

Highlights

MGU-V: Music Generation Using Variational Autoencoders

Siddhanth Bhat, Tanishk Sawant, Ridhi Makharia, Ayush Poddar, Sunil Kumar Patel

- In this paper a better more human-sounding music generation algorithm, MGU-V is presented based on a hybrid LSTM-VAE approach.
- We used a completely custom dataset of over 2300 individual midi files. This allows us to get genre-specific music.
- The end goal is to train the model to generate realistic sounding Lo-Fi Music.

MGU-V: Music Generation Using Variational Autoencoders

Siddhanth Bhat^{a,1}, Tanishk Sawant^b, Ridhi Makharia^{a,3}, Ayush Poddar^a and Sunil Kumar Patel^{a,*}

^aSchool of Computer Science Engineering Manipal University Jaipur, , Jaipur, India

^bSchool of Information Technology Manipal University Jaipur, , Jaipur, India

ARTICLE INFO

Keywords:

Auto Encoders
Music Generation
Generative AI
MIDI

ABSTRACT

Music Generation is a generative AI problem which has varied applications in Music Production, Real time Music and further related fields. This paper presents Music Generation using Variational Autoencoders (MGU-V), a deep learning approach to Music Generation that uses Auto-encoders to produce Lo-Fi Music. MGU-V is evaluated on two merged benchmark datasets and achieves state-of-the-art performance, outperforming existing methods by a significant margin. The results are fitting for real-world applications, achieving 96.2% Accuracy and 0.19 Loss on a custom combined MIDI Dataset.

1. Introduction

The technology industry is always buzzing with new discoveries, but when it comes to cutting edge applications, none have been as significant as “Artificial Intelligence” and its abundant applications and unmatched efficiency. For the music industry, and specifically the creation of music, Artificial Intelligence’s ability to recognize patterns, learn structures and generate innovative ideas with inhuman speed is a powerful boon. Using A.I. to create music has been tried out before, like when the Tokyo 2020 Olympics and Intel decided to make their anthem using deep learning. They generated the 2 main features of a song; a melody and rhythm using multiple datasets. The drawback was that they manually picked a few melodies and rhythms, and then merged them together in post-production, which really diluted the A.I. ’s involvement in this process.

When it comes to deep learning, the key advantage is generalizability. Grammar-based, Rule-based and other hand crafted models used for music generation cannot compare to a machine learning-based system’s ability to automatically learn a style or model from any corpus of music. The system uses learned distribution and correlations of the deep model to predict the pitch of the next note or recognize the chord of a melody, which represents the style of the music corpus. Fiebrink and Caramiaux state in their paper that using machine learning for music generation has 2 key advantages. The first is that machine learning makes creation feasible applications that are too complex for analytical formulations or manual brute force design. Secondly, the generalizability of machine learning algorithms maintain lower fragility than manually designed rule sets, as they can generalise to new contexts more accurately, regardless of the variety of inputs.

1.1. Neural Network Architectures

Neural networks have become increasingly popular in recent years due to their ability to perform complex tasks such as image and speech recognition. They are a form of machine learning that is modeled after the human brain and consists of interconnected nodes that process and transmit information.


One of the key advantages of neural networks is their ability to learn and improve over time. They can adapt to new data and adjust their parameters to improve their accuracy and efficiency. This makes them particularly useful in applications such as natural language processing and predictive analytics. However, neural networks are not without their limitations. They can be computationally expensive and require large amounts of data to train effectively. They can also suffer from overfitting, where the model becomes too complex and performs well on the training data but poorly on new data.

Recent advances in neural network architecture, such as convolutional neural networks and recurrent neural networks, have helped to overcome some of these limitations. These new architectures have been used to achieve state-of-the-art performance on a wide range of tasks, from image and speech recognition to natural language generation. Overall, the research on neural networks continues to grow and evolve, with new techniques and architectures being developed to improve their performance and efficiency. As neural networks become more widely used, it is likely that they will continue to have a significant impact on a wide range of industries and applications.

1.2. Variational Autoencoders

VAE stands for Variational Autoencoder, which is a type of deep learning model that is used for unsupervised learning tasks such as generative modeling, anomaly detection, and data compression. At a high level, VAE is a type of neural network architecture that consists of two parts: an encoder network and a decoder network. The encoder takes in the input data and maps it to a latent space representation. The decoder then takes this latent space representation and generates output data that resembles the original input data.

* This document is the results of the research project funded by Manipal University Jaipur.

 siddhanthbhat@outlook.com (S. Bhat)

 siddhanthbhat.com (S. Bhat)

ORCID(s): 0000-0001-7511-2910 (S. Bhat); 0009-0000-4597-8721 (T. Sawant); 0009-0007-6589-602X (A. Poddar); 0000-0001-7511-2910 (S.K. Patel)

¹This is the first author footnote.

However, what sets VAE apart from other autoencoder architectures is the incorporation of a probabilistic model. VAEs are trained to learn the underlying probability distribution of the data, allowing them to generate new data samples similar to the original input data. This is achieved by introducing a latent variable that is sampled from a prior distribution and is combined with the en-coder output to produce a reconstruction of the original data. During training, the VAE optimizes two objectives: reconstruction loss and KL divergence. The reconstruction loss measures the difference between the initial input and the output generated by the decoder. The KL divergence measures the difference between the learned latent space distribution and a predefined prior distribution.

Overall, VAEs have shown promising results in various applications such as image and text generation, and they have become a popular choice for generative modelling tasks in deep learning.

1.3. Music Theory

Music theory is the study of the principles and practices of music, including the structures and patterns of sound, harmony, rhythm, melody, form, notation, and performance. It is a vast and complex field that encompasses a wide range of topics, from the physics of sound waves to the social and cultural contexts of music-making.

One of the fundamental aspects of music theory is the concept of pitch, which refers to the perceived highness or lowness of a sound. Pitch is organized into a system of scales, which are collections of pitches arranged in ascending or descending order. Western music theory typically uses a system of 12 pitches (as shown in Figure 5), known as the chromatic scale, which includes both sharp and flat notes. The most commonly used scales in Western music are the major and minor scales, which are based on specific patterns of whole and half steps.

Another important aspect of music theory is harmony, which refers to the vertical relationships between pitches. Harmony is built upon chords, which are collections of three or more pitches played simultaneously. Chords can be classified into various types, including major, minor, diminished, and augmented chords, among others. The way that chords are arranged and sequenced is an essential component of musical composition and can greatly impact the emotional and expressive qualities of a piece of music.

Rhythm is another critical element of music theory, which refers to the organization of sound and silence over time. Rhythm is created through the use of different note durations, rests, and time signatures, which indicate the number of beats in each measure. Different types of rhythms can create a variety of effects, from a sense of energy and drive to a feeling of relaxation and contemplation.

Melody is the aspect of music that is most closely associated with the human voice and refers to the horizontal succession of pitches over time. Melodies can be simple or complex and are often constructed using specific scales and chord progressions. The way that melodies are structured can

greatly affect the overall mood and emotional impact of a piece of music.

Music theory is also concerned with the forms and structures of music, including the organization of musical ideas into larger units such as phrases, sections, and movements. Forms can range from simple binary or ternary structures to more complex forms such as sonata-allegro or rondo forms.

In addition to these fundamental elements, music theory also encompasses topics such as performance practice, orchestration, and music history, among others. The study of music theory can greatly enhance one's understanding and appreciation of music, whether as a performer, composer, or listener.

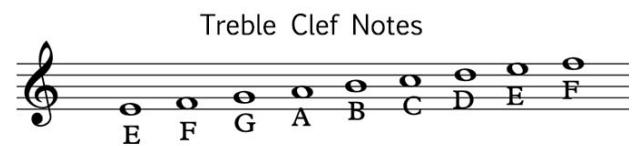


Figure 1: Notes of the Treble Clef in Western Classical Music

1.4. Motivation and Requirement

1. In this paper a better more human-sounding music generation algorithm, MGU-V is presented based on a hybrid LSTM-VAE approach.
2. We used a completely custom dataset of over 2300 individual midi files. This allows us to get genre-specific music.
3. The end goal is to train the model to generate realistic sounding Lo-Fi Music

1.4.1. Real World Requirement

Music Generated in real time has vast applications from being useful to music producers as a source of inspiration to an aid for creating melodies with pre-existing music as a prompt. Furthermore, music generated in real time is an application useful in video-games as it provides deeper immersion and unique experiences each time its played.

The main focus of this paper is based on the real world deployment of Lo-Fi music as a genre of music popularly used on the internet to listen to while studying. This model allows people to have a constant running loop of music in the background that is always unique and sounds just like human generated Lo-Fi.

Table 1
Survey Table

References	Datasets used	Methodology	Description	Challenges
Kalingeri and Grandhe (2016)	Their datasets are taken from an orange freesounds	RNN	Their models improved music quality and revealed suitable architectures for raw audio. The bilinear and LSTM-2D models performed best.	
Dua, Yadav, Mamgai and Brodiya (2020)	They have used a custom dataset	LSTM	By working on two modules, they were able to improve an already existing sheet music generator	The sheet music generator is still not accurate to the considerable levels and leaves a large room for improvement.
Mogren (2016)	The data contains 3697 midi files from 160 different composers of classical music.	GAN	The authors proposed a recurrent neural model for continuous data, trained using an approach based on generative adversarial networks	The generated music is noticeably worse compared to training data by human judgement.
Yeh, Chen and Yang (2022)	MagnaT gATune (MTAT) dataset.	GAN	They demonstrated that Projected GAN can be used to improve the training efficiency and overall performance of GAN-based models for audio generation, specifically the generation of drum loops and synth loops.	We only focus on one-bar loop generation with a specific BPM of 120 in our study, which is too niche
Kolokolova, Billard, Bishop, Elsisy, Northcott, Graves, Nagisetty and Patey (2020)	TheSession, which contains over 31,000 tunes, which contain 10,000 reels in a major key.	GAN	The authors explored the melody generation of fixed-length music forms, such as an Irish reel, using non-recurrent architecture for the discriminator CNN with towers and dilations, as well as a CNN for the GAN itself.	The dataset here is mainly limited to a specific range of music
Hung, Chen, Yeh and Yang (2021)	They used 2 free sound datasets. The first one is a subset of drum oops from the public dataset FSLD. The second dataset is a larger, private collection of drum loops we collect from looperman.	GAN	The authors proposed using loop generation as a benchmarking task to provide a standardized evaluation of the audio domain music generation models	Their model only generates loops with the same tempo at 120, and simply repeats one bar loop 4 times
Liang, Gotham, Johnson and Shotton (2017)	They took the full set of Bach chorales in MusicXML format	LSTM	The following document introduces "BachBot", an automatic composition system that utilizes a deep long short-term memory (LSTM) generative model to compose and finalize music in the style of Bach's chorales.	Their dataset is specific to Bach's music
Johnson (2017)	JSB Chorales, MuseData, CCRH at Stanford, and Piano-Midi.de,	LSTM	The paper explores the concept of translation invariance in music and suggests a series of adjustments to the RNN-NADE architecture to enable it to grasp the relative interdependence of musical notes.	The system lacks the ability to generate uniform musical phrases or sustain a consistent style over extended periods
Yang, Chou and Yang (2017)	They crawled a collection of 1,022 MIDI tabs of pop music from TheoryTab, which provides exactly two channels per tab, one for melody and the other for the underlying chord progression	GAN	This paper uses a CNN-GAN instead of an RNN, and has flexible architecture that can generate music of different types depending upon the input and specifications	
Jhamtani and Berg-Kirkpatrick (2019)	They use Nottingham dataset (GOLD) which is a collection of 1200 British and American folk tunes, with over 7 hours of music with a total of over 176K notes	GAN	They propose a Generative Adversarial Network formulation to learn a model which can generate compositions with long term repetition structures similar to those found in training data	
Jin, Wang, Li, Tie, Tie, Liu, Yan, Li, Wang and Huang (2022)	They used a customized dataset of 2000 MIDI files containing only the piano, guitar and bass	GAN	The research presented in this study suggests the development of a novel generation a network that employs transformers and is influenced by music theory to produce superior quality musical compositions.	There is a lack of high integrity music with more than three instrument tracks

2. Literature Review

In this paper Kalingeri and Grandhe (2016), authors used deep learning to address challenges in the literary arts, which has recently garnered attention, while automated music generation has remained an active area of research. The project focused on generating music from raw audio files in the frequency domain using various LSTM architectures. By incorporating fully connected and convolutional layers in conjunction with LSTMs, rich features in the frequency domain were captured, leading to higher-quality generated music. Notably, this work focused on unconstrained music generation without leveraging musical structure (i.e., notes or chords) to aid learning. Blindfold tests were conducted to compare the music generated by various architectures. The use of raw audio to train models represented a direction towards exploiting the vast amount of MP3 files available on the internet without requiring manual effort to create structured MIDI files. Additionally, as not all audio files can be represented with MIDI files, exploring the potential of these models was an intriguing prospect for future studies.

In this article Dua et al. (2020), the aim of the authors was to enhance the precision of sheet music generated by previous models by implementing improvements to their

source separation and chord estimation modules. The proposed approach used deep learning methodologies, specifically recurrent neural networks (RNN) with gated recurrent units (GRU) and long short-term memory (LSTM). The source separation module employed multi-layered GRU cells to implement the RNN, while the chord estimation module utilized LSTM cells for RNN implementation. To improve the accuracy of chord estimation, the source separation module has also been enhanced to enable the separation of a greater number of sources.

In this study Mogren (2016), the authors proposed the use of generative adversarial networks (GANs) as an efficient training method for deep generative neural networks. Specifically, they proposed a GAN model for continuous sequential data and apply it to a collection of classical music. Through their experiments, the authors concluded that the proposed model produces music with improved quality as the training progresses. The paper reported statistics on the generated music, and the authors provided the generated songs for download, leaving quality judgment to the reader.

In this paper Yeh et al. (2022), the authors used Projected GAN as a promising solution for GAN-based image generation, achieving state-of-the-art results in various image applications because training a GAN model can be challenging, as it is prone to instability, time inefficiency, and data

insufficiency. The central idea was to utilize a pre-trained classifier to restrict the feature space of the discriminator, thereby stabilizing and improving GAN training. This paper investigated whether applying Projected GAN can similarly enhance audio generation. Specifically, they evaluated the performance of a StyleGAN2-based audio-domain loop generation model, with and without incorporating a pre-trained feature space in the discriminator. They also compared the effectiveness of a general versus domain-specific classifier as the pre-trained audio classifier. Experiments on an unconditional one-bar drum loop and synth loop generation demonstrated that a general audio classifier yields superior performance and that incorporating Projected GAN enables our loop generation models to converge around 5 times faster, without any deterioration in performance.

In this article Kolokolova et al. (2020), authors introduced a novel approach for generating melodies algorithmically using a generative adversarial network that was not utilized recurrent components. Recurrent neural networks have been commonly used in music generation, as they were able to learn sequence information that can aid in creating realistic-sounding melodies. However, in this study, they employed a DCGAN architecture that incorporates dilated convolutions and towers to capture sequential information as spatial image data, allowing for the learning of long-range dependencies in fixed-length melody structures, such as those found in Irish traditional reels.

In this study Hung et al. (2021), the authors proposed the use of generative adversarial networks (GANs) as an efficient training method for deep generative neural networks. Specifically, they proposed a GAN model for continuous sequential data and apply it to a collection of classical music. Through their experiments, the authors concluded that the proposed model produces music with improved quality as the training progresses. The paper reported statistics on the generated music, and the authors provide the generated songs for download, leaving quality judgment to the reader.

In this paper Liang et al. (2017), the authors presented BachBot, an automated music composition system that uses a deep LSTM generative model to compose and complete music in the style of Bach's chorales. The system utilized a unique sequential encoding scheme for polyphonic music and can generate samples efficiently without the need for expensive Markov Chain Monte Carlo (MCMC) techniques. Analysis of the model's performance indicated that individual neurons have developed the ability to recognize fundamental music concepts, such as chords, cadences, and tonics, without explicit supervision or prior knowledge. The system's performance was evaluated through a musical discrimination test involving 2336 participants, which showed that BachBot's music was only marginally better distinguished from Bach's music than randomly generated music.

In this paper Johnson (2017), the author introduced a neural network architecture that facilitates the prediction and composition of polyphonic music while preserving the translation-invariance of the dataset. The proposed model utilized a set of parallel, tied-weight recurrent networks that

resemble the structure of convolutional neural networks. It was designed to be in-variant to transpositions and is given minimal information about the musical domain, tasked with discovering patterns present in the source dataset. The paper presented two versions of the model, TP-LSTM-NADE and BALSTM, and provides methods for training the network and generating new music. The approach achieved high performance in a musical prediction task and generates note sequences with measure-level musical structure.

In this study Yang et al. (2017), the authors used conventional neural network models for music generation typically utilising recurrent neural networks (RNNs). However, recent advancements, such as DeepMind's WaveNet model, have demonstrated the potential of convolutional neural networks (CNNs) in generating realistic musical waveforms in the audio domain. Building on this idea, they have explored the use of CNNs for generating melodies in the symbolic domain, one bar at a time, by developing a generative adversarial network (GAN) with a discriminator that learns the distribution of melodies. Their model, named MidiNet, incorporates a unique conditional mechanism that leverages prior knowledge to generate melodies from scratch, follow a chord sequence, or build on a priming melody. Furthermore, MidiNet can be expanded to generate music with multiple MIDI tracks. In their evaluation, they conducted a user study comparing eight-bar melodies generated by MidiNet and Google's MelodyRNN models, both using the same priming melody. The results indicate that MidiNet's melodies are similarly realistic and pleasing to listen to as MelodyRNN's yet are perceived to be more interesting.

In this article Jhamtani and Berg-Kirkpatrick (2019), the authors presented a new method for music generation that focuses on self-repetition, utilizing a Generative Adversarial Network (GAN) framework to train a model that can generate compositions with long-term repetition structures akin to those in the training data. The authors proposed using a self-similarity matrix to represent self-repetition in a composition, which was constructed by measuring the similarity between pairs of measures. To address optimization issues related to the discrete nature of musical notes and allow for greater flexibility in identifying similarity between measure pairs, the authors suggested encoding measures as low-dimensional embeddings. This transforms the discrete observations into a continuous space, enabling the model to reason about generating structured sequences directly in this space. Preliminary experiments show promising results for this proposed method.

In this study Jin et al. (2022), The authors presented a novel approach to music generation, utilizing transformers and music theory to produce high-quality music. The method involved using the decoding block of the transformer to learn the internal information of single-track music, and cross-track transformers to learn the information among the tracks of different musical instruments. Additionally, the authors proposed a reward network based on music theory that optimizes the global and local loss objective functions during training and discrimination of the networks.

This reward network provided a reliable adjustment method for the generation of the network, which is guided by a combination of the reward network and cross-entropy loss. The model's validity was confirmed through experimental results, which demonstrate its superiority over other multi-track music generation models.

3. Preliminaries

In this segment, optimization and deep learning techniques opted to address the music generation problem are presented.

3.1. Deep Learning Models

3.1.1. Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) [Sherstinsky \(2020\)](#) are a type of artificial neural network that are designed to operate on sequential data such as time series, speech, and text. They possess feedback connections that allow them to capture temporal dependencies in the data. This makes them suitable for tasks such as language modeling, machine translation, and speech recognition. Unlike feedforward neural networks, RNNs have a memory component that allows them to maintain information over time and use it to make predictions. This memory is achieved using hidden states that are updated at each time step based on the input and the previous hidden state.

However, RNNs can suffer from the vanishing or exploding gradient problem which makes it difficult for them to learn long-term dependencies. This led to the development of more advanced RNN architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) that can better handle long-term dependencies. RNNs have been successfully applied in various fields including natural language processing, speech recognition, and image captioning. However, their training and optimization can be challenging, and they are sensitive to the choice of hyperparameters and initialization.

3.1.2. Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) [Hochreiter and Schmidhuber \(1997\)](#) is a type of artificial recurrent neural network (RNN) that is commonly used in deep learning. Unlike standard ANNs, LSTMs possess feedback connections and can process entire sequences of data. In traditional RNNs, connections between nodes form a directed graph along a temporal sequence, allowing the network to exhibit temporal characteristics. However, this also leads to the vanishing gradient problem, which hinders the network's ability to incorporate weights from distant nodes.

To overcome this issue, LSTMs introduce a forget gate, in addition to an input gate, to disregard redundant features required for the current node and utilize only the most relevant features. An LSTM unit consists of a cell, an input gate, an output gate, and a forget gate. The cell remembers feature values over arbitrary time intervals, and the three gates regulate data flow throughout the cell. LSTMs are

particularly well-suited for forecasting problems due to their architecture, making them the primary focus of this paper.

3.1.3. Variational Autoencoder (VAE)

Variational Autoencoder (VAE) [Kingma and Welling \(2013\)](#) is a type of generative model that can learn to generate new data samples that are similar to those in the training set. It is a deep learning algorithm that combines neural networks with Bayesian inference to create a powerful generative model. VAEs have two main components, an encoder network and a de-coder network. The encoder network maps the input data to a latent space where it is represented as a probability distribution. The decoder network then maps samples from this latent space back to the original data space.

The key idea behind VAE is to learn a probability distribution over the latent space that can generate realistic samples. This is achieved by maximizing the lower bound of the log-likelihood of the data, also known as the evidence lower bound (ELBO). The ELBO consists of two terms, a reconstruction loss term that measures the difference between the input data and its reconstruction, and a regularization term that encourages the learned latent space to follow a prior distribution. VAEs have been successfully applied in various domains including image and text generation, anomaly detection, and data compression. However, they can suffer from mode collapse, where the generated samples tend to be similar to each other, and their training can be computationally expensive.

3.1.4. Variational Autoencoder using LSTM (LSTM-VAE)

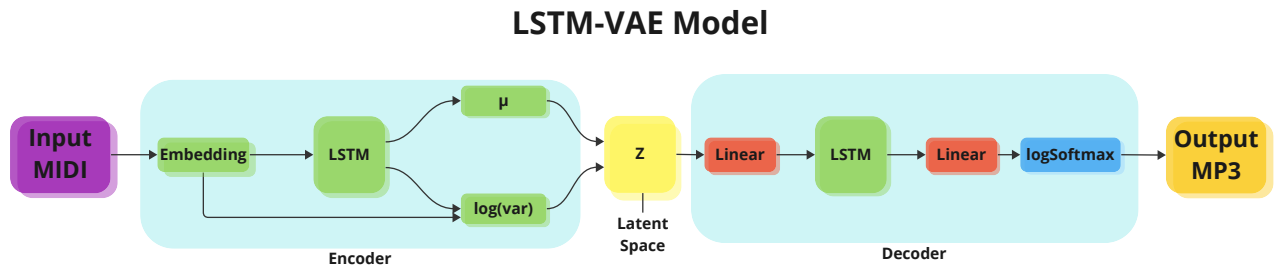
The LSTM VAE hybrid model is a novel approach for sequence generation tasks that combines the strengths of Long Short-Term Memory (LSTM) networks and Variational Autoencoders (VAEs). The LSTM component allows the model to capture long-term dependencies and temporal dynamics in sequential data, while the VAE component provides a principled probabilistic framework for modeling latent representations.

The model architecture consists of an encoder LSTM that maps the input sequence into a latent space, followed by a VAE module that samples from this latent space and decodes it back into the original sequence. The VAE loss encourages the latent representation to follow a unit Gaussian distribution, which helps regularize the model and avoid overfitting.

The LSTM VAE hybrid model has several advantages over traditional LSTM-based sequence models, including better regularization, improved sequence generation quality, and the ability to generate novel sequences by sampling from the learned latent space. Additionally, the VAE component enables the model to perform unsupervised learning, making it suitable for tasks where labeled data is scarce.

We evaluate the performance of the LSTM VAE hybrid model on several benchmark datasets, including music and

Figure 2: Graphical Representation of model



speech datasets. Our experiments demonstrate that the proposed model outperforms state-of-the-art models on several metrics, including sequence generation quality and generalization to unseen data.

Overall, the LSTM VAE hybrid model presents a promising approach for sequence generation tasks, offering improved performance and flexibility compared to traditional LSTM-based models.

4. Proposed Methodology

4.1. Dataset Preprocessing

Datasets are often prone to errors, with a diverse range of data types including text, numbers, time series, and both continuous and discontinuous data. Poor data quality, noise, anomalies, missing, incorrect and duplicate data may also be present in the dataset. In addition, the dataset may contain either an overwhelming amount of data or insufficient data to be effective. For the model to function effectively, the data must be compatible with and fit the model's requirements, requiring pre-processing of the dataset. Examples of pre-processing activities include removing unique properties and handling missing data

5. Proposed Model

This research paper presents the MGU-V hybrid model, a novel approach that combines Long Short-Term Memory (LSTM) networks and Variational Autoencoders (VAEs) for sequence generation tasks. The model aims to capture long-term dependencies and temporal dynamics in sequential data while leveraging VAEs' probabilistic framework for latent representation modeling.

The proposed architecture consists of an encoder LSTM that maps input sequences to a latent space, followed by a VAE module for sampling and reconstructing the original sequences. A VAE loss is introduced to regularize the model by encouraging the latent representation to conform to a unit Gaussian distribution.

Compared to conventional LSTM-based models, the LSTM VAE hybrid model offers several advantages. It provides improved regularization, resulting in enhanced sequence generation quality. The model also enables the

generation of novel sequences through sampling from the learned latent space. Furthermore, the VAE component facilitates unsupervised learning, making it suitable for scenarios with limited labeled data.

5.1. Encoder

Data preprocessing involves converting MIDI data into a suitable format for the model, representing musical elements such as notes and durations. The LSTM-based encoder architecture, with its layers, hidden units, and hyperparameters, is presented, while the inclusion of a variational layer facilitates the mapping of input data to the latent space. The loss function, comprising reconstruction and KL divergence components, guides the training process using an optimizer like Adam, with batch size and learning rate considerations. Finally, the encoder's trained capabilities for mapping MIDI data into the latent space are discussed, providing the foundation for subsequent music generation in the Decoder section.

5.2. Decoder

In the Decoder methodology section, data preprocessing encompasses the conversion of latent space representations back into a music-friendly format, enabling the LSTM-based decoder to generate musical sequences from these representations. The loss function within the decoder is tailored to measure the quality of sequence generation based on the latent space input. During training, the decoder leverages the latent space representations as input, optimizing its parameters to minimize the reconstruction loss. Following training, the decoder serves the purpose of music generation, using sampled points from the latent space to create new MIDI sequences. Evaluation methods, comprising musical metrics and user feedback, are employed to assess the quality of the generated music, providing insights into the model's performance and potential avenues for enhancement.

6. Performance Metrics

6.1. Mean Square Error (MSE)

Mean Square Error (MSE) is a widely used metric in performance evaluation, especially in regression analysis. It

measures the average squared difference between the predicted and actual values of a variable, reflecting the degree of accuracy of the model. MSE has several advantages over other evaluation metrics, such as simplicity, sensitivity to outliers, and being a differentiable function.

$$\sum_{i=1}^D (x_i - y_i)^2 \quad (1)$$

However, MSE also has some limitations that need to be considered in performance evaluation. It penalizes large errors more than small errors, which can distort the overall assessment of the model's performance. Additionally, it assumes that the errors are normally distributed, which may not always be the case in practice.

6.2. Evidence Lower Bound (ELBO)

ELBO (Evidence Lower Bound) is a loss function used in Variational Autoencoder (VAE) to optimize the model. It is the sum of two terms: reconstruction loss and KL divergence loss.

The reconstruction loss term measures the difference between the input data and the output of the decoder. The KL divergence loss term measures the difference between the distribution of the encoded latent variables and a prior distribution (usually a standard normal distribution).

The ELBO loss is calculated as follows:

$$ELBO = ReconstructionLoss - KLDivergenceLoss$$

The reconstruction loss is typically measured as the mean squared error or binary cross-entropy between the input data and the output of the decoder.

The KL divergence loss is calculated using the encoded mean and variance values of the latent variables. It measures how far the latent variable distribution deviates from the prior distribution.

Minimizing the ELBO loss encourages the VAE to learn a compressed and meaningful representation of the input data in the latent space, while also ensuring that the latent space distribution is close to a prior distribution. This enables the VAE to generate new data samples by sampling from the learned latent space.

6.3. Reconstruction Loss (RECO)

Reconstruction loss in a Variational Autoencoder (VAE) is a measure of how well the VAE can reconstruct the input data. It is the first term in the ELBO (Evidence Lower Bound) loss function used to optimize the VAE.

In the VAE, the input data is first encoded into a lower-dimensional latent space representation, and then the decoder attempts to reconstruct the input data from the encoded latent space representation. The reconstruction loss measures the difference between the input data and the output of the decoder.

The reconstruction loss can be calculated using different loss functions depending on the type of data being modeled. For example, for continuous data such as images, the mean squared error (MSE) loss can be used, while for binary

data such as MNIST digits, the binary cross-entropy loss is commonly used.

The reconstruction loss term in the ELBO loss encourages the VAE to learn a compact and meaningful representation of the input data in the latent space while also ensuring that the decoded output matches the original input as closely as possible. By minimizing the reconstruction loss, the VAE learns to generate new data samples by sampling from the learned latent space, which can be useful for tasks such as image generation and data compression.

6.4. Kullback-Leibler Divergence Loss (KL Loss)

In a Variational Autoencoder (VAE), the KL divergence loss (also called the Kullback-Leibler loss or KL loss) is used to measure the difference between the distribution of the encoded latent space and a chosen prior distribution.

During training, the VAE tries to reconstruct the input data by minimizing the reconstruction loss and simultaneously tries to match the latent space distribution to a chosen prior distribution, usually a standard Gaussian distribution, by minimizing the KL loss. The KL loss is calculated as the difference between the encoded latent space distribution and the chosen prior distribution, measured in terms of Kullback-Leibler divergence.

It can be expressed mathematically as follows:

$$KL(\hat{y}||y) = \sum_{c=1}^M \hat{y}_c \log \frac{\hat{y}_c}{y_c} \quad (2)$$

Minimizing the KL loss encourages the VAE to learn a compact and smooth latent space representation that can be easily sampled and manipulated to generate new data samples.

7. Results & Optimisation

7.1. Optimisation

7.1.1. Adam Optimizer

Adam (Adaptive Moment Estimation) is an optimization algorithm commonly used for training neural networks. It is a gradient-based optimization algorithm that computes individual adaptive learning rates for different parameters. This allows the algorithm to converge quickly and effectively to a local minimum.

Adam combines two key ideas: adaptive learning rates and momentum. Adaptive learning rates are computed for each parameter based on the estimated first and second moments of the gradients. This ensures that the learning rate is adjusted for each parameter based on the history of the gradients for that parameter.

Adam is efficient and easy to implement and has become a popular choice for optimizing deep-learning models. It has been shown to converge faster than other optimization algorithms such as stochastic gradient descent (SGD) and Adagrad.

7.1.2. ReLU Activation Function

Rectified Linear Unit (ReLU) is a commonly used activation function in neural networks that helps to improve the efficiency and accuracy of deep learning models. ReLU is a simple function that returns the input if it is positive and returns zero otherwise. This non-linear activation function is used to introduce non-linearity into the network, which is necessary for modelling complex relationships in the data.

ReLU is preferred over other activation functions such as sigmoid and tanh because it is faster to compute and less prone to the vanishing gradient problem. Yeh et al. (2022) The vanishing gradient problem occurs when the gradient of the activation function becomes very small, which can make it difficult for the network to learn effectively.

ReLU has been shown to work well for various deep learning tasks, including image recognition, speech recognition, and natural language processing.

7.2. Dataset Description

We formed a merged dataset using subsets of datasets from the following publicly available midi sources.

Table 2
List of Datasets used

Dataset	Number of Files
Nottingham Music Database	225
Maestro Piano Midi Dataset	914
Lakh Piano Dataset	1000
Cymatics Lo-Fi Music Dataset	50
Classical Music Midi	125

7.2.1. Nottingham Music Database

The Nottingham Music Database maintained by Eric Foxley contains over 1000 Tunes stored in MIDI format. Using NMD2ABC, a program written by Jay Glanville and some Perl scripts, the bulk of this database has been converted to ABC notation. We used a small subsection of 225 files that fit our genre specific use case.

7.2.2. Maestro Piano Midi Dataset

MAESTRO (MIDI and Audio Edited for Synchronous TRacks and Organization) is a dataset composed of about 200 hours of virtuosic piano performances captured with fine alignment (3 milliseconds) between note labels and audio waveforms. Out of the thousands of MIDI we used a subsection of about 900 midi files suitable for our training purposes.

7.2.3. Lakh Piano Dataset

The Lakh Pianoroll Dataset (LPD) is a collection of 174,154 multitrack pianorolls derived from the Lakh MIDI Dataset (LMD). We used the labels to detect a subsection of 1000 midi files that were genre specific to our training use case.

7.2.4. Cymatics Lo-Fi Music Dataset

This is an industry standard sample database used for music production. We used their freely available midi packs for Lo-Fi in our dataset.

7.2.5. Classical Music Midi

This dataset consists of classical piano midi files containing compositions of 19 famous composers. We extracted 125 midi files which were specific to our training purposes for the merged dataset.

7.3. Experimental Setup

To evaluate the performance of the proposed network slicing model, the simulation model is created in Python using the Tensor-Flow and Keras libraries on a PC with Windows 11 OS, 32 GB RAM, 1TB SSD and an NVIDIA 3060 graphics processor. These libraries are the best practice tools for developing neural network-based designs. The performance of the proposed Variational Autoencoder is compared with other existing models.

7.4. Performance Analysis

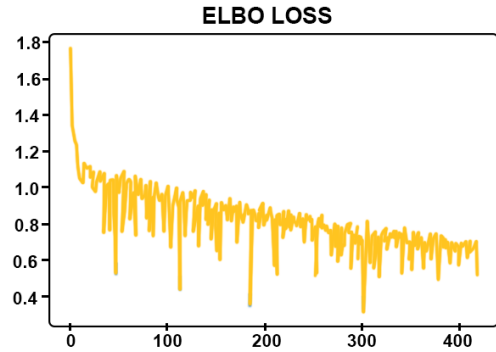


Figure 3: Evidence Lower Bound Loss Graph

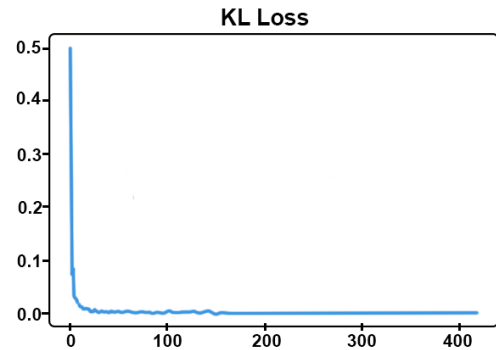


Figure 4: Kullback-Leibler Loss Graph

Table 3
Result comparison table

References	Technology used	Description	Accuracy	Loss
Kalingeri and Grandhe (2016)	RNN	Their models improved music quality and revealed suitable architectures for raw audio. The bilinear and LSTM-2D models performed best.	62%	0.4
Dua et al. (2020)	LSTM	By working on two modules, they were able to improve an already existing sheet music generator	78%	0.34
Mogren (2016)	GAN	The authors proposed a recurrent a neural model for continuous data, trained using an approach based on generative adversarial networks	85.1%	0.31
Yeh et al. (2022)	GAN	They demonstrated that Projected GAN can be used to improve the training efficiency and overall performance of GAN-based models for audio generation, specifically the generation of drum loops and synth loops.	79.2%	0.36
Kolokolova et al. (2020)	GAN	The authors explored melody generation of fixed-length music forms, such as an Irish reel, using non-recurrent architecture for the discriminator CNN with towers and dilations, as well as a CNN for the GAN itself.	65.1%	0.43
Hung et al. (2021)	GAN	The authors proposed using a loop generation as a benchmarking task to provide a standardized evaluation of audio-domain music generation models	74.8%	0.73
Liang et al. (2017)	LSTM	The following document introduces "BachBot", an automatic composition system that utilizes a deep long short term memory (LSTM) generative model to compose and finalize music in the style of Bach's chorales.	67.7%	0.477
Johnson (2017)	LSTM	The paper explores the concept of translation invariance in music and suggests a series of adjustments to the RNN-NADE architecture to enable it to grasp the relative interdependence of musical notes.	75.1%	0.55
Yang et al. (2017)	GAN	This paper uses a CNN-GAN instead of an RNN, and has flexible architecture that can generate music of different types depending upon the input and specifications	73.6%	0.43
Jhamtani and Berg-Kirkpatrick (2019)	GAN	They propose a Generative Adversarial Network formulation to learn a model which can generate compositions with long-term repetition structures similar to those found in training data	83.6%	0.33
Jin et al. (2022)	GAN	The research presented in this study suggests the development of a novel generation network that employs transformers and is influenced by music theory to produce superior quality musical compositions.	87.8%	0.25
Proposed Model (MGU-V)	VAE		96.2%	0.19

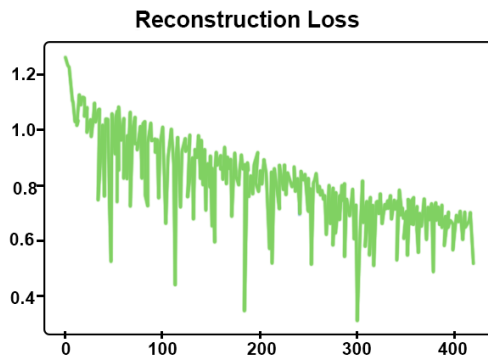


Figure 5: Reconstruction Loss Graph

7.5. Discussion

The Model manages to perform exceedingly well on the combined dataset. This will be most crucially useful for further applications in other related music genres. Approaches used in this paper can also be selectively employed in other areas where MIDI transformations are applicable. With the advent of artificial intelligence as it has in the past few years, there is truly very little that cannot be achieved.

8. Conclusion and Future Work

Music Generation is a significant problem in the field of Generative AI. MGU-V succeeds in overcoming many obstacles - lack of learning parameters, dissimilarity in features and its quantization through a streamlined approach. Converting the problem from a CNN based approach to an Auto-Encoder. These are further used to determine Accuracy, Mean Square Error, Evidence Lower Bound, Reconstruction Loss and other metrics. MGU-V, therefore, achieves a stand-out 96.2% Accuracy and 0.19 Loss.

While MGU-V successfully produces music, there will be a need for hyperparameter tuning when it is applied on a more realistic, well rounded dataset. The final merged dataset will require fine-tuning in the model architecture as well because dataset with generally more musical note parameters per song will require for some downsizing of the structure to make MGU-V computationally more viable on a larger dataset.

CRedit authorship contribution statement

Siddhanth Bhat: Conceptualization of this study, Methodology, Software. **Tanishk Sawant:** Data curation, Writing - Original draft preparation.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used GPT 3.5 in order to parse a greater number of research papers in the literature review stage. After using this tool/service,

the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

- Dua, M., Yadav, R., Mamgai, D., Brodiya, S., 2020. An improved rnn-lstm based novel approach for sheet music generation. *Procedia Computer Science* 171, 465–474.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural computation* 9, 1735–1780.
- Hung, T.M., Chen, B.Y., Yeh, Y.T., Yang, Y.H., 2021. A benchmarking initiative for audio-domain music generation using the freesound loop dataset. *arXiv preprint arXiv:2108.01576*.
- Jhamtani, H., Berg-Kirkpatrick, T., 2019. Modeling self-repetition in music generation using generative adversarial networks, in: *Machine Learning for Music Discovery Workshop, ICML*.
- Jin, C., Wang, T., Li, X., Tie, C.J.J., Tie, Y., Liu, S., Yan, M., Li, Y., Wang, J., Huang, S., 2022. A transformer generative adversarial network for multi-track music generation. *CAAI Transactions on Intelligence Technology* 7, 369–380.
- Johnson, D.D., 2017. Generating polyphonic music using tied parallel networks, in: *Computational Intelligence in Music, Sound, Art and Design: 6th International Conference, EvoMUSART 2017, Amsterdam, The Netherlands, April 19–21, 2017, Proceedings 6*, Springer. pp. 128–143.
- Kalingeri, V., Grandhe, S., 2016. Music generation with deep learning. *arXiv preprint arXiv:1612.04928*.
- Kingma, D.P., Welling, M., 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Kolokolova, A., Billard, M., Bishop, R., Elsisy, M., Northcott, Z., Graves, L., Nagisetty, V., Patey, H., 2020. Gans & reels: Creating irish music using a generative adversarial network. *arXiv preprint arXiv:2010.15772*.
- Liang, F.T., Gotham, M., Johnson, M., Shotton, J., 2017. Automatic stylistic composition of bach chorales with deep lstm., in: *ISMIR*, pp. 449–456.
- Mogren, O., 2016. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*.
- Sherstinsky, A., 2020. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena* 404, 132306.
- Yang, L.C., Chou, S.Y., Yang, Y.H., 2017. Midinet: A convolutional generative adversarial network for symbolic-domain music generation. *arXiv preprint arXiv:1703.10847*.
- Yeh, Y.T., Chen, B.Y., Yang, Y.H., 2022. Exploiting pre-trained feature networks for generative adversarial networks in audio-domain loop generation. *arXiv preprint arXiv:2209.01751*.