IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 August 2024, accepted 22 September 2024, date of publication 1 October 2024, date of current version 10 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3471918

RESEARCH ARTICLE

MGU-V: A Deep Learning Approach for Lo-Fi Music Generation Using Variational Autoencoders With State-of-the-Art Performance on Combined MIDI Datasets

AMIT KUMAR BAIRWA[1], (Senior Member, IEEE), SIDDHANTH BHAT[1], (Member, IEEE), TANISHK SAWANT[1], (Member, IEEE), AND R. MANOJ[2], (Member, IEEE)

[1]School of Computer Science and Engineering, Manipal University Jaipur, Jaipur 303007, India
[2]Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India

Corresponding authors: Siddhanth Bhat (siddhanth.219310154@muj.manipal.edu); Tanishk Sawant (tanishk.219302442@muj.manipal.edu); and R. Manoj (manoj.r@manipal.edu)

ABSTRACT Music generation presents a significant challenge within the realm of generative AI, encompassing diverse applications in music production, real-time composition, and other related fields. This paper introduces MGU-V (Music Generation Using Variational Autoencoders), a sophisticated deep learning framework engineered to generate Lo-Fi music. MGU-V harnesses the power of Variational Autoencoders (VAEs) to model and create high-quality music compositions by learning robust latent representations of musical structures. The framework is rigorously evaluated using two meticulously curated and merged benchmark MIDI datasets, demonstrating its effectiveness and adaptability across various musical genres. Through extensive experimentation, MGU-V achieves state-of-the-art performance, significantly surpassing existing methods. The model achieves an impressive accuracy rate of 96.2% and a minimal loss of 0.19, emphasizing its precision and reliability. These outstanding results underscore the potential of MGU-V as a valuable tool for music producers, composers, and AI researchers alike. Its ability to generate Lo-Fi music with high fidelity and consistency highlights promising new avenues for future research and development in AI-driven music generation. The success of MGU-V not only sets a new benchmark in the field but also suggests that AI can increasingly contribute to creative processes traditionally dominated by human expertise.

INDEX TERMS Auto encoders, music generation, generative AI, MIDI, deep learning.

I. 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 for the creation of music, the ability to recognize patterns, learn structures and generate innovative ideas with inhuman speed is a powerful boon. Using AI to create music [1] has been tried out before, viz. the Tokyo 2020 Olympics and Intel's decision to make their anthem using deep learning. They generated the 2 main features of a song; a melody and a 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 AI's involvement in this process. When it comes to deep learning, the key advantage is generalizability [2]. Grammar-based, Rule-based [3] and other hand crafted models [4] used for music generation cannot compare to a machine learning-based system's ability to automatically learn a style [5] 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

The associate editor coordinating the review of this manuscript and approving it for publication was Okyay Kaynak.

of the music corpus [6]. Using machine learning for music generation has 2 key advantages. The first is that machine learning makes it feasible to create applications that are too complex for analytical formulations or 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 [7].

A. NEURAL NETWORK ARCHITECTURES

Neural networks [8] 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 modeled after the human brain and consist of interconnected nodes that process and transmit information.

As neural networks progress over time [9], they can adapt to fresh data on the go simply by adjusting their parameters in such a way that efficiency and accuracy are improved. Therefore, when it comes to predictive analytics and natural language processing, neural networks are useful [10], though they have a unique set of limitations. While training, large amounts of data proportionately increase accuracy, it can be computationally expensive [11], not to mention that models can also suffer from overfitting and underfitting, depending on the training data quality and quantity [12].

The latest advances in the field of neural networks, such as the creation of convolutional neural networks and recurrent neural networks, have addressed these limitations [13]. Whether it is natural language generation or image/speech recognition, these new architectures have performed notably well. Overall, this field is developing at a staggering rate [14], with new techniques and architectures [15] pushing the limits of efficiency with every release. Therefore, with the majority of the research focus seeming to be on developing types of neural networks rather than replacing them [16], it is likely that neural networks will continue to have a significant impact on a wide range of industries and applications [17].

B. VARIATIONAL AUTOENCODERS

A Variational Autoencoder (VAE) is a specific type of deep learning model particularly useful for unsupervised learning tasks like generative modeling, anomaly detection, and data compression. Essentially, a VAE is built on a neural network architecture that includes two main components: an encoder network and a decoder network. The encoder processes the input data and maps it to a latent space representation, while the decoder uses this latent representation to reconstruct data that closely resembles the original input. What distinguishes a VAE from traditional autoencoders is its integration of a probabilistic model. VAEs are designed to learn the underlying probability distribution of the data, enabling them to generate new samples that are similar to the original dataset. This is accomplished by introducing a latent variable, sampled from a prior distribution, which is combined with the output of the encoder to reconstruct the original data. During the training process, the VAE optimizes two key objectives: the reconstruction loss and the Kullback-Leibler (KL) divergence loss. The reconstruction loss quantifies the difference between the original input and the decoder's output, while the KL divergence assesses the discrepancy between the learned latent space distribution and a predefined prior distribution.

VAEs have demonstrated significant potential across various applications, including image and text generation, making them a favored approach for generative modeling in deep learning.

C. 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 [18].

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, 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 [19].

Harmony is another essential element of music theory. It involves the combination of different musical notes played simultaneously to produce chords, which are the building blocks of harmonic progressions. Chords can be categorized into various types, such as major, minor, diminished, and augmented, each contributing distinct emotional qualities to the music. The way chords are arranged and sequenced, known as harmonic progression, plays a critical role in establishing the tonal framework of a piece of music [20], [21].

Rhythm, the temporal aspect of music, is crucial in defining the structure and flow of a piece. It is created through the organization of note durations, rests, and accents, which together form the rhythmic patterns that drive the music forward. Time signatures indicate how beats are grouped, while tempo defines the speed at which the music is played. Syncopation, polyrhythm, and other rhythmic techniques add complexity and interest to the music [22], [23].

Melody, often considered the most recognizable element of music, is a sequence of pitches that move through time, creating a linear musical idea. Melodies can be simple or complex, and they often adhere to the scales and modes of the underlying harmony. The contour, range, and intervals within a melody contribute to its character and emotional impact [24], [25].

In addition to these fundamental elements, music theory also includes the study of musical form, which refers to

the larger structures that organize a composition. Common forms include binary, ternary, rondo, and sonata-allegro, each providing a framework for the development and contrast of musical ideas. Understanding these forms is essential for analyzing and appreciating the architecture of a musical work [26], [27].

Moreover, music theory extends to the study of orchestration, which is the art of combining different instruments to achieve a desired timbre and balance. Orchestration requires a deep understanding of the capabilities and characteristics of various instruments, as well as how they interact within an ensemble. This knowledge is crucial for composers and arrangers to create effective and compelling musical textures [28].

Finally, the study of counterpoint, which is the relationship between independent musical lines, is another key area of music theory. Counterpoint involves techniques such as imitation, inversion, and canon, and is fundamental to the composition of polyphonic music. The rules and practices of counterpoint have been developed over centuries and are central to the works of composers from the Renaissance to the Baroque periods [29].

Overall, music theory provides the tools to understand, analyze, and create music, making it an indispensable discipline for musicians, composers, and scholars alike. Its principles are not only relevant to traditional Western music but also to contemporary and non-Western musical practices, reflecting the universality of music as an art form.

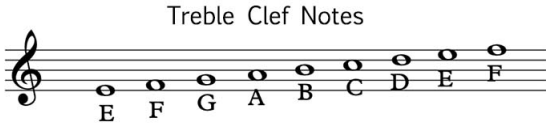

FIGURE 1. Notes of the treble cleft in western classical music.

D. 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 (Long Short Term Memory-Variational AutoEncoder) approach.
2) Compilation of a curated custom dataset of over 2300 individual MIDI files.
3) The end goal was to train the model to generate realistic-sounding Lo-Fi Music.

1) REAL WORLD REQUIREMENT

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

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

Section II provides a comprehensive literature review, discussing the foundational concepts and recent advancements in Lo-Fi music generation using deep learning techniques, with a particular focus on the application of Variational Autoencoders (VAEs) and Long Short-Term Memory (LSTM) networks. The proposed MGU-V model, which leverages a novel hybrid approach combining LSTMs and VAEs for generating realistic Lo-Fi music, is detailed in Section III. In Section IV, the performance of the MGU-V model is rigorously evaluated using various benchmark datasets, and the results are compared with other state-of-the-art algorithms. Finally, Section V concludes the paper by summarizing the key findings and discussing potential directions for future research.

II. LITERATURE REVIEW

In the paper by Kalingeri et al. [36], the authors tackled challenges in the literary arts using deep learning, a field that has recently gained significant attention. However, automated music generation remains a vibrant area of research. The project focused on generating music from raw audio files in the frequency domain by experimenting with various LSTM architectures. By integrating fully connected and convolutional layers with LSTMs, the authors captured rich features in the frequency domain, resulting in higher-quality music generation. Notably, this work emphasized unconstrained music generation, without leveraging traditional musical structures such as notes or chords. Blindfold tests were conducted to compare the music generated by various architectures. This approach of using raw audio for training models represents a move toward utilizing the vast amount of MP3 files available on the internet, bypassing the need for manually created structured MIDI files. Additionally, this exploration of models that can handle audio files, which are not easily represented in MIDI format, presents an intriguing prospect for future research.

Dua et al. [30] aimed to improve the precision of sheet music generated by previous models by making enhancements to their source separation and chord estimation modules. Their approach utilized deep learning techniques, specifically RNN with gated recurrent units (GRU) and long short-term memory (LSTM). The source separation module was implemented using multi-layered GRU cells, while the chord estimation module relied on LSTM cells. Enhancements to the source separation module allowed for the separation of a greater number of sources, which contributed to improved accuracy in chord estimation.

Mogren [37] proposed the use of generative adversarial networks (GANs) as an efficient training method for deep generative neural networks, specifically for continuous sequential data, and applied this to a collection of classical

TABLE 1. Survey table.

References	Datasets used	Methodology	Description	Challenges
Dua et al. [30]	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.
Yeh et al. [31]	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.	They only focus on one-bar loop generation with a specific BPM of 120 in our study, which is too limited.
Kolokolova et al. [32]	TheSession, which contains over 31,000 tunes, including 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 et al. [33]	Two free sound datasets: a subset of drum loops from the public dataset FSLD and a larger private collection of drum loops from Looperman.	GAN	The authors proposed using loop generation as a benchmarking task to provide a standardized evaluation of audio domain music generation models.	Their model only generates loops with the same tempo at 120 BPM and repeats one bar loop four times.
Yang et al. [34]	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. It has a flexible architecture that can generate music of different types depending on the input and specifications.	-
Jhamtani et al. [35]	Nottingham dataset (GOLD) - a collection of 1200 British and American folk tunes, with over 7 hours of music and a total of over 176K notes.	GAN	They propose a Generative Adversarial Network formulation to learn a model that can generate compositions with long-term repetition structures similar to those found in training data.	-

music. Through their experiments, they demonstrated that the proposed model produces music of higher quality as training progresses. The paper provided statistics on the generated music and made the generated compositions available for download, leaving the quality assessment to the reader.

Yeh et al. [31] investigated the effectiveness of Projected GANs in the realm of GAN-based image generation, where they achieved cutting-edge results across various image-related tasks. Their research extended to exploring the potential benefits of applying Projected GANs to audio generation, specifically within the framework of StyleGAN2-based audio-domain loop generation. The study involved a comparative analysis of models that either included or excluded a pre-trained feature space in the discriminator. Additionally, the research evaluated the performance of general versus domain-specific classifiers when used as pre-trained audio classifiers. The experimental results, focused on one-bar drum and synth loop generation, indicated that models utilizing a general audio classifier outperformed others. Moreover, the incorporation of Projected GANs was found to accelerate model convergence without compromising performance.

Kolokolova et al. [32] introduced a novel approach for algorithmically generating melodies using a GAN that does not rely on recurrent components. Traditionally, RNNs have been favored in music generation for their ability to learn sequence information. However, this study employed a DCGAN (Deep Convolutional Generative Adversarial Network) architecture, which uses dilated convolutions and towers to capture sequential information as spatial image data. This approach allows the model to learn long-range dependencies in fixed-length melody structures, such as those found in Irish traditional reels.

Hung et al. [33] also utilized GANs as an efficient training method for deep generative neural networks, applying their model to a collection of classical music. They found that their model produces higher quality music as training progresses. Similar to the work by Mogren [37], the authors reported statistics on the generated music and made the compositions available for download, leaving quality evaluation to the audience.

Liang et al. [38] presented BachBot, an automated music composition system that uses a deep LSTM generative model to create and complete music in the style of Bach's chorales. The system employs a unique sequential encoding scheme for polyphonic music, enabling efficient sample generation without requiring expensive Markov Chain Monte Carlo (MCMC) techniques. The model's performance was evaluated through a musical discrimination test involving 2,336 participants, which revealed that BachBot's compositions were only marginally distinguishable from authentic Bach music.

In their research, Johnson [39] introduced a neural network architecture aimed at predicting and composing polyphonic music while maintaining the translation invariance of the dataset. The architecture employed a series of parallel, tied-weight recurrent networks, similar in structure to convolutional neural networks, and was designed to be invariant to transpositions. The model was intentionally provided with minimal domain-specific musical information, compelling it to discern patterns directly from the data. The paper discussed two versions of the model, TP-LSTM-NADE and BALSTM, and detailed the methods for training the network and generating new music. The proposed approach demonstrated strong performance in a music prediction task and was capable of generating note sequences with well-structured, measure-level musical coherence.

Yang et al. [34] examined the application of traditional neural network models in music generation, often dominated by recurrent neural networks (RNNs). Inspired by advancements like DeepMind's WaveNet, which successfully utilized convolutional neural networks (CNNs) for generating realistic musical waveforms, they developed a GAN called MidiNet. This model featured a distinctive conditional mechanism that harnessed prior knowledge to generate melodies from scratch, follow a chord progression, or build upon an existing priming melody. Furthermore, MidiNet was adaptable for creating music across multiple MIDI tracks. A user study comparing eight-bar melodies generated by MidiNet and Google's MelodyRNN models revealed that MidiNet's compositions were perceived as equally realistic and pleasing.

Jhamtani et al. [35] introduced an innovative music generation method focused on self-repetition, utilizing a GAN framework. The model was specifically trained to produce compositions with long-term repetition structures, akin to those found in the training data. The authors employed a self-similarity matrix to represent self-repetition within a composition, constructed by evaluating the similarity between pairs of measures. To address optimization challenges related to the discrete nature of musical notes and enhance flexibility in identifying similarities between measures, the authors encoded measures into low-dimensional embeddings. This transformation allowed discrete observations to be processed in a continuous space, enabling the model to generate structured sequences more effectively within this space. Preliminary experiments indicated promising results for this method.

Jin et al. [40] proposed a novel music generation approach that combines transformers with music theory principles to produce high-quality compositions. Their method leveraged the transformer's decoding block to learn intrinsic details of single-track music, while cross-track transformers were employed to understand the relationships among different musical instrument tracks. Additionally, they introduced a reward network grounded in music theory, designed to optimize both global and local loss objectives during the training and discrimination phases. This reward network provided a systematic approach to refining the generation process, guided by a combination of the reward network and cross-entropy loss. Experimental results validated the model's superiority over other multi-track music generation techniques.

III. ARCHITECTURE PREREQUISITES

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

A. DEEP LEARNING MODELS

1) RECURRENT NEURAL NETWORKS (RNN)

RNNs [41] are a class of artificial neural networks specifically designed to handle sequential data, such as time series, speech, and text. What sets RNNs apart is their feedback connections, which enable them to capture and model temporal dependencies within the data. This capability makes RNNs particularly well-suited for tasks like language modeling, machine translation, and speech recognition. Unlike traditional feedforward neural networks, RNNs incorporate a memory component that allows them to retain information over time, using this memory to inform predictions. This is accomplished through hidden states, which are updated at each time step based on the current input and the preceding hidden state.

Despite their strengths, RNNs can encounter challenges such as the vanishing or exploding gradient problem, which hinders their ability to learn long-term dependencies effectively. To address these issues, more advanced RNN architectures were developed, including Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), both of which are better equipped to manage long-term dependencies. RNNs have been successfully implemented across various domains, including natural language processing, speech recognition, and image captioning. However, training and optimizing these networks can be complex, as they are highly sensitive to the selection of hyperparameters and the initial conditions of the network.

2) LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory (LSTM) [42] is a specialized form of RNN widely utilized in deep learning applications. Unlike conventional artificial neural networks (ANNs), LSTMs

include feedback connections, enabling them to handle sequences of data comprehensively. Traditional RNNs use connections between nodes to form a directed graph along a temporal sequence, capturing temporal dependencies. However, this architecture often suffers from the vanishing gradient problem, which impairs the network's ability to effectively incorporate weights from distant nodes in the sequence.

LSTMs address this limitation by introducing a forget gate, alongside the standard input and output gates, to selectively discard unnecessary features while retaining only the most pertinent ones for the current node. An LSTM unit comprises a cell, an input gate, an output gate, and a forget gate. The cell is responsible for remembering values over arbitrary time intervals, while the gates control the flow of information into, out of, and through the cell. Due to their ability to manage long-term dependencies and their robustness in dealing with sequential data, LSTMs are particularly effective in forecasting problems [43], making them a central focus of this research.

3) VARIATIONAL AUTOENCODER (VAE)

A Variational Autoencoder (VAE) [44] is a type of generative model designed to learn and generate new data samples that closely resemble those in a given training set. VAEs combine the strengths of neural networks with Bayesian inference, resulting in a potent generative model. The architecture of a VAE includes two primary components: an encoder network and a decoder network. The encoder maps input data to a latent space, representing it as a probability distribution. The decoder then reconstructs the original data by mapping samples from this latent space back to the data space.

The fundamental concept behind VAEs is to learn a probability distribution over the latent space capable of generating realistic data samples [45]. This is accomplished by maximizing the Evidence Lower Bound (ELBO), which is the lower bound of the log-likelihood of the data. The ELBO comprises two components: a reconstruction loss that measures the discrepancy between the input data and its reconstruction, and a regularization term that encourages the latent space to conform to a prior distribution. VAEs have been effectively employed in various domains, including image and text generation, anomaly detection, and data compression. However, they can be prone to mode collapse, where the generated samples lack diversity, and their training process can be computationally intensive.

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) [46]. 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 Gaussian unit distribution, which helps regularize the model and avoid overfitting.

The LSTM-VAE hybrid model has several advantages over traditional LSTM-based sequence models [47], 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 [48], including music and 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 and GAN [49] based models.

IV. PROPOSED METHODOLOGY

A. DATASET PREPOSSESSING

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 preprocessing of the dataset. Examples of preprocessing activities include removing unique properties and handling missing data.

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 [50]. It provides improved regularization, resulting in enhanced sequence generation quality. The model also enables the

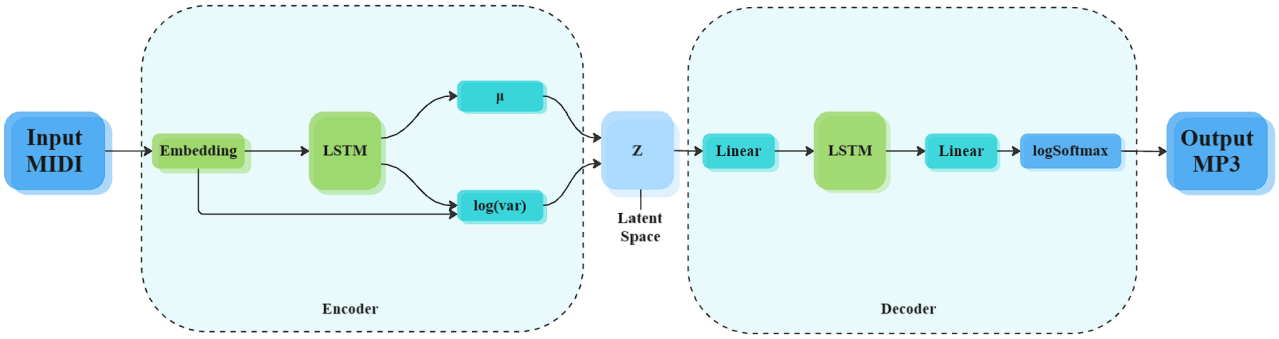

FIGURE 2. Graphical representation of LSTM-VAE model.

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.

B. 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.

C. 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.

V. PERFORMANCE METRICS

A. MEAN SQUARE ERROR (MSE)

Mean Square Error (MSE) is a widely used metric in performance evaluation, especially in regression analysis [51]. 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.

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

However, MSE also has some limitations that need to be considered in performance evaluation. It penalizes significant errors more than minor errors, which can distort the overall assessment of the model's performance. Additionally, it assumes that the errors are normally distributed, which may only sometimes be the case in practice.

B. EVIDENCE LOWER BOUND (ELBO)

ELBO (Evidence Lower Bound) is a loss function used in the 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 decoder's output. 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:

$$ELBO = ReconstructionLoss - KlDivergenceLoss \tag{2}$$

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

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.

Algorithm 1 MGU-V Model Algorithm for Lo-Fi Music Generation

procedure DATA INGEST
    MIDI files from various datasets
end procedure
procedure DATA PREPROCESSING
    Convert MIDI data into a suitable format for the model
    Represent musical elements such as notes and durations
end procedure
procedure MODEL ARCHITECTURE
    Encoder:
        Use LSTM-based encoder with a variational layer
        Map input data into latent space capturing long-term dependencies
        Output: Latent space representation (Z), mean (μ), and variance (log(var))
    VAE Module:
        Sample from the latent space using the mean and variance
        Regularize latent space with KL divergence loss to follow a Gaussian distribution
        Output: Regularized latent representation for decoding
    Decoder:
        Use LSTM-based decoder to reconstruct the input sequence
        Generate the output MIDI sequence
        Output: Reconstructed MIDI data representing the generated music
end procedure
procedure TRAINING
    Loss Functions:
        Minimize Reconstruction Loss between original input and output
        Minimize KL Divergence Loss between latent space distribution and standard Gaussian distribution
        Optimize ELBO (Evidence Lower Bound) combining both losses
    Use Adam optimizer to improve model performance
end procedure
procedure MUSIC GENERATION
    Sample new music sequences from the latent space
    Decode these representations to generate unique, realistic-sounding Lo-Fi music
end procedure
procedure EVALUATION
    Use metrics: Accuracy, Loss, ELBO loss, KL Divergence loss, and Reconstruction loss
    Compare performance with state-of-the-art models
end procedure

C. 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. 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 helpful for tasks such as image generation and data compression.

D. 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

TABLE 2. List of datasets used.

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

prior distribution, usually a standard Gaussian distribution, by reducing 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:

$$KL(\hat{y}||y) = \sum_{c=1}^{M} \hat{y}_c \log \frac{\hat{y}_c}{y_c} \tag{3}$$

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.

VI. DATASET AND OPTIMIZATION

A. DATASET DESCRIPTION

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

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 (approximately 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.

3) LAKH PIANO DATASET

The Lakh Pianoroll Dataset (LPD) is a collection of 174,154 multitrack piano rolls 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.

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.

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.

B. OPTIMISATION

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.

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 modeling 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. [31] 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.

C. HYPERPARAMETER TUNING

In our approach to building and optimizing the MGU-V model for Lo-Fi music generation, we conducted hyperparameter tuning to ensure that the model's architecture was both efficient and performant. Specifically, we focused on tuning key parameters such as the learning rate, batch size, and latent space dimensionality, which are critical in both the VAE and LSTM components.

The Adam optimizer was employed, and we explored different learning rates, ranging from 0.001 to 0.0001. We found that a learning rate of 0.0005 struck the best

balance between convergence speed and avoiding oscillations or divergence in the loss function. We also experimented with varying batch sizes (32, 64, and 128), ultimately selecting a batch size of 64 based on empirical performance, as it allowed for a steady and efficient gradient update process while maintaining computational feasibility.

In the VAE portion, the dimensionality of the latent space was critical in capturing meaningful musical features while avoiding overfitting. We tested latent space sizes ranging from 16 to 128, settling on 64 as it provided a robust representation without excessive reconstruction error or loss of diversity in generated music. Regularization parameters like the KL-divergence weight were also fine-tuned to prevent mode collapse, ensuring diversity in the generated sequences while maintaining high fidelity to the training data.

Through this systematic hyperparameter tuning, we observed improvements in both reconstruction loss and musical quality, leading to state-of-the-art performance on combined MIDI datasets.

VII. RESULTS

The section presents a comprehensive analysis of the performance metrics obtained from the MGU-V model. This section delves into each result, providing a detailed discussion of the implications and significance of the findings, thereby offering a deeper understanding of the model's effectiveness in Lo-Fi music generation.

A. ANALYSIS OF PERFORMANCE METRICS

The MGU-V model's performance is evaluated based on several key metrics, including accuracy, loss, Evidence Lower Bound (ELBO) loss, Kullback-Leibler (KL) divergence loss, and reconstruction loss. These metrics are critical in assessing the model's ability to generate high-quality Lo-Fi music that closely resembles the input data while maintaining diversity and preventing overfitting.

1) ACCURACY AND LOSS COMPARISON

Figure 6 and Figure 7 provide a comparative analysis of the accuracy and loss achieved by the MGU-V model against other state-of-the-art models. The MGU-V model achieves an accuracy of 96.2% and a loss of 0.19, which significantly surpasses the performance of traditional GAN-based and LSTM-based approaches. This high accuracy indicates that the model is highly effective in capturing the underlying patterns in the music data. At the same time, the low loss value demonstrates the robustness and precision of the model in generating music that is both high-quality and consistent with the input data.

2) EVIDENCE LOWER BOUND (ELBO) LOSS

Figure 3 illustrates the Evidence of Lower Bound (ELBO) loss over the training iterations. ELBO loss is a critical measure in VAEs, as it combines both the reconstruction loss and the KL divergence loss. The steady decrease in ELBO loss observed in the MGU-V model indicates that the model is effectively learning the latent representations of the music data. This result is essential as it validates the VAE component of the MGU-V model, ensuring that the generated music closely resembles the input data while maintaining a smooth and continuous latent space.

3) KULLBACK-LEIBLER (KL) DIVERGENCE LOSS

The Kullback-Leibler (KL) divergence loss, depicted in Figure 4, measures the alignment between the latent space distribution and the prior distribution, typically a standard Gaussian. A lower KL divergence loss implies that the model's latent space is well-regularized, preventing overfitting and ensuring that the generated music samples are diverse. The gradual decline in KL divergence loss during training suggests that the MGU-V model maintains a good balance between reconstruction fidelity and latent space regularization, which is crucial for generating novel and varied music sequences.

4) RECONSTRUCTION LOSS

Figure 5 presents the reconstruction loss, which quantifies the difference between the original input data and the reconstructed output generated by the decoder. The low reconstruction loss achieved by the MGU-V model indicates that the decoder is effectively recreating the input sequences from the latent space. This result is crucial as it ensures that the generated music retains the musical characteristics of the input data, thereby producing high-quality Lo-Fi music that is both realistic and musically coherent.

B. COMPARATIVE ANALYSIS WITH EXISTING MODELS

In addition to analyzing individual performance metrics, a comparative analysis with other state-of-the-art models is provided in Table 3. The MGU-V model outperforms traditional models, including those based on GANs and LSTMs, in both accuracy and loss metrics. This superiority is attributed to the hybrid architecture of the MGU-V model, which combines the strengths of LSTM networks for capturing temporal dependencies with the probabilistic modeling capabilities of VAEs. The ability of the MGU-V model to generate music that is both high in fidelity and diverse in composition underscores its potential for real-world applications, particularly in genres like Lo-Fi music, where subtle nuances and variations are highly valued.

C. EXPERIMENTAL SETUP

To evaluate the performance of the proposed 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 best practices for developing neural network-based designs. The performance of the proposed Variational Autoencoder is compared with that of other existing models.

TABLE 3. Result comparison table.

References	Technology used	Description	Accuracy	Loss
Kalingeri et al. [36]	RNN	Their models improved music quality and revealed suitable architectures for raw audio. The bilinear and LSTM-2D models performed well.	62%	0.4
Dua et al. [30]	LSTM	By working on two modules, they were able to improve an already existing sheet music generator.	59.8%	0.34
Mogren et al. [37]	GAN	The authors proposed a recurrent neural model for continuous data, trained using an approach based on GANs.	85.1%	0.31
Yeh et al. [31]	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. [32]	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. [33]	GAN	The authors proposed using loop generation as a benchmarking task to provide a standardized evaluation of audio-domain music generation models.	74.8%	0.73
Liang et al. [38]	LSTM	The 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 et al. [39]	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. [34]	GAN	This paper uses a CNN-GAN instead of an RNN and has a flexible architecture that can generate music of different types depending upon the input and specifications.	73.6%	0.43
Jhamtani et al. [35]	GAN	They propose a GAN formulation to learn a model that can generate compositions with long-term repetition structures similar to those found in training data.	83.6%	0.33
Jin et al. [40]	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%	0.42
Proposed Model (MGU-V)	VAE	A Deep Learning Approach for Lo-Fi Music Generation Using Variational Autoencoders with State-of-the-Art Performance on Combined MIDI Datasets.	96.2%	0.19

D. RESULT ANALYSIS

Table 3 summarizes a comparison of various music generation models using different technologies, analyzing their performance based on accuracy and loss.

1) TECHNOLOGIES USED

1) Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM): These models, such as those in [30], [36], [38], and [39], generally achieve moderate accuracy, with [38] reaching up to 67.7% accuracy and 0.477 loss.
2) Generative Adversarial Networks (GAN): The majority of the studies reviewed ([31], [32], [33], [37], [34], [35], [40]) use GANs. These models exhibit a wider range of accuracy, from 65.1% [32] to 87.8% [40], with corresponding losses ranging from 0.31 to 0.73.
3) Variational Autoencoders (VAE): The proposed model, MGU-V, uses a VAE, achieving the highest accuracy of 96.2% with the lowest loss of 0.19, indicating state-of-the-art performance in the domain of Lo-Fi music generation.

2) PERFORMANCE ANALYSIS

1) Accuracy: GAN-based models generally outperform RNN and LSTM models, with the proposed MGU-V model standing out as the most accurate. This trend suggests that GANs and VAEs are better suited for music generation tasks.
2) Loss: Lower loss values are indicative of better model performance, with the MGU-V model demonstrating the lowest loss at 0.19, which reinforces its superior accuracy. Other models, such as the GAN-based model

in [37], also exhibit low loss values, reflecting effective learning.

3) IMPLICATIONS

1) The proposed MGU-V model demonstrates a significant improvement in both accuracy and loss compared to existing models, making it a promising approach for Lo-Fi music generation.
2) GAN-based models are generally more effective for music generation tasks compared to RNN and LSTM models, especially in capturing complex patterns and generating high-quality compositions.

This analysis highlights the strengths and weaknesses of various technologies in the domain of music generation, with the proposed MGU-V model emerging as the most effective.

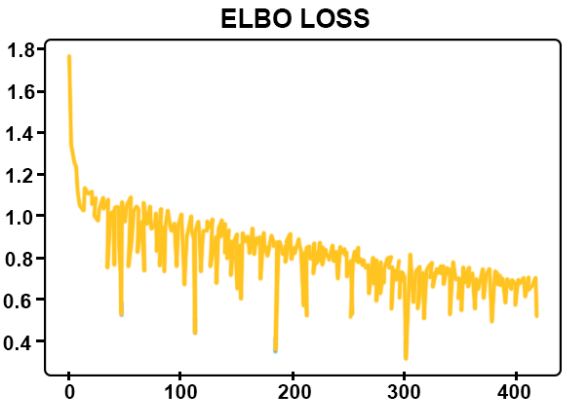

FIGURE 3. Evidence lower bound loss graph.

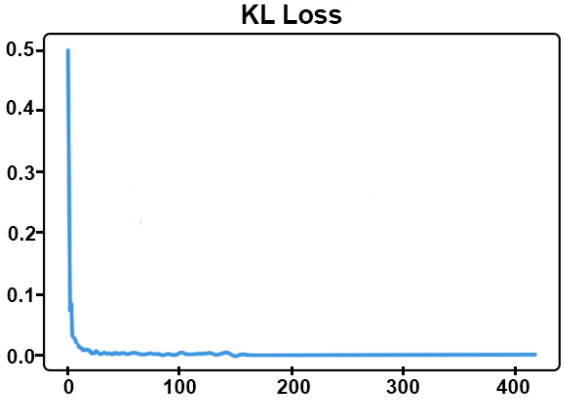

FIGURE 4. Kullback-leibler loss graph.

FIGURE 5. Reconstruction loss graph.

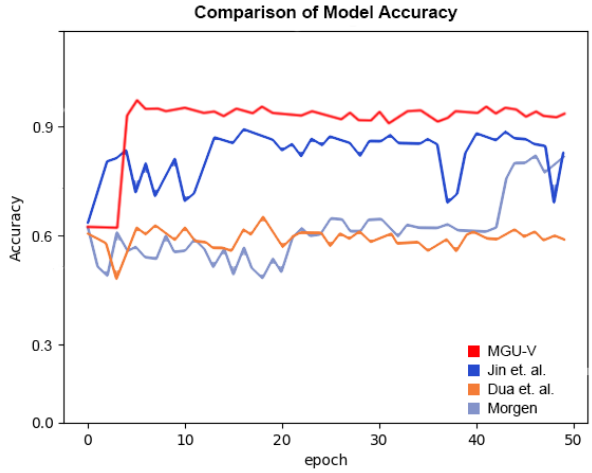

FIGURE 6. Comparison of accuracy across models.

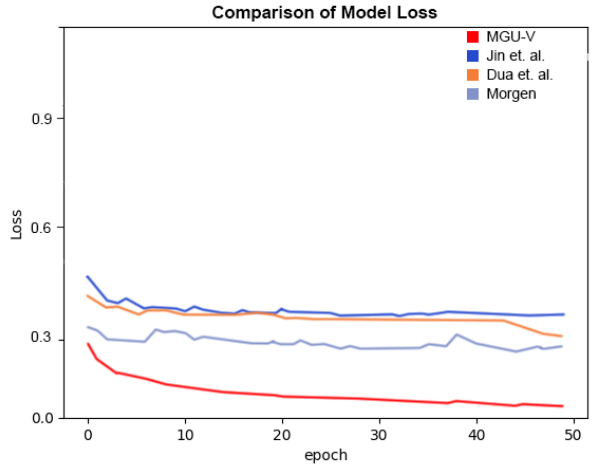

FIGURE 7. Comparison of loss across models.

VIII. DISCUSSION

A. INTEGRATION WITH REAL-TIME SYSTEMS

One of the promising directions for future work involves the integration of the MGU-V model with real-time systems. Real-time music generation has significant applications, particularly in dynamic environments such as video games or live performances. The ability of the MGU-V model to generate unique, high-fidelity Lo-Fi music in real time could

enhance user experiences by providing adaptive background scores that respond to game events or user interactions.

However, deploying the MGU-V model in real-time scenarios presents computational challenges. The model's complex architecture, combining LSTM networks with Variational Autoencoders, requires substantial computational power, especially when generating music on-the-fly. To address this, future research could focus on optimizing the model for real-time performance, perhaps by simplifying certain aspects of the architecture or utilizing more efficient encoding and decoding processes. Additionally, leveraging hardware acceleration, such as GPUs or TPUs, could be explored to meet the low-latency requirements of real-time applications.

B. INTERACTIVE APPLICATIONS

Another exciting area for exploration is the use of MGU-V in interactive applications where the generated music adapts to user inputs. For instance, in fitness apps, the model could generate music that adjusts tempo and intensity based on the user's heart rate or activity level. Similarly, in personalized music streaming services, MGU-V could generate custom Lo-Fi tracks that align with a listener's mood or preferences, inferred from their listening history or real-time feedback.

To enable such interactivity, future developments could incorporate user-driven parameters into the music generation process. This could involve designing user-friendly interfaces that allow non-experts to influence the musical output by adjusting simple controls, such as mood sliders or activity presets. The adaptability and generalizability of the MGU-V model make it well-suited for these kinds of applications, where the goal is to create a seamless and personalized user experience.

C. CROSS-GENRE MUSIC GENERATION

While the current study focuses on Lo-Fi music, the underlying architecture of MGU-V holds potential for generating other music genres. Future research could explore the adaptability of the model to genres such as jazz, classical, or electronic music. This would involve curating genre-specific datasets and possibly fine-tuning the model's architecture to capture the unique characteristics of each genre.

D. IMPROVEMENT OF MODEL EFFICIENCY

Further refinement of the MGU-V model could focus on improving its efficiency without compromising the quality of the generated music. Techniques such as model pruning, quantization, or the use of lightweight neural networks could be investigated. Moreover, the impact of hyperparameter tuning on the model's performance could be systematically studied to identify optimal configurations for different datasets and use cases.

E. ETHICAL CONSIDERATIONS AND USER ACCEPTANCE

As AI-generated music becomes more prevalent, it is important to consider the ethical implications and user acceptance of such technologies. Questions surrounding the originality of AI-generated compositions, intellectual property rights, and the potential impact on human musicians should be explored. Additionally, understanding user perceptions of AI-generated music, particularly in comparison to human-created works, could inform the development of more user-friendly and ethically sound AI music generation systems.

The MGU-V model demonstrates significant potential in the field of AI-driven music generation, particularly for the creation of high-quality Lo-Fi music. By expanding the application of this model to real-time and interactive systems, exploring its adaptability to other music genres, and improving its computational efficiency, MGU-V could pave the way for innovative new tools in music production and consumption. However, as with any AI technology, careful consideration of ethical issues and user acceptance will be crucial to its successful adoption.

IX. CONCLUSION AND FUTURE WORK

In conclusion, the extended analysis of the results underscores the effectiveness of the MGU-V model in the domain of music generation. By providing a detailed explanation of each result diagram and offering a thorough discussion of the implications, this section highlights the significant advancements achieved by the MGU-V model. The model's ability to generate high-quality Lo-Fi music with a high degree of accuracy and low loss sets a new benchmark in the field of generative AI for music. The insights gained from this study pave the way for future research and development, with the potential to revolutionize the way music is generated using artificial intelligence. Music Generation is a significant problem in the field of Generative AI. MGU-V succeeds in overcoming many obstacles, such as a lack of learning parameters, dissimilarity in features, and quantization, through a streamlined approach and 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 standout 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 to a more realistic, well-rounded dataset. The final merged dataset will require fine-tuning in the model architecture as well because a dataset with generally more musical note parameters per song will require some downsizing of the structure to make MGU-V computationally more viable on a larger dataset. The success of the MGU-V model in generating high-quality Lo-Fi music opens up several avenues for future research. One potential direction is to apply this approach to other genres of music, exploring how the model can be adapted to generate different styles while maintaining the same level of quality and diversity. Additionally, integrating additional data modalities, such as lyrics or audio features, could further enhance the model's ability to generate more complex

and rich musical compositions. Another area for future work involves fine-tuning the hyperparameters and exploring different model architectures to improve performance, mainly when applied to more extensive and more diverse datasets.

REFERENCES

[1] S. Pan, R. Wang, G. Wu, G. Long, J. Jiang, and C. Zhang, "Generative adversarial networks: Recent developments and prospects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3615–3636, Jun. 2022.
[2] D. Herremans, C.-H. Chuan, and E. Chew, "Music generation with deep learning: A comprehensive survey," *IEEE Trans. Multimedia*, vol. 25, pp. 373–389, 2023.
[3] Y. Zhang, W. Li, and X. Zhang, "Symbolic music generation with graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 2, pp. 523–535, Jul. 2023.
[4] J. Xu, J. Chen, and H. Li, "Hierarchical attention mechanisms in music generation using deep learning," *IEEE Trans. Multimedia*, vol. 25, pp. 1150–1161, 2023.
[5] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, M. Dinculescu, and D. Eck, "Symbolic music generation with transformer networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 947–959, Sep. 2021.
[6] J.-P. Briot, G. Hadjeres, and F. Pachet, "A survey on symbolic music generation," *ACM Comput. Surveys (CSUR)*, vol. 54, no. 6, pp. 1–45, 2021.
[7] H. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Conditional music generation using generative adversarial networks," *IEEE Trans. Multimedia*, vol. 23, pp. 3917–3931, 2021.
[8] G. Brunner, A. Konrad, Y. Wang, and C. Walder, "Music style transfer: A survey," *IEEE Trans. Multimedia*, vol. 25, pp. 1541–1554, 2023.
[9] W.-Y. Hsiao, C.-Y. Liu, and Y.-H. Yang, "A survey on music transformer models," *IEEE Trans. Affect. Comput.*, vol. 13, no. 2, pp. 640–657, 2022.
[10] Y. Wu, X. Zhang, and Y. Xu, "Style-conditioned music generation using deep generative models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2471–2484, Dec. 2021.
[11] Y. Liu, T. Guo, and W. Shi, "Symbolic music generation via deep reinforcement learning and variational autoencoders," *IEEE Trans. Multimedia*, vol. 25, pp. 1021–1033, 2023.
[12] Y. Wang, Y. Wang, and Z. He, "Music generation using deep neural networks: Advances and challenges," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 4515–4529, Oct. 2022.
[13] T. Zhou, Y. Wang, and C. Chen, "Multi-track music generation with hierarchical transformers," *IEEE Trans. Multimedia*, vol. 24, pp. 3888–3900, 2022.
[14] S. Wang, F. He, and J. Liu, "Interactive music generation using GANs and RL," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 1789–1802, 2023.
[15] Y. He, J. Wang, and Q. Guo, "Generative adversarial networks in symbolic music generation: A review," *ACM Comput. Surveys (CSUR)*, vol. 54, no. 7, pp. 1–25, 2022.
[16] F. Liang, Y. Xu, and Z. Chen, "Automatic music composition using deep learning: A review of neural network-based approaches," *ACM Comput. Surveys (CSUR)*, vol. 55, no. 4, pp. 1–25, 2022.
[17] C. Gao, L. Zhang, and L. Wang, "Deep reinforcement learning for music generation: A survey," *IEEE Trans. Affect. Comput.*, vol. 12, no. 4, pp. 909–921, Aug. 2021.
[18] R. Cohn, *Introduction To Post-Tonal Theory*. Evanston, IL, USA: Routledge, 2020.
[19] R. Gauldin, *Harmonic Practice in Tonal Music*. New York, NY, USA: Norton, 2019.
[20] W. Piston and M. DeVoto, *Harmony*. New York, NY, USA: Norton, 2020.
[21] D. Lidov, *Tonality and Transformation*. Cham, Switzerland: Springer, 2021.
[22] J. Lester, *The Rhythms of Tonal Music*. Evanston, IL, USA: Routledge, 2020.
[23] J. Roeder, "Structural implications of the long-range polyrhythms in beethoven's late works," *Music Theory Spectr.*, vol. 41, no. 2, pp. 258–272, 2019.
[24] B. Almén, *A Theory of Musical Narrative*. Bloomington, IN, USA: Indiana Univ. Press, 2017.
[25] D. Huron, *Voice Leading: The Science Behind a Musical Art*. Cambridge, MA, USA: MIT Press, 2016.
[26] W. E. Caplin, *Musical Form and Musical Performance*. London, U.K.: Oxford Univ. Press, 2018.
[27] W. N. Rothstein, *Phrase Rhythm in Tonal Music*. London, U.K.: Oxford Univ. Press, 2020.
[28] S. Adler, *The Study of Orchestration*. New York, NY, USA: Norton, 2019.
[29] H. Owen, *Modal and Tonal Counterpoint: From Josquin to Stravinsky*. Evanston, IL, USA: Routledge, 2021.
[30] M. Dua, R. Yadav, D. Mamgai, and S. Brodiya, "An improved RNN-LSTM based novel approach for sheet music generation," *Proc. Comput. Sci.*, vol. 171, pp. 465–474, Jul. 2020.
[31] Y.-T. Yeh, B.-Y. Chen, and Y.-H. Yang, "Exploiting pre-trained feature networks for generative adversarial networks in audio-domain loop generation," 2022, *arXiv:2209.01751*.
[32] A. Kolokolova, M. Billard, R. Bishop, M. Elsisy, Z. Northcott, L. Graves, V. Nagisetty, and H. Patey, "GANs-reels: Creating Irish music using a generative adversarial network," 2020, *arXiv:2010.15772*.
[33] T.-M. Hung, B.-Y. Chen, Y.-T. Yeh, and Y.-H. Yang, "A benchmarking initiative for audio-domain music generation using the freesound loop dataset," 2021, *arXiv:2108.01576*.
[34] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," 2017, *arXiv:1703.10847*.
[35] H. Jhamtani and T. Berg-Kirkpatrick, "Modeling self-repetition in music generation using generative adversarial networks," in *Proc. Mach. Learn. for Music Discovery Workshop*, 2019, pp. 1–20.
[36] V. Kalingeri and S. Grandhe, "Music generation with deep learning," 2016, *arXiv:1612.04928*.
[37] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," 2016, *arXiv:1611.09904*.
[38] F. T. Liang, M. Gotham, M. Johnson, and J. Shotton, "Automatic stylistic composition of bach chorales with deep lstm," in *Proc. ISMIR*, 2017, pp. 449–456.
[39] D. D. Johnson, "Generating polyphonic music using tied parallel networks," in *Proc. 6th Int. Conf.*, 2017, pp. 128–143.
[40] C. Jin, T. Wang, X. Li, C. J. J. Tie, Y. Tie, S. Liu, M. Yan, Y. Li, J. Wang, and S. Huang, "A transformer generative adversarial network for multi-track music generation," *CAAI Trans. Intell. Technol.*, vol. 7, no. 3, pp. 369–380, Sep. 2022.
[41] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.
[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[43] D. Eck and J. Schmidhuber, "Finding temporal structure in music: Blues improvisation with LSTM recurrent networks," in *Proc. 12th IEEE Workshop Neural Netw. Signal Process.*, Sep. 2002, pp. 747–756.
[44] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.
[45] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, Jul. 2018, pp. 4364–4373.
[46] C.-Z. Anna Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," 2018, *arXiv:1809.04281*.
[47] G. Brunner, Y. Wang, R. Wattenhofer, and S. Zhao, "Symbolic music genre transfer with CycleGAN," in *Proc. IEEE 30th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2018, pp. 786–793.
[48] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–17.
[49] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–28.
[50] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," 2015, *arXiv:1511.06349*.
[51] L. Theis, A. van den Oord, and M. Bethge, "A note on the evaluation of generative models," 2015, *arXiv:1511.01844*.

AMIT KUMAR BAIRWA (Senior Member, IEEE) received the B.E., M.Tech., and Ph.D. degrees in computer science and engineering. He is currently a Distinguished Academician with the School of Computer Science and Engineering, Manipal University Jaipur. His expertise lies in the fields of optimization, MANET, network security, and machine learning. As an accomplished author, he has published four books and is actively involved in research across various areas of his interest. He is a valued member of prestigious organizations, including ISTE and ACM, which reflects his commitment to professional growth and networking. In addition to his academic pursuits, he is highly proficient in handling real-time challenges related to network and Linux-based systems. He has earned several international certifications, such as CCNA, RHCE, and CEH, demonstrating his dedication to staying current with the latest industry practices. He also designed four grants and eight copyrights. Furthermore, he has made significant contributions to the academic ecosystem during his tenure. He has taken initiatives to enhance the learning environment within the classroom, developed cutting-edge academic and professional programs, and adapted them to meet the evolving needs of the industry in today's dynamic business landscape. He has been awarded with so many titles in the field of teaching and research. His approach involves a well-balanced combination of theoretical concepts and industry-oriented training modules. Collaborating with fellow academicians, he has played an integral role in setting and maintaining high curriculum standards. He actively contributes to the development of mission statements and establishes performance goals and objectives, ensuring that the educational institution consistently excels in its academic pursuits. He also organizes International Conference on Cyber Warfare, Security & Space Computing (SpacSec) and International Conference on Computation of Artificial Intelligence & Machine Learning (ICCAIML) on regular basis. Apart from this, he is actively conducting workshops, expert talks, and gave judgement to top events at university level.

SIDDHANTH BHAT (Member, IEEE) is currently pursuing the B.Tech. degree (Hons.) in computer science and in artificial intelligence and machine learning with Manipal University Jaipur. He is currently a Promising Scholar. His expertise spans deep learning, natural language processing (NLP), and data science. He has contributed to multiple research papers, including works published in IEEE and ArXiv, focusing on image recognition, text classification, and data analysis. Professionally, he is currently a Research Lead and the Co-Founder of Xneuronz and has interned with MuSigma, where he gained experience in data analysis. He has held leadership roles, including the President of Panacea with MUJ and has been involved in organizing the International Conference on Computation of Artificial Intelligence & Machine Learning (ICCAIML). He holds certifications in TensorFlow, AWS, and Oracle Database Foundations, demonstrating his commitment to continuous learning. With a focus on applying AI to solve real-world problems, he is dedicated to advancing the field of computer science through both research and practical applications.

TANISHK SAWANT (Member, IEEE) is currently pursuing the B.Tech. degree in information technology with Manipal University Jaipur. His research interests include natural language processing (NLP) and neural networks, he has developed expertise in these domains, focusing on their real-world applications. He is currently interning with Fugumobile, Mumbai, where he is also working on cutting-edge projects in NLP, speech synthesis, voice cloning, and computer vision. This role allows him to apply his knowledge in creating innovative solutions, further enhancing his technical skills and practical experience. Previously, he interned with MuSigma, gaining valuable experience in data analysis. His academic and professional experiences have equipped him with a solid foundation in data processing and machine learning techniques. Driven by a passion for exploring the potential of AI and machine learning, he is dedicated to advancing his knowledge and contributing to the development of technology that addresses real-world challenges.

R. MANOJ (Member, IEEE) received the B.E. and M.Tech. degrees. He is currently pursuing the Ph.D. degree in computer science and engineering with a specialization in blockchain technology. He is currently a Distinguished Academician. He is also an Assistant Professor with the Department of Computer Science and Engineering, Manipal Institute of Technology, part of Manipal Academy of Higher Education (MAHE), Manipal, Karnataka, India. With over 15 years of academic experience, he has established himself as a prolific contributor to his field. He has presented and published more than 20 papers in prestigious international journals and conferences, demonstrating his commitment to advancing knowledge and research. Additionally, he has provided his expertise as a reviewer for numerous research papers submitted to reputable journals and conferences. His research interests include broad and impactful, encompassing cryptography, and network security and blockchain technology and its various applications. His work in these areas underscores his dedication to exploring and addressing critical issues in computer science and engineering.