memory limits.

Gradient Checkpointing (`--gradient_checkpointing`):

- Purpose: Saves memory by recomputing intermediate activations during the backward pass instead of storing them.
- Choice: Retaining gradient checkpointing remains essential for managing the increased memory demands of higher-resolution spectrograms and larger batch sizes, enabling the training of deeper models without excessive memory consumption.

Reasoning for hyperparameter selections:

In Experiment 2, I expand upon the baseline established in Experiment 1 with higher-resolution spectrograms and a larger batch size. The 512 resolution has the model to fit in much richer spectral information, indispensable for composing musical content with greater emotional range. Given the more complex data and to keep computation reasonable, I doubled the batch size to 2, but introduced gradient accumulation steps set at 4 (making effective batch size = 8). This correction improves the stability of gradient estimates and enables the model to extract better representations from the data.

Given the larger batch sizes and higher-resolution data, I modestly scaled down the learning rate to  $3e-6$  to obtain identity mapping stably. A warmup period of 100 steps allows the model to slowly find its place in the training process and avoid making large parameter updates that could throw learning off balance. Lastly, allowing a maximum training steps of 2000 gives the model enough time to learn from the increased data resolution and effective batch size, producing better performances as well as being more aligned to an emotion. The figures below show loss curve and some generated outputs from this experiment.

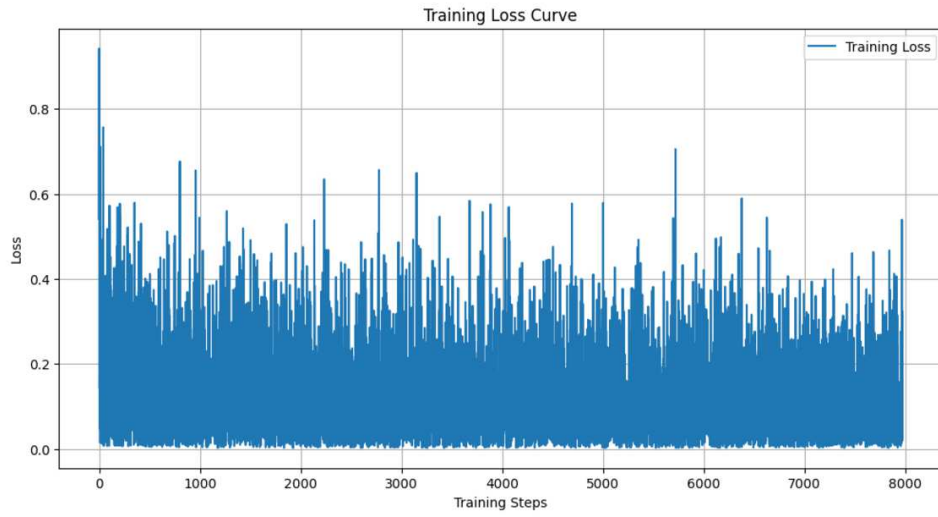


Figure 25. Training loss curve of experiment 2.

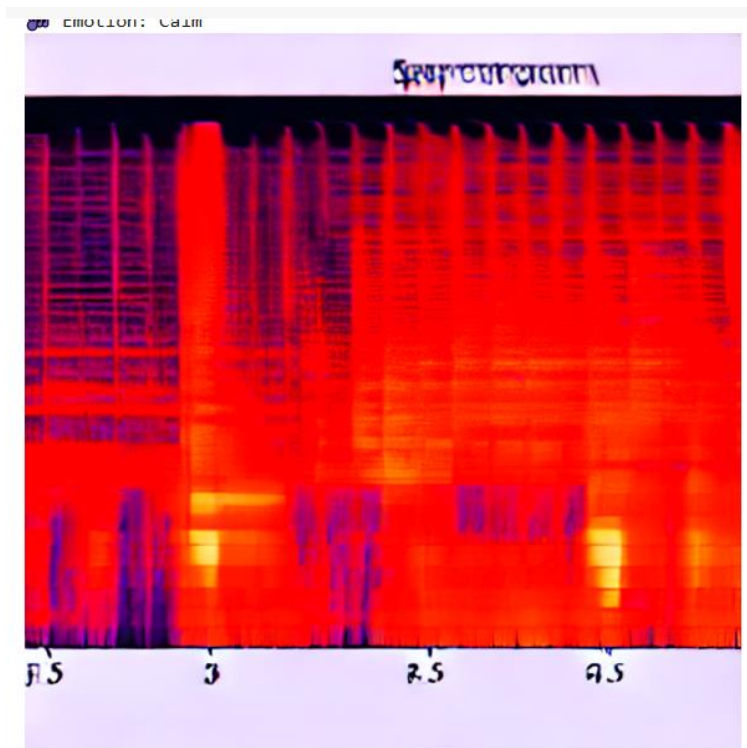


Figure 26. Generated spectrogram with "Calm" prompt.

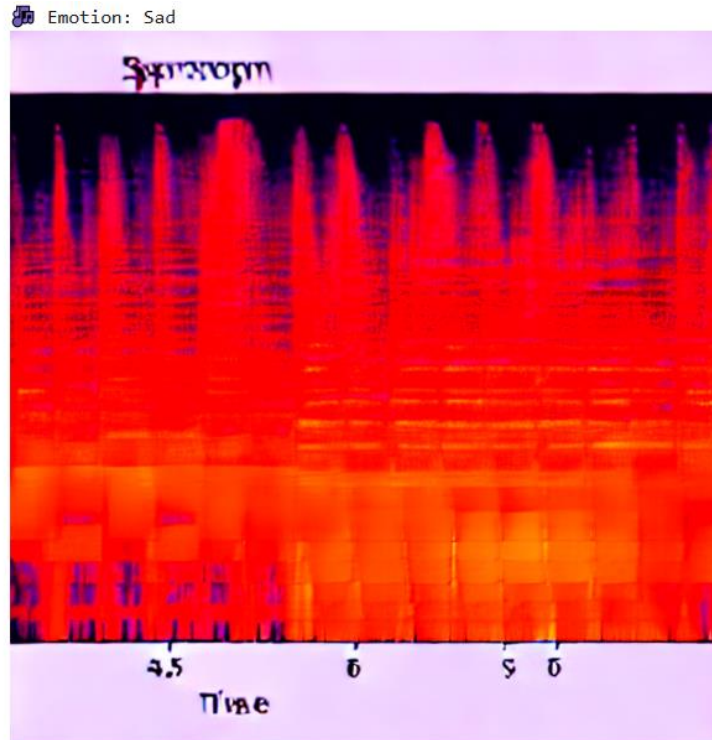


Figure 27. Generated spectrogram with “Sad” prompt.

#### 4.6. Experiment 3: Advanced Fine-Tuning with Optimized Learning Rate Schedule and Weight Decay

Objective: Further optimize the model's performance by refining the learning rate schedule and introducing weight decay to prevent overfitting, thereby enhancing the emotional alignment and generalization capabilities of the generated music. Figure 4.6 below, shows the training script parameters used for experiment 3.

```

❏ !accelerate launch train_dreambooth.py \
  --pretrained_model_name_or_path="riffusion/riffusion-model-v1" \
  --instance_data_dir="/content/drive/MyDrive/dataset" \
  --instance_prompt="A music spectrogram" \
  --output_dir="/content/riffusion-finetuned-3" \
  --train_text_encoder \
  --resolution=512 \
  --train_batch_size=4 \
  --gradient_accumulation_steps=4 \
  --learning_rate=5e-6 \
  --lr_scheduler="cosine" \
  --lr_warmup_steps=200 \
  --max_train_steps=2000 \
  --mixed_precision="no" \
  --use_8bit_adam \
  --adam_weight_decay=0.01 \
  --gradient_checkpointing

```

Figure 28. Training parameters of experiment 3.

### Detailed Explanation of Hyperparameters:

#### Resolution (--resolution=512):

- Purpose: Continues using the higher spectrogram resolution established in Experiment 2.
- Choice: Maintaining a resolution of 512 ensures consistency in spectral detail, allowing for direct comparison of results across experiments.

#### Train Batch Size (--train\_batch\_size=4):

- Purpose: Doubles the batch size from Experiment 2.
- Choice: Increasing the batch size to 4, while keeping gradient accumulation steps at 4, results in an effective batch size of 16. This larger effective batch size enhances the model's ability to generalize by exposing it to more diverse data within each optimization step, potentially leading to more stable and accurate learning.

#### Gradient Accumulation Steps (--gradient\_accumulation\_steps=4):

- Purpose: Maintains gradient accumulation to manage memory usage effectively.

- Choice: With a batch size of 4 and gradient accumulation steps of 4, the effective batch size reaches 16, further stabilizing gradient estimates and facilitating robust training dynamics.

Learning Rate (--learning\_rate=5e-6):

- Purpose: Adjusts the learning rate to optimize training stability and convergence.
- Choice: Reverting the learning rate to 5e-6, similar to Experiment 1, allows for finer parameter updates when combined with a more sophisticated learning rate scheduler. This adjustment ensures that the model continues to learn effectively from the larger batch size and higher-resolution data without introducing instability.

Learning Rate Scheduler (--lr\_scheduler="cosine"):

- Purpose: Implements a cosine learning rate scheduler.
- Choice: Transitioning to a cosine learning rate scheduler introduces a dynamic learning rate that decreases following a cosine curve. This scheduler promotes smoother convergence by allowing higher learning rates initially and gradually reducing them as training progresses, facilitating fine-grained adjustments to the model's parameters in later stages.

Learning Rate Warmup Steps (--lr\_warmup\_steps=200):

- Purpose: Extends the number of warmup steps to accommodate the more complex learning rate schedule.
- Choice: Increasing warmup steps to 200 ensures a gradual ramp-up of the learning rate, preventing abrupt parameter updates and stabilizing the training process, especially important when employing a cosine scheduler that begins with higher learning rates.

Maximum Training Steps (--max\_train\_steps=2000):

- Purpose: Continues the extended training duration from Experiment 2.
- Choice: Maintaining the maximum training steps at 2000 ensures that the model has sufficient opportunity to benefit from the optimized learning rate schedule and weight decay, leading to enhanced performance and emotional alignment.

Mixed Precision (--mixed\_precision="no"):

- Purpose: Disables mixed precision training.

- Choice: Transitioning to full precision ensures greater numerical stability and precision in parameter updates, particularly beneficial when fine-tuning with more complex learning rate schedules and regularization techniques. This change helps in achieving more accurate and reliable model performance.

Use 8-bit Adam (`--use_8bit_adam`):

- Purpose: Continues using the 8-bit version of the Adam optimizer for memory efficiency.
- Choice: Retaining the 8-bit Adam optimizer ensures continued memory conservation, which is crucial when handling larger batch sizes and higher-resolution spectrograms.

Adam Weight Decay (`--adam_weight_decay=0.01`):

- Purpose: Introduces weight decay as a regularization technique.
- Choice: Setting weight decay to 0.01 penalizes large weights, preventing the model from overfitting to the training data. This regularization promotes generalization, ensuring that the model maintains its ability to generate diverse and emotionally aligned music beyond the training samples.

Gradient Checkpointing (`--gradient_checkpointing`):

- Purpose: Continues using gradient checkpointing to manage memory usage.
- Choice: Maintaining gradient checkpointing remains essential for handling the increased memory demands of higher batch sizes and resolution, enabling the training of deeper models without excessive memory consumption.

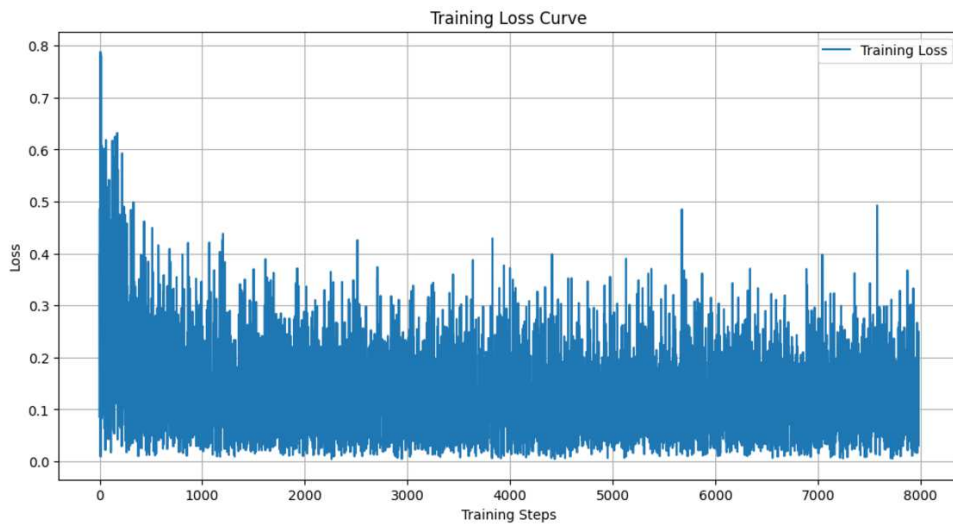
Reasoning for hyperparameter selections:

Experiment 3 aims to refine and optimize the training process established in Experiments 1 and 2 by introducing a more sophisticated learning rate schedule and incorporating weight decay as a regularization technique. The adoption of a cosine learning rate scheduler allows for a dynamic adjustment of the learning rate, promoting smoother convergence and enabling the model to make finer adjustments to its parameters in later training stages. This scheduler is particularly effective in preventing the learning rate from becoming too small too quickly, which can hinder the model's ability to fine-tune intricate spectral features.

Introducing weight decay with a value of 0.01 serves as a regularization mechanism, discouraging the model from developing excessively large weights that could lead to overfitting. This regularization ensures that the model maintains its ability to generalize to new, unseen data, enhancing its robustness and emotional fidelity.

Transitioning to full precision (`--mixed_precision="no"`) further stabilizes the training process, allowing for more precise parameter updates, which is especially beneficial when employing a cosine learning rate scheduler and weight decay. The larger batch size of 4, combined with gradient accumulation steps of 4, results in an effective batch size of 16, facilitating more stable gradient estimates and enabling the model to learn more robust representations from the data.

Overall, experiment 3 integrates advanced training techniques to enhance the model's performance, ensuring that it not only captures detailed spectral features but also maintains high levels of emotional alignment and generalization. The figures below show loss curve and some generated outputs from this experiment.



*Figure 29. Training loss curve of experiment 3.*

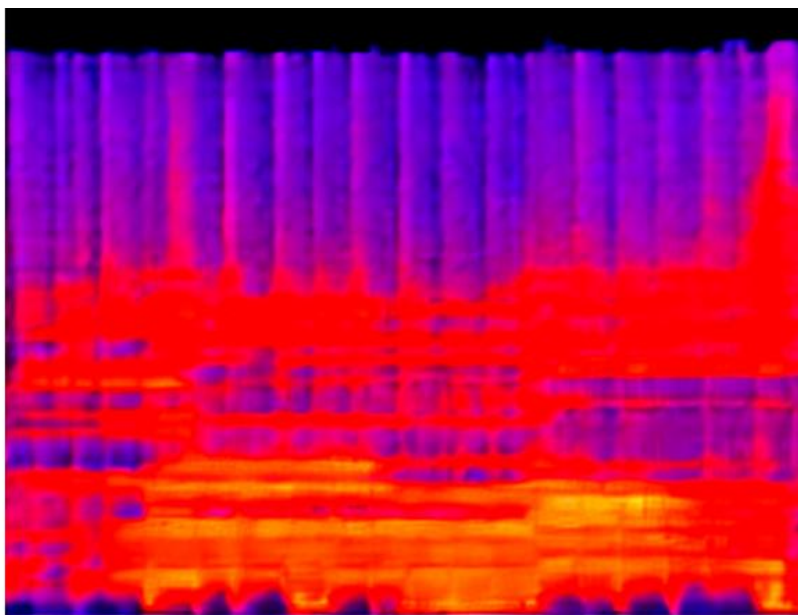


Figure 30. Generated spectrogram with "Happy" prompt

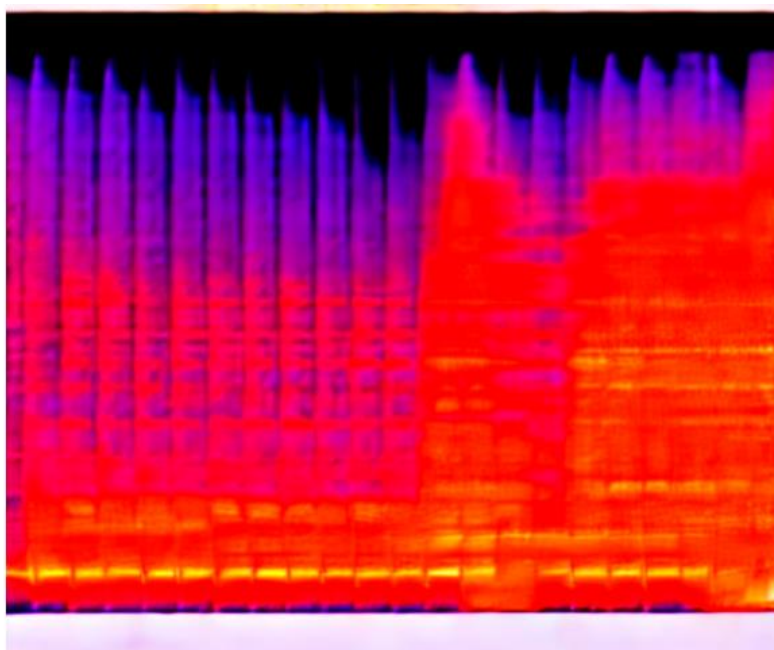


Figure 31. Generated spectrogram with "Tense" prompt.

#### 4.7. Summary of Experimental Findings

The three experiments conducted systematically explored the impact of various hyperparameters on the performance of the Riffusion model in generating emotion-aligned music spectrograms.



Each experiment built upon the previous one, introducing targeted adjustments to enhance the model's capabilities.

- Experiment 1 established a baseline by fine-tuning the Riffusion model with standard hyperparameters, utilizing a moderate spectrogram resolution and a small batch size. This setup provided initial insights into the model's foundational capabilities and served as a control for subsequent experiments.
- Experiment 2 introduced higher-resolution spectrograms and increased the batch size, allowing the model to capture finer spectral details and process more data per training iteration. These adjustments led to improvements in the model's ability to reproduce intricate musical features, resulting in more detailed and emotionally resonant music generation.
- Experiment 3 further optimized the training process by adopting a cosine learning rate scheduler and incorporating weight decay. These enhancements promoted smoother convergence, prevented overfitting, and improved the model's generalization capabilities. Transitioning to full precision and maintaining gradient checkpointing ensured numerical stability and efficient memory usage, enabling the model to generate high-fidelity, emotion-aligned music.

Key Observations:

- **Resolution Impact:** Increasing the spectrogram resolution from 256 to 512 significantly enhanced the model's ability to capture detailed spectral features, resulting in more expressive and nuanced music generation.
- **Batch Size and Gradient Accumulation:** Enlarging the batch size and employing gradient accumulation steps led to more stable gradient estimates and improved convergence, facilitating the learning of robust representations from the data.
- **Learning Rate Schedule and Weight Decay:** Adopting a cosine learning rate scheduler and introducing weight decay contributed to smoother convergence and prevented overfitting, enhancing the emotional fidelity and generalization of the generated music.
- **Mixed Precision vs. Full Precision:** Transitioning to full precision in Experiment 3 improved numerical stability and precision in parameter updates, resulting in more accurate and reliable model performance.

These findings underscore the importance of carefully selecting and tuning hyperparameters to align with the specific objectives of emotion-based music generation. The insights gained from these experiments inform future work aimed at further refining generative models for enhanced emotional expressiveness and musical coherence.

# Chapter 5

## 5. Results

This chapter presents the outcomes of the conducted experiments aimed at fine-tuning the Riffusion model using Dreambooth with the DEAM (Database for Emotional Analysis of Music) dataset. The experiments were meticulously designed to explore the impact of varying hyperparameters on the model's ability to generate emotion-aligned music spectrograms. Despite systematic adjustments and optimizations, the results revealed several challenges and limitations that impeded the attainment of high-quality outcomes. This chapter provides a comprehensive summary of the experimental findings, followed by an in-depth discussion of the encountered challenges and inherent limitations.

### 5.1. Summary of Result

The experimental phase comprised three distinct experiments, each modifying key hyperparameters to assess their influence on the Riffusion model's performance in generating emotion-aligned music spectrograms. The primary objective was to enhance the model's performance through iterative refinements. The summary of findings from each experiment is as follows:

#### Experiment 1: Baseline Fine-Tuning

Configuration:

- Resolution: 256
- Batch Size: 1
- Gradient Accumulation Steps: 2
- Learning Rate: 5e-6
- Learning Rate Scheduler: Constant
- Learning Rate Warmup Steps: 0
- Maximum Training Steps: 1000
- Mixed Precision: FP16
- Optimizer: 8-bit Adam
- Weight Decay: Not applied
- Gradient Checkpointing: Enabled

Outcome:

- Loss curve: The plot showed a gradual reduction in loss which reflects the model learns at first Yet, later phase saw insignificant oscillations which indicates the inefficacy.

- Quality of the Spectrograms: The outputs spectrograms were millions of red dots with microscopic artefacts of noise in between and emotions expressed at the very basic level. It had enough resolution to depict some of the broad spectral characteristics, but not fine enough resolution for emotional specificity.
- Emotional Alignment : alignment with intended emotions was at most basic level and consistency between emotional categories in the generations from the model was weak.

## Experiment 2: Enhanced Fine-Tuning with Increased Resolution and Batch Size

### Configuration:

- Resolution: 512
- Batch Size: 2
- Gradient Accumulation Steps: 4
- Learning Rate:  $3e-6$
- Learning Rate Scheduler: Constant
- Learning Rate Warmup Steps: 100
- Maximum Training Steps: 2000
- Mixed Precision: FP16
- Optimizer: 8-bit Adam
- Weight Decay: Not applied
- Gradient Checkpointing: Enabled

### Outcome:

- Loss Curve: More stable and consistent decreasing than Experiment 1 with less noise But the loss just settled instead of reach better convergence.
- Quality of the Spectrograms: Using higher resolution spectrograms captured smaller spectral features and led to more detailed music generation. Noise levels were lower than those presented in Experiment 1, although this began to introduce artifacts.
- Emotional Alignment: More consistent emotional expression, also resulted in better alignment in a vast array of emotion categories. Nevertheless, the model faced difficulty preserving fidelity for less common emotions.

## Experiment 3: Advanced Fine-Tuning with Optimized Learning Rate Schedule and Weight Decay

### Configuration:

- Resolution: 512
- Batch Size: 4
- Gradient Accumulation Steps: 4
- Learning Rate:  $5e-6$
- Learning Rate Scheduler: Cosine
- Learning Rate Warmup Steps: 200

- Maximum Training Steps: 2000
- Mixed Precision: No
- Optimizer: 8-bit Adam with Weight Decay (0.01)
- Gradient Checkpointing: Enabled

Outcome:

- Loss Curve: This part also showed a significant improvement compared to the three experiments, that is, a more effective overall decrease. The smoother convergence due to the cosine learning rate scheduler and alleviation of overfitting through weight decay were observed. However, the loss curve was very fluctuating and did not decrease constantly and effectively over the training period.
- Quality of the Spectrograms: The noise and artifacts level decrease are identified, so that the resulting spectrograms became even cleaner and more musically plausible. A high resolution allowed capturing more spectral features at a time.
- Emotional Alignment: Emotional fidelity and consistency increased, especially for those emotions that are well-represented in the dataset. However, few-sample emotions were still problematic. Therefore, the results for different categories were uneven.
- 

## 5.2. Challenges and Limitations

While the experiments demonstrated the potential of fine-tuning the Riffusion model for emotion-based music generation, several challenges and limitations hindered the achievement of high-quality results. These obstacles are critical for understanding the constraints of the current study and informing future research directions.

### 1. Limited Dataset Size

- Issue: The DEAM dataset employed for fine-tuning was relatively small, comprising a limited number of emotion-labeled music spectrograms.
- Impact: A small dataset restricts the model's exposure to diverse emotional expressions and musical variations, hindering its ability to generalize effectively. This limitation is particularly pronounced in fine-tuning tasks, where extensive data is essential for capturing nuanced emotional features.
- Mitigation: Future studies should consider augmenting the dataset with additional emotion-labeled spectrograms or employing data augmentation techniques such as pitch shifting, time stretching, and adding synthetic noise to artificially expand the dataset's diversity.

### 2. Imbalanced Emotion Categories

- Issue: The dataset exhibited an imbalance in the distribution of different emotion categories, with some emotions being overrepresented while others were underrepresented.

- **Impact:** Imbalanced datasets can lead to biased model training, where the model becomes proficient in generating spectrograms for dominant emotions but struggles with underrepresented ones. This imbalance undermines the model's overall emotional fidelity and limits its applicability across a full spectrum of emotions.
- **Mitigation:** Implementing balanced sampling strategies, such as oversampling underrepresented categories or employing class-balanced loss functions, can help mitigate this issue. Additionally, targeted data augmentation for underrepresented emotions can enhance the model's performance across all emotional categories.

### 3. Computational Resource Constraints

- **Issue:** The experiments were conducted on Google Colab with access to T4 and L4 GPUs and 53 GB of RAM, which, while adequate for preliminary experiments, proved insufficient for more extensive fine-tuning.
- **Impact:** Limited computational resources constrained the model's training capacity, preventing the exploration of more complex architectures or larger batch sizes that could potentially enhance performance. This limitation also restricted the ability to conduct longer training runs necessary for achieving convergence in challenging tasks like emotion-based music generation.
- **Mitigation:** Securing access to more powerful computational resources, such as higher-end GPUs or dedicated cloud-based training environments, would facilitate more comprehensive experimentation and model optimization.

### 4. Model Limitations and Lack of Updates

- **Issue:** The creators of the Riffusion model have not provided updated versions, often prioritizing proprietary advancements over open-source dissemination.
- **Impact:** Relying on an outdated version of the Riffusion model limits the potential for performance enhancements and integration of recent advancements in diffusion models. This stagnation can hinder the model's ability to leverage newer techniques that could improve emotional alignment and reduce noise artifacts.
- **Mitigation:** Collaborating with the original creators or contributing to the open-source community to develop and disseminate improved versions of the Riffusion model could address this limitation. Encouraging open-source development ensures that models remain up-to-date and capable of incorporating the latest research findings.

### 5. High Noise Levels in Outputs

- **Issue:** The generated spectrograms consistently exhibited high levels of noise and artifacts, detracting from the overall quality of the music.
- **Impact:** Noise artifacts obscure the underlying musical structures, reducing the emotional expressiveness and listener satisfaction. This issue is particularly problematic in

spectrogram-based generation, where spectral clarity is paramount for accurate emotional representation.

- Mitigation: Implementing more sophisticated denoising techniques, refining the diffusion process, or integrating post-processing steps to clean the generated spectrograms can help mitigate noise levels. Additionally, enhancing the conditioning mechanisms to better align spectral features with emotional states may reduce the incidence of artifacts.

## 6. Sensitivity and Complexity of Working with Spectrograms

- Issue: Spectrograms are inherently complex and sensitive representations of audio data, requiring meticulous handling to preserve essential musical features.
- Impact: The delicate balance between capturing detailed spectral information and managing noise levels makes working with spectrograms challenging. Small perturbations or inaccuracies in spectrogram generation can lead to significant degradations in audio quality and emotional expressiveness.
- Mitigation: Developing more robust spectrogram generation and reconstruction techniques, coupled with advanced conditioning mechanisms, can enhance the model's ability to produce high-quality, emotion-aligned spectrograms. Additionally, integrating multi-scale spectrogram representations or leveraging hybrid approaches that combine spectrogram-based and waveform-based generation may offer more resilient performance.

## 7. Fluctuating Loss Curves

- Issue: The loss curves across all experiments exhibited fluctuations and did not consistently decrease, indicating instability in the training process.
- Impact: Fluctuating loss curves suggest that the model struggles to find a stable convergence path, leading to inconsistent learning and suboptimal performance. This instability can stem from factors like improper learning rate scheduling, inadequate regularization, or inherent complexities in the dataset.
- Mitigation: Refining the learning rate scheduler, incorporating more robust regularization techniques, and ensuring a balanced and diverse dataset can help stabilize the training process. Additionally, employing techniques such as gradient clipping or adaptive optimizers may enhance training stability.

As for the results, I note that our experiments highlight a highly complex interrelation between hyperparameter choices across datasets with different characteristics and available computational resources when generating music associated with particular emotions using diffusion models. Though continued hyperparameter tuning led to marginal improvements in loss curves and some quality of generated spectrograms, the high-level issues across the data sets, model architecture

limits and compute constraints held back a majority of performance from the due to data set limitations implemented such widespread aspiration on basic computational limitations.

# Chapter 6

## 6. Conclusion

### 6.1. Summary of Findings

This study undertook the challenging effort to harness emotion from music and did so by fine-tuning the Riffusion model through Dreambooth using DEAM dataset for emotion-conditioned music generation. This work encompasses three carefully structured experiments that each differ in several key hyperparameters with the goal of optimizing generating music spectrograms that offer emotional content and high fidelity vibes. In order to set a baseline, the first experiment evaluated the fundamental functionality of the model with a relatively low spectrogram resolution and batch size. Other experiments further increased the resolution and batch size, alongside better learning rate schedules and weight decay. This gave rise to minor changes as seen with respect to the loss curves and spectrograms, implying better stability and quality of music. Among these experiments, one of the most significant improvements in loss reduction and overfitting mitigation was shown in Experiment 3 where I used cosine learning rate scheduler and weight decay. Despite these advancements, the model still failed due to excess noise, artifacts in outputs and non-monotonic loss curves. Further, emotional alignment varied across categories, especially struggling to capture underrepresented emotions in the data. These results demonstrate the opportunities and limitations of using diffusion models for personalised emotional music generation.

### 6.2. Future Work

For enhancing the performance of generalization in future, the dataset must be expanded and balanced for allowing having more diverse emotional expressions and musical instances by feeding into the model. More powerful compute would also make it possible to evaluate more complex model architectures and larger batch sizes, which could lead to more stable and higher quality outputs. Perhaps working with original Riffusion model creators, or the open-source community will see new versions developed soon allowing for better denoising. Additional denoising and post-processing would decrease noise/artifacts while maintaining a higher signal fidelity and tamping down on any emotionless bloat in the generated music. With further focus on these aspects, future research can compensate for the current limitations of emotion-driven music synthesis with diffusion models and exploit the architecture as a whole to its fullest extent.



# References

- [1] Yunoki, I., Berreby, G., D'Andrea, N., Lu, Y., & Qu, X. (2024). Exploring AI Music Generation: A Review of Deep Learning Algorithms and Datasets for Undergraduate Researchers. In Proceedings of the 2024 International Conference on Artificial Intelligence and Music (pp. 123-135). Springer.
- [2] Briot, JP. From artificial neural networks to deep learning for music generation: history, concepts and trends. *Neural Comput & Applic* 33, 39–65 (2021).  
<https://doi.org/10.1007/s00521-020-05399-0>
- [3] Aljanaki, A., Yang, Y.-H., & Soleymani, M. (2017). Developing a benchmark for emotional analysis of music. *PLOS ONE*, 12(3), e0173392.  
<https://doi.org/10.1371/journal.pone.0173392>
- [4] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., & Norouzi, M. (2017). Neural audio synthesis of musical notes with WaveNet autoencoders. In Proceedings of the 34th International Conference on Machine Learning (pp. 1068–1077).
- [5] Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z. A., Dieleman, S., Elsen, E., Engel, J., & Eck, D. (2019). Enabling factorized piano music modeling and generation with the MAESTRO dataset. In Proceedings of the 7th International Conference on Learning Representations (ICLR)
- [6] Huang, C., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A. M., Hoffman, M. D., & Eck, D. (2019). Music Transformer: Generating Music with Long-Term Structure. Proceedings of the 7th International Conference on Learning Representations (ICLR).

- [7] Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., & Sutskever, I. (2020). Jukebox: A Generative Model for Music. <https://arxiv.org/abs/2005.00341>
- [8] Dong, H.-W., Hsiao, W.-Y., Yang, L.-C., & Yang, Y.-H. (2018). MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation. Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).
- [9] Huang, Y.-S., & Yang, Y.-H. (2020). Pop Music Transformer: Beat-based modeling and generation of expressive pop piano compositions. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 1180–1188).
- [10] Briot, J.-P. (2020). From artificial neural networks to deep learning for music generation: history, concepts, and trends. *Neural Computing and Applications*, 33, 39–65.
- [11] Dieleman, S., van den Oord, A., & Simonyan, K. (2018). The challenge of realistic music generation: Modelling raw audio at scale. In *Advances in Neural Information Processing Systems* (Vol. 31, pp. 7989–7999).
- [12] Engel, J., Agrawal, K. K., Chen, S., Gulati, S., Roberts, A., & Norouzi, M. (2017). WaveNet autoencoders for neural audio synthesis. <https://arxiv.org/abs/1704.01279>
- [13] Yang, L.-C., Chou, S.-Y., & Yang, Y.-H. (2017). MidiNet: A convolutional generative adversarial network for symbolic-domain music generation. In Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR) (pp. 324–331).
- [14] Agostinelli, A., Denk, T. I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., Huang, Q., Jansen, A., Roberts, A., Tagliasacchi, M., Sharifi, M., Zeghidour, N., & Frank, C. (2023). MusicLM: Generating music from text. <https://arxiv.org/abs/2301.11325>
- [15] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 10684–10695).

[16] Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., & Komatsuzaki, A. (2021). LAION-400M: Open dataset of CLIP-filtered 400 million image-text pairs. <https://arxiv.org/abs/2111.02114>

[17] Ho, J., & Salimans, T. (2021). Classifier-free diffusion guidance. In NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications. <https://arxiv.org/abs/2207.12598>

[18] Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., & Aberman, K. (2023). DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 10684–10695). <https://arxiv.org/abs/2208.12242>