Jurnal Masyarakat Informatika, 17(2), 2026
DOI:10.14710/jmasif.17.2.78593
ISSN: 2777-0648

Effects of Temperature Sampling and Tempo Variation on LSTM-Based Music Generation

Imam Fathurrahman[*1)], Muhammad Djamaluddin[1)], Ramli Ahmad[2)], Hariman Bahtiar[3)], Muhamad Sadali[1)], Zulkipli[1)]

[1)] *Department of Informatics, Hamzanwadi University, Lombok, Indonesia*

[2)] *Department of Computer Engineering, Hamzanwadi University, Lombok, Indonesia*

[3)] *Department of Information Systems, Hamzanwadi University, Lombok, Indonesia*

[*] *Corresponding author: i.fathurrahman@hamzanwadi.ac.id*

Abstract

Long Short-Term Memory (LSTM) models for symbolic music generation may produce repetitive sequences under deterministic decoding. This study examines how sampling temperature and externally imposed tempo variation affect the statistical characteristics of generated MIDI music. A factorial experiment combined four decoding settings (Argmax and temperature sampling at T = 0.6, 0.7, and 1.0) with fixed- and dynamic-tempo rendering. Using one trained LSTM checkpoint and controlled seeds, 40 MIDI files were generated across eight conditions, with five runs per condition. The dataset was split by MIDI file before sequence construction to reduce data leakage. Generated outputs were assessed using pitch entropy, repetition ratios, N-gram overlap, pitch-class distribution similarity, rhythmic diversity, and tempo statistics. At T = 1.0, mean pitch entropy was highest (5.57 ± 0.13 bits), immediate repetition lowest (0.10 ± 0.01), and Chroma Jensen-Shannon Divergence lowest (0.0244 ± 0.0163). Higher temperatures also reduced trigram overlap with the validation corpus, indicating a diversity-similarity trade-off. Dynamic-tempo rendering introduced timing variation without changing pitch-based metrics under matched sampling conditions. These results distinguish effects of token sampling from tempo rendering but do not establish perceptual musical quality.

Keywords: *Neural music generation, Long Short-Term Memory, temperature sampling, dynamic tempo, MIDI, musical diversity*

1 Introduction

The advancement of Artificial Intelligence (AI) has had a significant impact on the domains of art and creativity, including music. One of the most prominent approaches is *neural music generation*, which leverages artificial neural networks to learn melodic, harmonic, and rhythmic patterns from musical datasets and subsequently generate new compositions in a generative manner. The Long Short-Term Memory (LSTM) model, a variant of the Recurrent Neural Network (RNN), is often chosen due to its ability to capture long-term dependencies in sequential data, including musical notation [1 – 6]. LSTM models are capable of producing coherent compositions that resemble the stylistic characteristics of human-generated music [7 – 9]. Architectural innovations such as Bi-LSTM, hybrid GRU-LSTM, and attention integration have further enhanced the model's ability to recognize long-range dependencies and maintain musical consistency [10], [11], [15], [20].

One of the crucial aspects in the generation process is the probabilistic sampling mechanism, where the *temperature* parameter plays a key role in controlling the level of creativity and diversity in the generated output. Higher temperature values tend to produce more exploratory variations, whereas lower temperatures yield more conservative and predictable outputs [14]. Recent studies have

Received: 17 October 2025; Revised: 17 September 2026; Accepted: 11 November 2026; Published: 19 November 2026
Copyright (c) 2026 The authors. Published by Department of Informatics, Universitas Diponegoro
This is an open access article under the CC-BY-SA license.

emphasized that temperature-based sampling can improve melodic novelty and diversity, although a proper balance is required to preserve musical structure [8], [13].

In addition to temperature, tempo variation is an important factor in shaping the temporal characteristics of generated music. Previous studies have investigated the use of recurrent neural networks, including LSTM-based models, to reproduce expressive timing and performance variations [8], [13], [16], [17]. Studies on expressive timing and dynamics, including those examining Chopin's Mazurkas, have further explored the ability of neural models to reproduce characteristics of human musical performances [12]. However, tempo variation can also be introduced externally during MIDI rendering, independently of the model's learned musical token sequence. This distinction is important when evaluating the effects of sampling strategies and tempo-rendering conditions.

A major challenge in AI-based music generation lies in evaluating generated outputs beyond subjective listening assessments. Musical quality is often assessed through listener surveys or Mean Opinion Score (MOS) ratings [7], [11], [16], [17]. Objective quantitative metrics can provide complementary information about the statistical and structural characteristics of generated music. Such metrics include pitch entropy, repetition measures, N-gram overlap, and pitch-class distribution measures, which can characterize melodic diversity, repetitive patterns, local sequence similarity, and tonal distribution, respectively [2], [12], [14], [19], [21], [22]. Nevertheless, these metrics capture specific properties of generated sequences and should not be interpreted as direct or comprehensive measures of perceived musical quality [16], [18].

Previous studies on LSTM-based symbolic music generation have investigated architectural improvements, including Bi-LSTM, attention mechanisms, and hybrid models, as well as sampling and performance-related techniques [1 – 3], [8], [12], [21 – 23], [26]. However, the relationship between sampling variability and temporal variation remains an important consideration in music generation. Temperature sampling modifies the probability distribution used to select musical tokens, whereas tempo variation affects the temporal characteristics of the rendered MIDI output. This study investigates these complementary aspects through an LSTM-based symbolic music generation pipeline that combines temperature-controlled sampling with fixed and dynamic tempo-rendering conditions. The generated outputs are evaluated using complementary quantitative metrics to characterize pitch diversity, repetition, local sequence similarity, pitch-class distribution, and temporal variation. In this experimental setup, dynamic tempo is externally imposed during MIDI rendering rather than learned by the LSTM. The evaluation therefore focuses on measurable characteristics of the generated sequences and rendered outputs.

Building upon these developments, this study investigates the effects of temperature-controlled sampling and tempo-rendering conditions on LSTM-based symbolic music generation. The generated MIDI outputs are evaluated using pitch entropy, immediate repetition ratio, N-gram overlap, Chroma Jensen-Shannon Divergence (JSD), and tempo statistics to characterize melodic diversity, repetition, local sequence similarity, pitch-class distribution, and temporal variation. To the best of our knowledge, this study provides a quantitative examination of temperature-controlled sampling under different tempo-rendering conditions within an LSTM-based music generation framework. Rather than claiming improvements in perceived musical quality, the study focuses on the measurable effects of these generation and rendering settings on the resulting symbolic sequences and MIDI outputs.

2 Literature Review

In recent years, the application of deep learning for automatic music generation has advanced rapidly, with the LSTM architecture being the most widely used due to its ability to capture long-term dependencies in sequential data. Zhang [1] validated the superiority of LSTM over standard RNNs in learning long-range dependencies, making it highly effective for symbolic music generation tasks.

Subsequent studies have reinforced these findings. Conner et al. [3] demonstrated that the LSTM architecture can generate pitch and duration sequences that form complete musical compositions, although it still faces challenges such as repetition and limited training data. Uppuluri [6] emphasized that while LSTM effectively preserves melodic and harmonic coherence, it tends to struggle with maintaining long-term harmonic structures.

Furthermore, several studies have sought to extend the architecture and methodology of music generation. Joshi et al. [2] explored multi-layered LSTM models trained on MIDI data and found that deeper architectures improved the structural consistency of generated music. Gupta et al. [21] compared LSTM with Variational Autoencoders (VAEs) and showed that hybrid architectures enhanced melodic clarity and diversity compared to traditional algorithmic approaches. This is further supported by Agarwal [22], who highlighted the optimization of LSTM and VAE for symbolic, sequence-based music generation tasks.

Nevertheless, many studies still face limitations regarding musical variation. Liu (2025) noted that most AI models overlook aspects of human performance expressiveness, causing the generated music to sound mechanical and emotionally shallow [23]. To address this issue, researchers have begun exploring temperature-based sampling techniques. Fathima et al. (2024) demonstrated that temperature adjustment significantly influences melodic creativity lower-values increase structural regularity, while higher values enhance diversity but reduce coherence [8]. Similar findings were reported by Evangeline et al. (2023), who observed that carefully tuned temperature settings can yield music that remains both coherent and varied [24].

In addition to temperature, tempo has also been shown to play a critical role in enhancing musical expressiveness. Jedrzejewska et al. [12] investigated the use of LSTM-based models for reproducing tempo and expressive dynamics in Chopin's Mazurkas. Their work highlights the importance of temporal and performance-related characteristics in computational music generation. Similarly, Gunawan et al. [25] developed an automatic music generator based on LSTM and GRU models with both objective and subjective evaluations. Although the generated music was perceived as engaging, respondents noted a lack of expressiveness, dynamics, and variation. Consequently, they recommended incorporating tempo and dynamic factors into future models.

From an industry perspective, Zhang et al. [1] emphasized that despite the continuous evolution of AI-based generative music models, their controllability and interactivity remain far from matching the creativity of human composers, particularly within the context of popular music.

Overall, prior research reveals a strong trend toward improving the quality of LSTM-based generative music. However, the combined effects of temperature-controlled sampling and externally imposed tempo variation on the statistical characteristics of LSTM-generated music remain insufficiently examined. To address this gap, the present study investigates the effects of temperature-controlled sampling and externally imposed tempo variation on the statistical and temporal characteristics of LSTM-generated music.

3 Research Methods

Figure 1 illustrates the methodological workflow of this study. The diagram presents the main stages, starting from dataset collection to the evaluation of the generated music. Each stage is systematically interconnected to ensure a coherent process of model training and evaluation within the LSTM framework.

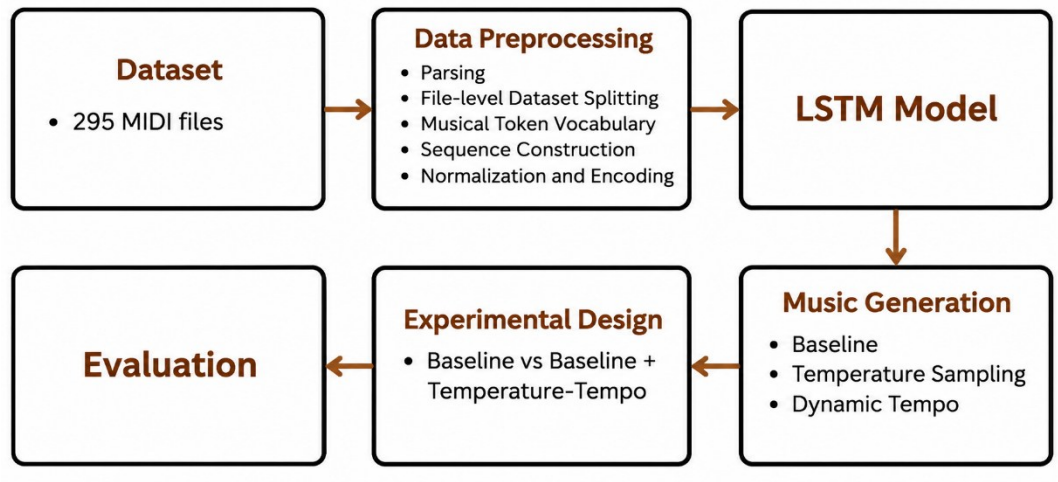


Figure 1 Research Methodology Workflow

3.1 Dataset

The dataset used in this study was obtained from a public source on Kaggle (https://www.kaggle.com/datasets/soumikrakshit/classical-music-midi) in the *.midi/.mid* file format. It contains 295 digital music files represented in the Musical Instrument Digital Interface (MIDI) format. The MIDI format was selected because it allows the extraction of notes, chords, and durations in a standardized structure, facilitating analysis and machine learning processes. Prior to use, each MIDI file was verified for readability using the *Music21* library. Files that were corrupted or did not conform to the standard MIDI format were excluded from the training process.

3.2 Data Preprocessing

The preprocessing stage was designed to transform the MIDI data into numerical sequences suitable for LSTM-based music generation. The process consisted of MIDI parsing, file-level dataset splitting, vocabulary construction, sequence generation, and numerical normalization. To prevent potential data leakage, the dataset was divided at the MIDI-file level before the sliding-window sequences were generated:

3.2.1 Parsing

Each MIDI file was processed using the Music21 library to extract its musical events, including individual notes and chords. Individual notes (note.Note) were represented using their corresponding pitch names, whereas chords (chord.Chord) were represented as symbolic combinations of their constituent pitches. These representations were subsequently converted into discrete musical tokens to facilitate sequential processing by the LSTM model.

3.2.2 File-Level Dataset Splitting

Before constructing the input sequences, the 295 MIDI files were divided into training and validation subsets at the file level using an approximately 80:20 split. The resulting dataset consisted of 235 training files of approximately 80% and 60 validation files of approximately 20%. The split was performed before sequence construction to prevent sequences originating from the same MIDI file from being distributed across both subsets.

The training and validation MIDI files were then parsed independently. Sliding-window sequence construction was performed separately within each partition, ensuring that overlapping sequences derived from the same MIDI file could not cross the training–validation boundary. This file-level separation was intended to reduce data leakage between the training and validation subsets.

3.2.3 Musical Token Vocabulary

The musical token vocabulary was constructed exclusively from the training subset to prevent information from the validation data from influencing vocabulary construction. Each unique musical token observed in the training data was assigned an integer index using a token-to-integer mapping.

A special unknown token (<unk>) was added at index 0 to represent musical tokens encountered in the validation subset but not observed during training. The resulting vocabulary contained 731 classes, comprising 730 training-derived musical tokens and one special unknown token. This mapping was subsequently used to convert both training and validation sequences into integer representations.

3.2.4 Sequence Construction

After dataset splitting and vocabulary construction, the musical tokens were converted into input-target pairs using a sliding-window approach. The sequence length was set to 100 tokens, with a stride of 2. Thus, each input sequence consisted of 100 consecutive musical tokens, while the immediately following token was used as the prediction target.

The sliding windows were generated independently for each MIDI file. No sequence was allowed to cross the boundary between different MIDI files. Tokens not present in the training-derived vocabulary were mapped to the <unk> index. This procedure produced 186,774 training sequences and 40,989 validation sequences.

3.2.5 Normalization and Encoding

The integer-encoded input sequences were converted into floating-point representations and normalized by dividing each token index by the vocabulary size (731). This operation transformed the token IDs into normalized scalar values. The normalized inputs were subsequently expanded to include a single feature dimension, resulting in an input tensor with the shape [samples, sequence_length, 1].

In this implementation, token IDs were treated as normalized scalar values rather than categorical representations. No embedding layer or one-hot encoding was applied to the input sequences. Consequently, the numerical distances between token representations reflect their assigned vocabulary indices rather than explicitly encoded musical relationships. This design choice should be considered when interpreting the model's learning behavior and generated musical sequences.

The prediction targets were retained as integer class labels rather than converted into one-hot vectors. Accordingly, the LSTM model was trained using sparse categorical cross-entropy loss. The output layer produced a probability distribution over the 731 musical token classes, enabling the model to predict the next token from the preceding sequence.

3.3 LSTM Model Architecture

The proposed model employs a sequential deep learning architecture based on three stacked LSTM layers to capture temporal dependencies in symbolic musical sequences. Each LSTM layer consists of 256 hidden units. Batch Normalization is applied after each LSTM layer to support training stability, while Dropout is used to reduce overfitting, with dropout rates of 0.4, 0.3, and 0.2 for the first, second, and third LSTM layers, respectively.

The output of the third LSTM layer is passed to a Dense layer containing 128 neurons with ReLU activation. This layer is followed by Batch Normalization and a Dropout layer with a rate of 0.3. The final output layer consists of 731 neurons with Softmax activation, corresponding to the 731-class musical token vocabulary. Consequently, the model produces a probability distribution over the possible next musical tokens.

The model was optimized using the Adam optimizer with an initial learning rate of 0.001 and sparse categorical cross-entropy as the loss function. This loss function was selected because the target sequences were represented as integer class labels rather than one-hot encoded vectors. Model training was configured for a maximum of 25 epochs with a batch size of 256.

Several callbacks were employed during training. ModelCheckpoint monitored the validation loss and saved the model weights corresponding to the best validation performance. EarlyStopping monitored validation loss with a patience of five epochs and restored the best weights when training was stopped. In addition, ReduceLROnPlateau reduced the learning rate by a factor of 0.5 when the validation loss did not improve for two consecutive epochs, with a minimum learning rate of 0.00001. The best-performing checkpoint was subsequently used for all music generation experiments.

3.4 Music Generation

Music generation was performed using the trained LSTM model and an autoregressive prediction strategy. Each generation run was initialized with a seed sequence consisting of 100 musical tokens obtained from the validation subset. This seed sequence served exclusively as the initial context for autoregressive prediction and was not included in the generated output. The model subsequently predicted musical tokens one at a time until 300 new tokens had been generated. Only these newly generated tokens were converted into the final MIDI files and used for quantitative evaluation.

Two sampling strategies were investigated. The first strategy was deterministic greedy decoding (Argmax), in which the token with the highest predicted probability was selected at each generation step. The second strategy employed temperature-controlled sampling with three temperature settings: $T = 0.6$, $T = 0.7$, and $T = 1.0$. Temperature sampling modifies the predicted probability distribution before token selection. Specifically, the temperature parameter controls the sharpness of the probability distribution, thereby influencing the degree of sampling variability during generation.

In addition to the sampling strategies, two tempo-rendering modes were considered. The fixed-tempo mode maintained a constant tempo of 120 beats per minute (BPM) throughout the generated MIDI sequence. In contrast, the dynamic-tempo mode introduced tempo changes within the range of 60-140 BPM. Tempo values were randomly generated using a deterministic seed and updated every 40 generated events. The same tempo-rendering procedure was applied across the sampling strategies to enable controlled comparisons.

Importantly, tempo variation was externally imposed during MIDI rendering and was not predicted or learned by the LSTM model. Therefore, the dynamic-tempo procedure represents a rendering-level intervention rather than a learned tempo-generation capability. By excluding the seed tokens from the final MIDI files and evaluation, the quantitative metrics characterize the newly generated musical continuation rather than a mixture of seed and generated content. Nevertheless, the generation process remained conditioned on the validation-derived seed sequence. The generated token sequences were converted into MIDI files and subsequently evaluated using the quantitative musical metrics described in the following subsection.

3.5 Experimental Design

A controlled factorial experimental design was implemented to investigate the effects of sampling strategy and tempo-rendering mode on LSTM-based symbolic music generation. Two experimental factors were considered: (1) sampling strategy, consisting of four levels (Argmax, T = 0.6, T = 0.7, and T = 1.0), and (2) tempo-rendering mode, consisting of two levels (fixed tempo at 120 BPM and dynamic tempo within 60–140 BPM).
The combination of these factors resulted in eight experimental conditions:

- Argmax with fixed tempo (120 BPM).
- Argmax with dynamic tempo (60–140 BPM).
- Temperature sampling at T = 0.6 with fixed tempo.
- Temperature sampling at T = 0.6 with dynamic tempo.
- Temperature sampling at T = 0.7 with fixed tempo.
- Temperature sampling at T = 0.7 with dynamic tempo.
- Temperature sampling at T = 1.0 with fixed tempo.
- Temperature sampling at T = 1.0 with dynamic tempo.

All conditions used the same trained LSTM checkpoint, vocabulary, input sequence length, and generation length. To evaluate the variability of generated outputs, five deterministic generation seeds (42, 100, 2024, 777, and 999) were employed for each condition. The seed prompts were selected from the validation sequences and consistently reused across the corresponding experimental conditions. Each run generated 300 new musical tokens.

Consequently, the experimental design comprised 40 generation runs (eight conditions × five seeds). The resulting MIDI files were evaluated using multiple objective metrics. Results were summarized across the five generation runs using the mean and standard deviation to characterize the performance and variability of each experimental condition.

3.6 Evaluation

The generated MIDI files were evaluated using complementary quantitative metrics to characterize pitch diversity, rhythmic variation, local sequential similarity, pitch-distribution characteristics, repetition, chroma-distribution similarity, and tempo variation. The evaluation metrics included pitch entropy, rhythmic diversity, bigram and trigram overlap, register centroid offset, pitch Wasserstein distance, immediate repetition ratio, Chroma Jensen–Shannon Divergence (JSD), and tempo statistics. Each metric was computed for individual generated MIDI files and summarized across the five generation seeds using the mean and standard deviation.

1) Pitch Entropy

Pitch entropy measures the diversity of the pitch distribution in a generated musical sequence. It is calculated using Shannon entropy:

$$H(P) = -\sum_{i=1}^{K} p_i \log_2(p_i) \quad (1)$$

where p_i represents the relative frequency of pitch i, and K denotes the number of distinct pitches. Higher entropy indicates a more diverse pitch distribution, although it does not necessarily imply greater musical quality or coherence.

2) Rhythmic Diversity

Rhythmic diversity is assessed using Duration Diversity, Duration Entropy, Inter-Onset Interval (IOI) Diversity, and IOI Entropy. Duration Diversity represents the number of unique note-duration values in a generated sequence, whereas Duration Entropy measures the distributional diversity of these duration values. Similarly, IOI Diversity counts the unique time intervals between consecutive musical events, while IOI Entropy characterizes their distribution. For a discrete rhythmic feature distribution, entropy is calculated using Shannon entropy:

$$H(X) = -\sum_{j=1}^{M} q_j \log_2(q_j) \quad (2)$$

where q_j represents the relative frequency of the j-th duration or IOI value, and M denotes the number of distinct values. Higher diversity and entropy indicate greater variation in the corresponding rhythmic features.

3) N-gram Overlap (Bigram and Trigram)

N-gram overlap measures the degree of local sequential similarity between generated pitch sequences and the validation reference corpus. The reference corpus is constructed from the pitch events extracted from the 60 MIDI files in the validation subset. The pitch events are ordered chronologically before bigram and trigram extraction.

For each generated sequence, unique bigrams and trigrams are compared with the corresponding unique N-gram sets in the reference corpus. The overlap ratio is calculated as:

$$\text{N-gram Overlap} = \frac{|G_n \cap R_n|}{|G_n|} \quad (3)$$

where G_n denotes the set of unique generated N-grams, R_n denotes the set of unique reference N-grams, and n represents the N-gram order (2 for bigrams and 3 for trigrams). Higher values indicate that a greater proportion of the generated unique N-grams are also present in the reference corpus. This metric measures local sequence overlap rather than overall musical similarity or long-term structural coherence.

4) Pitch Distribution Distance

Two complementary metrics are used to examine the relationship between the pitch characteristics of generated music and the validation reference corpus.

First, Register Centroid Offset measures the absolute difference between the mean pitch of the generated sequence and that of the reference corpus:

$$D_{\text{centroid}} = |\mu_G - \mu_R| \quad (4)$$

where μ_G and μ_R represent the mean pitch values of the generated sequence and reference corpus, respectively. Lower values indicate closer alignment between their average pitch registers.

Second, Pitch Wasserstein Distance measures the difference between the generated and reference pitch distributions using the one-dimensional Wasserstein distance. Lower values indicate greater similarity between the two pitch distributions. Unlike the centroid offset, this metric considers the distribution of pitch values rather than only their mean.

5) Immediate Repetition Ratio

Immediate Repetition Ratio measures the proportion of consecutive pitch events that repeat the same pitch. It is calculated as:

$$R_{\text{repeat}} = \frac{\sum_{t=1}^{N-1} \mathbb{I}(p_t = p_{t+1})}{N-1} \quad (5)$$

where p_t denotes the pitch at position t, N represents the number of pitch events, and $\mathbb{I}(\cdot)$ is an indicator function that returns 1 when two consecutive pitches are identical and 0 otherwise. A lower value indicates fewer immediate consecutive pitch repetitions.

6) Tempo Variation Analysis

Tempo variation is evaluated using the mean tempo, tempo standard deviation, minimum and maximum BPM, and tempo entropy. These measures characterize the temporal variation introduced during MIDI rendering.

The mean tempo represents the average BPM across tempo events, while the standard deviation measures the variability of tempo values. Tempo entropy characterizes the distribution of tempo values using Shannon entropy. Higher standard deviation and entropy indicate greater variation in the applied tempo sequence.

In this study, dynamic tempo is externally imposed during MIDI rendering, with tempo values varied within the range of 60–140 BPM and updated every 40 generated events. Therefore, tempo-related metrics represent the effect of the dynamic-tempo rendering procedure rather than tempo characteristics learned by the LSTM model. The fixed-tempo condition uses a constant tempo of 120 BPM.

7) Chroma Jensen–Shannon Divergence

(Chroma JSD) was used to measure the difference between the harmonic pitch-class distribution of generated music and that of the reference corpus. Each MIDI sequence was converted into a 12-dimensional chroma distribution representing the relative occurrence of the twelve pitch classes (C–B). The generated and reference chroma distributions were normalized to sum to one. Chroma JSD was calculated as:

$$\text{JSD}(P \parallel Q) = \frac{1}{2} D_{\text{KL}}(P \parallel M) + \frac{1}{2} D_{\text{KL}}(Q \parallel M), \qquad M = \frac{P+Q}{2} \quad (6)$$

where P and Q represent the normalized chroma distributions of the generated music and reference corpus, respectively; M is their average distribution; and D_{KL} denotes Kullback–Leibler divergence. Lower Chroma JSD values indicate greater similarity between the generated and reference harmonic pitch-class distributions. The metric was computed for each generated MIDI sample and summarized across five generation seeds for each experimental condition.

3.7 Multi-Metric Visualization

A normalized radar chart is used to provide an integrated visualization of the quantitative evaluation results across the eight experimental conditions. The visualization includes complementary metrics representing pitch diversity, rhythmic diversity, pitch-distribution similarity, local sequential overlap, and immediate repetition. Before visualization, the selected metrics are normalized to a common scale to facilitate comparison across different measurement ranges. Metrics for which higher values indicate greater diversity or overlap are interpreted accordingly, whereas distance and repetition metrics are transformed or interpreted in the opposite direction when constructing a higher-is-better profile. The radar chart compares the four sampling strategies (Argmax, T = 0.6, T = 0.7, and T = 1.0) under both fixed and dynamic tempo-rendering modes. This visualization is intended to illustrate the trade-offs among the evaluated musical characteristics rather than establish a universally optimal generation condition.

4 Results and Discussion

4.1 Experimental Setup

The experiments were conducted in a local Jupyter Lab environment. The computational hardware and software specifications are summarized in Table 1. The dataset was stored locally to ensure faster and more responsive data processing and model training.

Table 1. Computational Environment and Hardware Specifications

Component	Specification
Processor	AMD Ryzen 5 3500x
RAM	32 GB
GPU	NVIDIA GeForce GTX 1660 Super
CUDA Version	12.6
GPU Version	560.94
Python Version	3.8.20

The experiments evaluated four sampling strategies: Argmax, temperature sampling with T = 0.6, T = 0.7, and T = 1.0. Each strategy was combined with two tempo-rendering configurations: fixed tempo at 120 BPM and dynamic tempo within the range of 60–140 BPM. This factorial design resulted in eight experimental conditions.

Five generation seeds were used for each condition, resulting in 40 generated MIDI files. All conditions used the same trained LSTM checkpoint and corresponding seed contexts to maintain consistency across generation experiments. The reported values are presented as mean ± standard deviation (SD) across the five generation runs. Thus, the reported variability reflects differences between generation runs using a shared checkpoint, rather than variability across independently trained models.

The evaluation considered pitch entropy, rhythmic diversity, immediate repetition ratio, bigram and trigram overlap, pitch distribution distances, Chroma Jensen-Shannon Divergence (Chroma JSD), and tempo statistics. These metrics provide complementary perspectives on the generated sequences, including pitch and rhythmic diversity, local repetition, sequential-pattern similarity to the validation corpus, pitch-distribution characteristics, pitch-class distribution similarity, and temporal variation.

4.2. Effect of Sampling Temperature on Pitch Diversity

Pitch entropy was used to quantify the diversity of pitch distributions in the generated MIDI sequences. Higher entropy indicates a more diverse distribution of generated pitches.

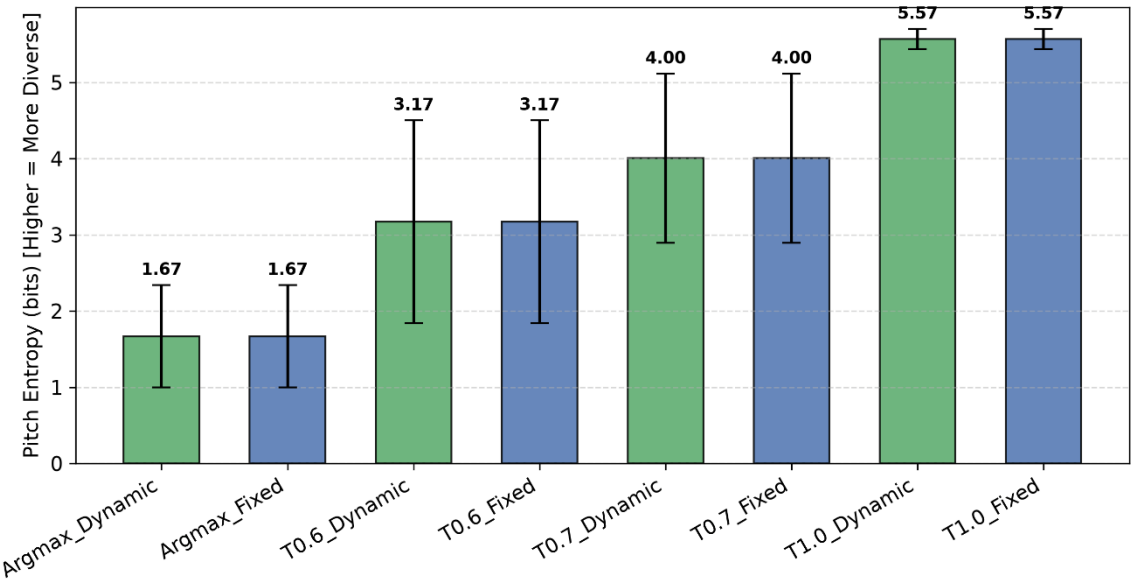


Figure 2 Pitch entropy across sampling strategies and tempo-rendering configurations (mean ± SD, n = 5). Higher values indicate greater pitch diversity.

Figure 2 presents the mean pitch entropy across the four sampling strategies under fixed and dynamic tempo configurations. Argmax produced the lowest pitch entropy, with a mean of 1.67 bits. The entropy increased to 3.17 bits for T = 0.6 and 4.00 bits for T = 0.7. The highest value was obtained with T = 1.0, reaching 5.57 bits.

These results indicate that increasing the sampling temperature is associated with greater pitch diversity. In deterministic Argmax sampling, the model repeatedly selects the most probable token, which can restrict the range of generated pitches. Temperature sampling introduces greater stochasticity into token selection, allowing less probable alternatives to be selected and increasing the diversity of the resulting pitch distribution.

An important observation is that pitch entropy remains identical between fixed and dynamic tempo conditions at each sampling setting. This indicates that the tempo-rendering procedure does not change the pitch-token sequence generated by the LSTM. Instead, the observed differences in pitch entropy are associated with the sampling strategy.

4.3. Immediate Repetition and Sequential Diversity

Immediate repetition ratio was used to measure the proportion of consecutive pitch events that repeat the preceding pitch. Lower values indicate fewer immediate repetitions. As shown in Figure 3, Argmax produced the highest immediate repetition ratio, with a mean of 0.62. The ratio decreased to 0.46 for T = 0.6, 0.26 for T = 0.7, and 0.10 for T = 1.0. The results show a consistent decrease in immediate repetition as the sampling temperature increases. This pattern is complementary to the pitch entropy findings, suggesting that higher temperatures produce more varied pitch sequences and reduce the tendency to generate consecutive repetitions.

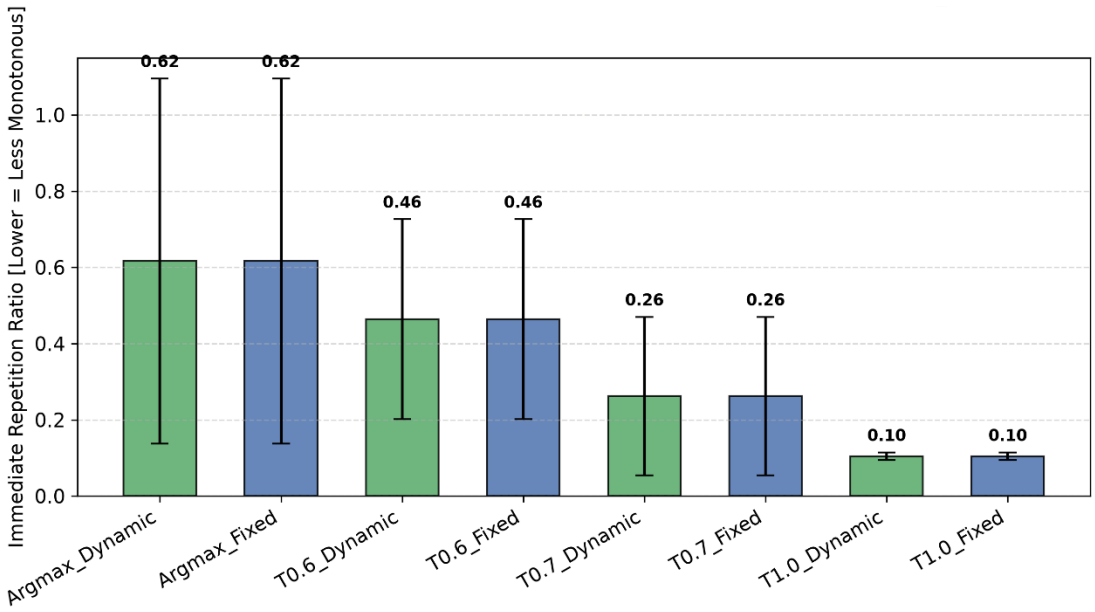


Figure 3 Immediate repetition ratio across sampling strategies and tempo-rendering configurations (mean ± SD, n = 5). Lower values indicate fewer consecutive repeated pitch events.

However, a lower immediate repetition ratio should not be interpreted as direct evidence of improved musical quality. Repetition is also a natural component of musical structure, and reducing it excessively may affect the coherence or stylistic consistency of a generated composition. Therefore, this metric should be interpreted together with other measures of sequential similarity and pitch distribution. The error bars also indicate variability across the five generation runs. In particular, the Argmax and T = 0.6 conditions exhibit relatively large standard deviations, whereas T = 1.0 shows a consistently low repetition ratio in the evaluated runs.

4.4 Rhythmic Diversity and Temporal Dynamics

Rhythmic diversity was evaluated using Duration Diversity, Duration Shannon Entropy, Inter-Onset Interval (IOI) Diversity, and IOI Shannon Entropy. These metrics characterize the variety and distribution of note durations and temporal intervals in the generated sequences. Figure 4 presents the rhythmic metrics across the four sampling strategies under fixed- and dynamic-tempo rendering conditions. Duration Diversity remained constant at 2.0 across all strategies, indicating that the generated sequences contained the same number of distinct duration values. In contrast, Duration Shannon Entropy increased from 0.41 bits for Argmax to 0.87 bits for T = 0.6, reaching a maximum of 0.89 bits at T = 0.7 before slightly decreasing to 0.87 bits at T = 1.0. This pattern indicates that temperature sampling altered the distribution of duration values without increasing the number of distinct durations.

The IOI-based metrics exhibited a different trend. IOI Diversity increased from 2.0 for Argmax to 2.2, 2.4, and 2.8 for T = 0.6, T = 0.7, and T = 1.0, respectively. Similarly, IOI Shannon Entropy increased progressively from 0.23 bits for Argmax to 0.36 bits for T = 0.6, 0.51 bits for T = 0.7, and 0.67 bits for T = 1.0. These findings suggest that higher sampling temperatures were associated with greater variability in inter-onset intervals in the generated sequences.

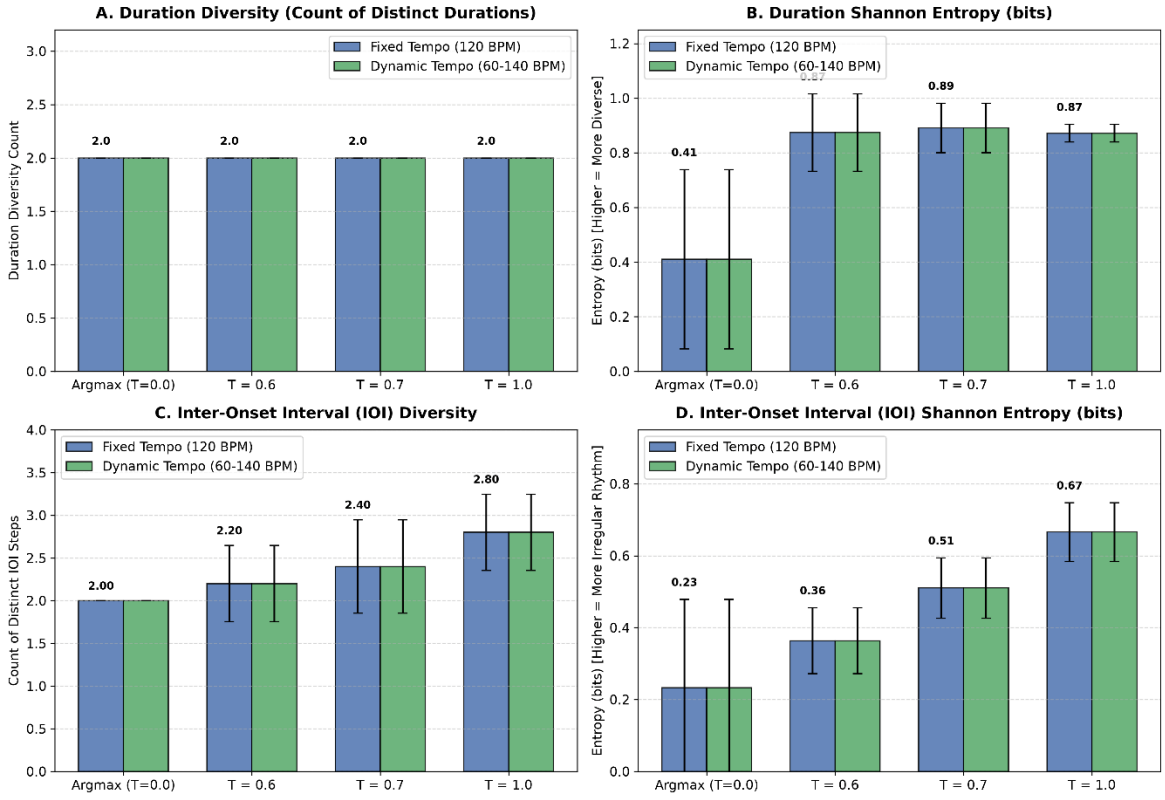


Figure 4 Rhythmic and temporal dynamics across sampling strategies and tempo-rendering conditions. (A) Duration diversity, (B) duration Shannon entropy, (C) inter-onset interval (IOI) diversity, and (D) IOI Shannon entropy. Bars represent mean values across five generation seeds, and error bars indicate sample standard deviation (SD). Fixed- and dynamic-tempo conditions yield identical symbolic duration and IOI metrics because tempo variation is applied during MIDI rendering without modifying symbolic event durations or onset positions.

The fixed and dynamic tempo conditions yielded identical rhythmic metric values for each sampling strategy. This outcome was expected because tempo variation was applied during MIDI rendering rather than predicted by the LSTM. Thus, differences in symbolic rhythmic metrics were associated with the sampling strategy and generated event sequences, whereas tempo rendering affected playback timing. Importantly, greater rhythmic diversity should not be interpreted as evidence of improved musical quality or rhythmic coherence.

4.5. N-gram Overlap with the Validation Corpus

N-gram overlap was examined to assess the similarity of local pitch patterns between generated sequences and the validation corpus. Bigram overlap represents the overlap of two-event pitch patterns, whereas trigram overlap represents the overlap of three-event patterns.

Figure 5 shows that Argmax achieved a bigram overlap of 95.0% and a trigram overlap of 42.5%. T = 0.6 achieved the highest bigram overlap at 95.2%, with a trigram overlap of 33.4%. For T = 0.7, the corresponding values were 93.7% and 28.6%, respectively. T = 1.0 produced the lowest overlap, with 90.9% for bigrams and 23.2% for trigrams.

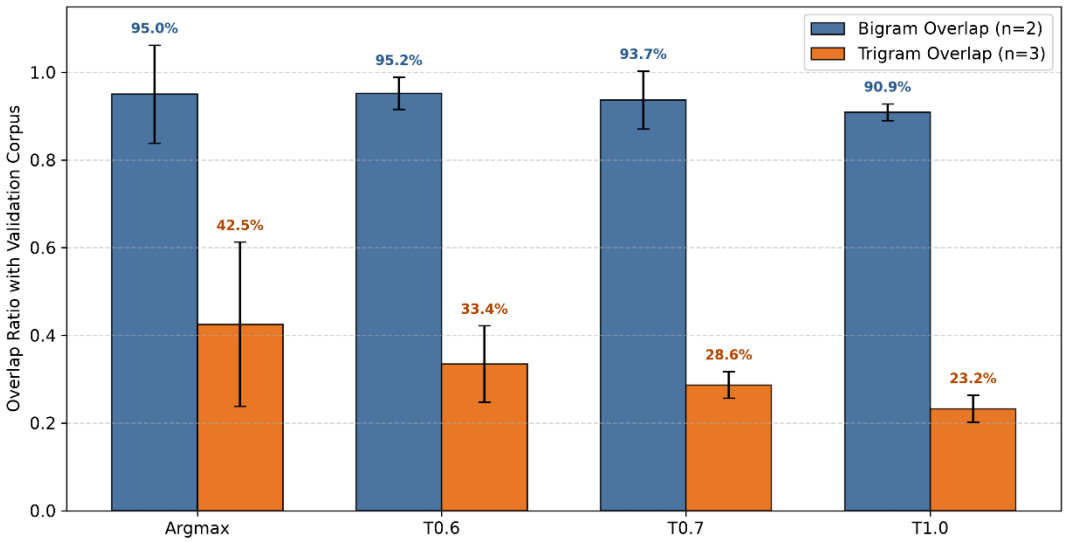


Figure 5 Bigram and trigram overlap between generated sequences and the validation corpus across four sampling strategies (mean ± SD, n = 5).

The results suggest that higher sampling temperatures are associated with lower local-pattern overlap with the validation corpus. This reduction is more pronounced for trigram patterns than for bigram patterns, indicating that changes in sampling temperature affect longer local pitch sequences. Nevertheless, lower n-gram overlap does not necessarily imply better musical generation. Higher overlap may indicate that generated sequences retain more patterns observed in the reference corpus, whereas lower overlap may reflect greater variation from those patterns. Therefore, n-gram overlap provides information about reference-pattern similarity rather than a direct measure of musical quality.

4.6. Chroma Distribution Similarity

Chroma Jensen-Shannon Divergence (JSD) was used to evaluate the similarity between the chroma distributions of generated music and the reference corpus. Chroma features represent pitch-class information by mapping musical pitches into twelve classes corresponding to the twelve semitones in an octave. JSD quantifies the divergence between two probability distributions and provides a symmetric measure of distributional similarity. Lower JSD values indicate that the generated chroma distribution is closer to the reference distribution, whereas higher values indicate greater divergence.

The Chroma JSD results for each sampling strategy, tempo-rendering condition, and generation seed are reported in Table 2. The table presents the individual seed-level values together with their mean and standard deviation, enabling comparison of chroma-distribution similarity across the experimental conditions. The results show a progressive decrease in mean Chroma JSD from Argmax (0.5900 ± 0.1243) to T = 0.6 (0.3486 ± 0.1776), T = 0.7 (0.2410 ± 0.1069), and T = 1.0 (0.0244 ± 0.0163). Among the evaluated sampling strategies, T = 1.0 exhibited the lowest divergence and therefore the closest pitch-class distribution to the reference corpus according to this metric.

The Fixed and Dynamic tempo conditions produced identical Chroma JSD values for every generation seed. This observation is consistent with the experimental design, in which dynamic tempo was applied during MIDI rendering without modifying the generated pitch sequence. Thus, the differences in Chroma JSD are associated with sampling strategy rather than tempo-rendering mode.

Table 2. Chroma JSD Results for Different Sampling Strategies

Sampling Strategy	Tempo Mode	Seed 42	Seed 100	Seed 2024	Seed 777	Seed 999	Mean	SD
Argmax	Fixed	0.4300	0.6675	0.4820	0.6722	0.6984	0.5900	0.1243
Argmax	Dynamic	0.4300	0.6675	0.4820	0.6722	0.6984	0.5900	0.1243
T = 0.6	Fixed	0.3273	0.1676	0.3170	0.2858	0.6454	0.3486	0.1776
T = 0.6	Dynamic	0.3273	0.1676	0.3170	0.2858	0.6454	0.3486	0.1776
T = 0.7	Fixed	0.2426	0.1639	0.1489	0.2320	0.4177	0.2410	0.1069
T = 0.7	Dynamic	0.2426	0.1639	0.1489	0.2320	0.4177	0.2410	0.1069
T = 1.0	Fixed	0.0210	0.0136	0.0513	0.0262	0.0099	0.0244	0.0163
T = 1.0	Dynamic	0.0210	0.0136	0.0513	0.0262	0.0099	0.0244	0.0163

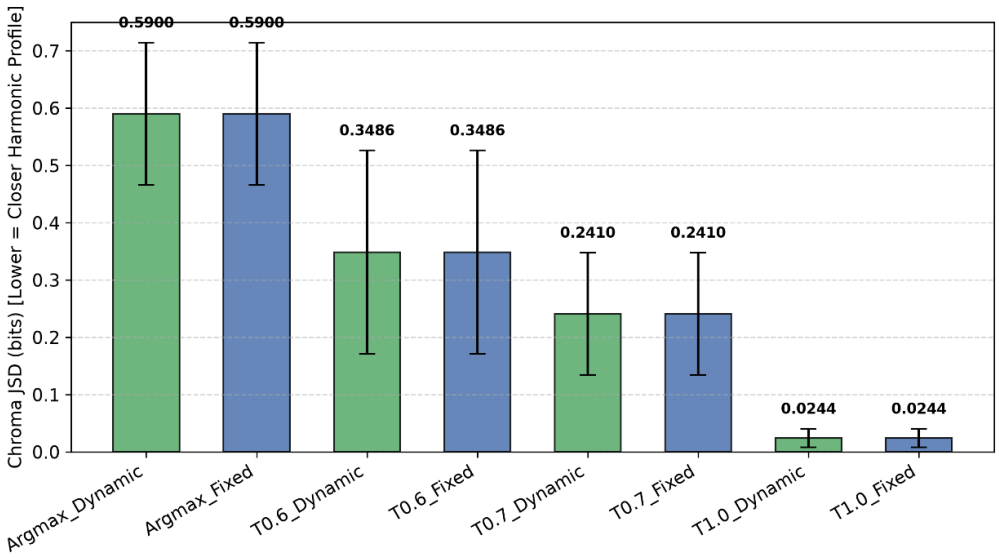


Figure 6 Chroma Jensen–Shannon Divergence (JSD) across sampling strategies under fixed and dynamic tempo rendering. Lower values indicate greater similarity between generated and reference chroma distributions. Error bars represent standard deviations across five generation seeds.

These results indicate that increasing sampling temperature was associated with a closer match between the generated and reference chroma distributions in this experiment. However, Chroma JSD measures distributional similarity rather than melodic coherence, harmonic correctness, or perceived musical quality. Therefore, it should be interpreted alongside pitch diversity, repetition, and n-gram overlap metrics.

The fixed- and dynamic-tempo conditions produced identical Chroma JSD values at each sampling temperature. This finding is consistent with the experimental design, in which tempo variation was applied during MIDI rendering without modifying the generated pitch-token sequence. Consequently, the observed differences in chroma distribution are associated with the sampling strategy rather than tempo rendering.

4.7. Pitch Distribution Characteristics

To further investigate the relationship between generated pitch distributions and the validation corpus, two distributional distance metrics were examined: Register Centroid Offset and Pitch Wasserstein Distance.

Register Centroid Offset measures the absolute difference between the pitch centroids of generated and validation sequences. Pitch Wasserstein Distance measures the difference between their pitch distributions. For both metrics, lower values indicate greater distributional similarity to the validation corpus.

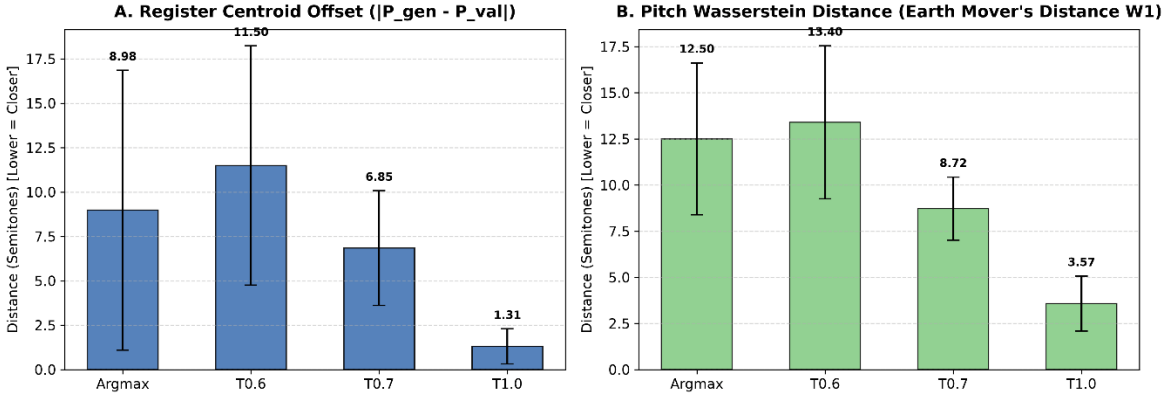


Figure 7 Pitch distribution distances across sampling strategies: (A) Register Centroid Offset and (B) Pitch Wasserstein Distance. Lower values indicate greater similarity to the validation pitch distribution (mean ± SD, n = 5).

As presented in Figure 7, T = 1.0 achieved the lowest Register Centroid Offset, with a mean of 1.31 semitones. This was followed by T = 0.7 at 6.85 semitones, Argmax at 8.98 semitones, and T = 0.6 at 11.50 semitones. A similar pattern was observed for Pitch Wasserstein Distance. T = 1.0 achieved the lowest mean distance of 3.57, followed by T = 0.7 at 8.72, Argmax at 12.50, and T = 0.6 at 13.40. These findings indicate that T = 1.0 produced a pitch distribution that was closer to the validation distribution according to both evaluated distance metrics. This is an important observation because higher sampling temperature not only increased pitch diversity but also resulted in a pitch distribution that was more similar to the validation corpus according to these measures.

However, distributional similarity does not establish melodic or harmonic similarity. These metrics do not directly evaluate melodic contour, harmonic progression, or long-range musical structure. Consequently, their results should be interpreted as evidence of pitch-distribution characteristics rather than as a comprehensive assessment of musical quality.

4.8. Effect of Fixed and Dynamic Tempo Rendering

The tempo experiment was designed to distinguish the effect of sampling temperature from the effect of tempo variation during MIDI rendering. Fixed rendering maintained a constant tempo of 120 BPM, whereas dynamic rendering introduced tempo changes within the range of 60–140 BPM.

Figure 8 compares pitch entropy under fixed and dynamic tempo rendering at matched sampling temperatures. The results show that pitch entropy remains identical between the two tempo configurations for all four sampling strategies. This pattern is consistent with the other pitch-based metrics, which also remain unchanged between matched fixed and dynamic conditions. These findings indicate that tempo variation is applied during the MIDI-rendering stage and does not modify the pitch-token sequences generated by the LSTM. Therefore, the differences in pitch diversity and immediate repetition across sampling temperatures should be attributed to the sampling process rather than to tempo rendering.

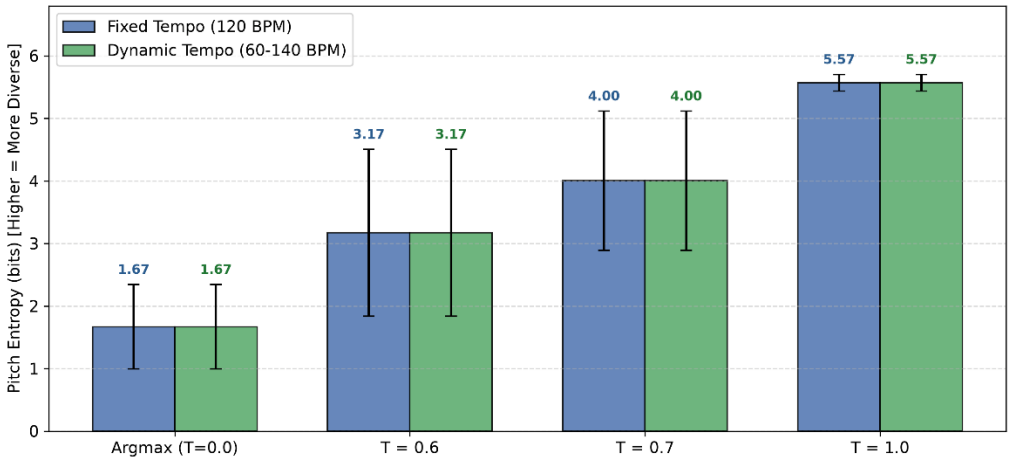


Figure 8 Comparison of pitch entropy under fixed and dynamic tempo rendering at matched sampling temperatures (mean ± SD, n = 5).

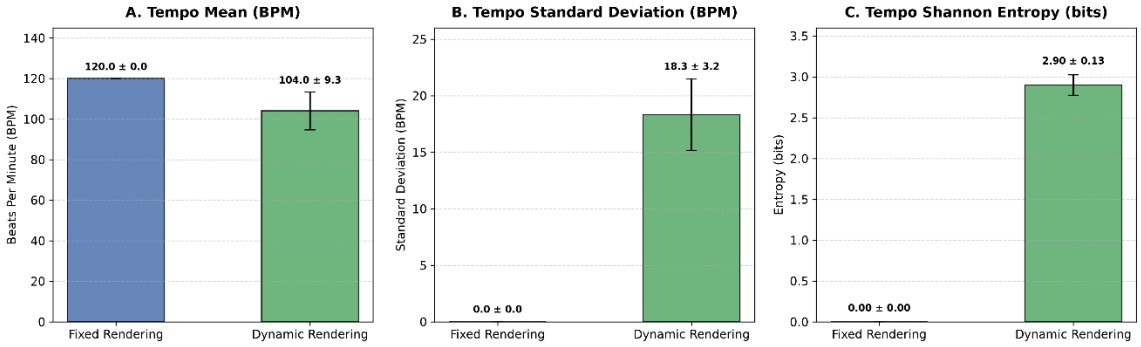


Figure 9 Tempo variation under fixed and dynamic rendering: (A) mean tempo, (B) tempo standard deviation, and (C) tempo Shannon entropy.

Figure 9 further illustrates the temporal characteristics of the two rendering configurations. Fixed rendering maintained a mean tempo of 120.0 BPM, with zero tempo standard deviation and zero tempo entropy. In contrast, dynamic rendering produced a mean tempo of approximately 104.0 BPM, a tempo standard deviation of approximately 18.3 BPM, and a Shannon entropy of approximately 2.90 bits. The results demonstrate that dynamic tempo rendering introduces temporal variation independently of the sampling temperature. This distinction is important because it separates the influence of stochastic token selection from the temporal presentation of the generated MIDI output.

4.9. Integrated Discussion and Multi-Metric Trade-off

The results demonstrate that sampling temperature influences multiple characteristics of LSTM-generated music, including pitch diversity, repetition, local sequence similarity, and pitch-distribution characteristics. Higher temperature increased pitch diversity and reduced immediate repetition, while lower-temperature configurations retained greater local n-gram overlap with the reference corpus. These findings reveal a trade-off between generating diverse sequences and preserving local patterns observed in the reference data.

The Chroma JSD results provide an additional perspective on this trade-off. T = 1.0 achieved the lowest Chroma JSD, indicating the closest chroma-distribution similarity to the reference corpus among the evaluated sampling strategies. This finding shows that greater pitch diversity does not

necessarily correspond to greater divergence in pitch-class distribution. Nevertheless, distributional similarity alone cannot establish melodic coherence or perceived musical quality.

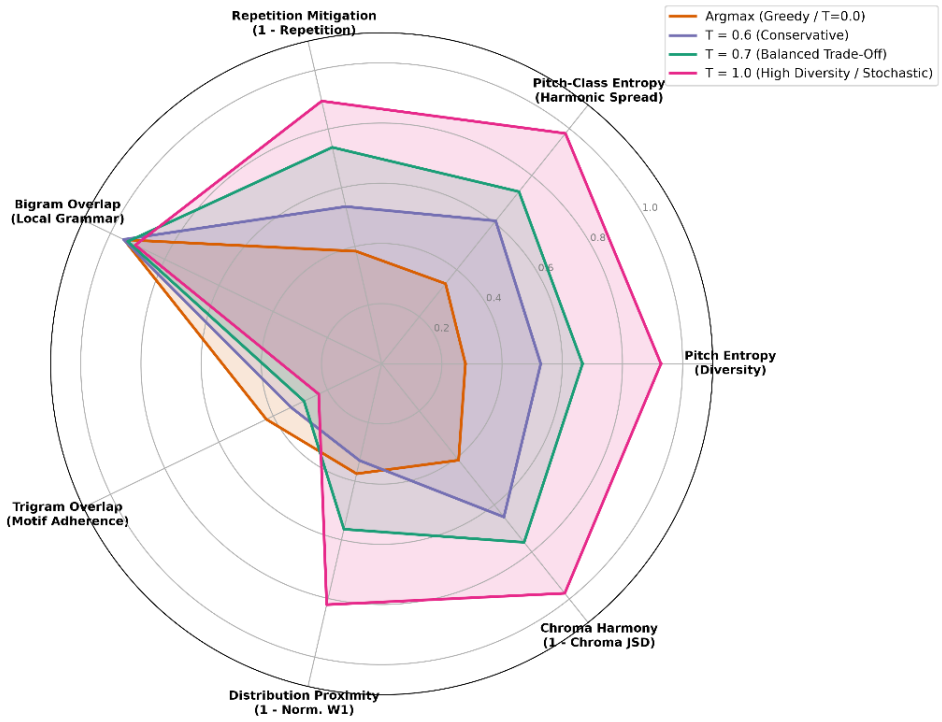


Figure 10 Normalized multi-metric comparison of sampling strategies. Metrics with lower-is-better directions were inverted so that higher normalized values indicate more favorable metric-specific outcomes. The radar chart summarizes metric-level trade-offs and does not represent an overall measure of musical quality.

Figure 10 summarizes the multi-metric trade-off across sampling strategies. The radar chart illustrates that each configuration exhibits different strengths across diversity, repetition mitigation, local sequence similarity, and distributional proximity. These metrics describe distinct aspects of generated music and should not be interpreted as interchangeable measures of overall musical quality. In particular, neither high diversity nor high reference-pattern overlap independently guarantees a musically meaningful result.

The factorial experiment further separates the effects of sampling temperature and tempo rendering. Temperature affects stochastic token selection, whereas dynamic tempo is applied during MIDI rendering. The identical pitch-based and chroma-based metrics under matched fixed- and dynamic-tempo conditions confirm that tempo rendering does not alter the generated pitch sequence. Its contribution is therefore limited to temporal presentation in the current implementation.

Several limitations should be considered when interpreting these findings. The five generation seeds were evaluated using a shared trained checkpoint and do not represent independent model-training runs. An additional limitation concerns the numerical representation of musical tokens. In this study, token IDs were normalized and treated as scalar values rather than represented using an embedding layer or one-hot encoding. Because token IDs are categorical identifiers without inherent musical ordering or distance, this representation may introduce artificial numerical relationships between tokens. Consequently, the model's learning behavior may be influenced by the assigned vocabulary indices rather than exclusively by the underlying musical characteristics. The findings should therefore be interpreted in the context of this specific token representation.

Moreover, the evaluation relies primarily on computational MIDI metrics and does not directly measure human perception. Future work should include multiple independently trained models,

embedding-based or alternative categorical representations, listening-based evaluation, and models that jointly learn pitch and expressive timing. Such investigations would provide a more comprehensive assessment of the robustness and musical characteristics of LSTM-generated compositions.

5 Conclusion

This study investigated the effects of temperature sampling and tempo variation on LSTM-based music generation. The results demonstrate that sampling temperature substantially influences the diversity and sequential characteristics of generated MIDI sequences. Higher temperatures, particularly T = 1.0, produced greater pitch entropy, lower immediate repetition, and smaller pitch-distribution distances. Rhythmic analysis further showed that IOI diversity and entropy increased with sampling temperature, whereas duration entropy reached its highest mean at T = 0.7. However, lower-temperature configurations retained higher N-gram overlap with the validation corpus, revealing a trade-off between pitch diversity and reference-pattern similarity. These findings indicate that no single sampling configuration can be considered universally optimal across all evaluation metrics.

The factorial evaluation demonstrated that dynamic-tempo rendering introduces playback variation independently of pitch-sequence generation. While sampling temperature affects stochastic token selection, tempo variation is externally applied during MIDI rendering without changing the generated symbolic pitch sequences, note durations, or symbolic onset positions. This separation provides a clearer understanding of the respective roles of sampling strategy and tempo rendering in LSTM-based music generation.

This study has several limitations. First, the five generation seeds were evaluated using a shared trained checkpoint and therefore do not represent independent model-training runs. Second, musical tokens were represented using normalized scalar token IDs rather than embedding-based or one-hot representations. Since token IDs are categorical identifiers without inherent musical ordering or distance, this encoding may introduce artificial numerical relationships among tokens and potentially influence the model's learning behaviour. Third, the evaluation relies primarily on quantitative MIDI-based metrics, which characterize statistical musical properties but do not directly establish perceived musical quality or coherence. Future research should incorporate multiple independent model-training runs, embedding-based or alternative categorical representations, human perceptual assessments, and integrated models capable of learning pitch and expressive tempo jointly. These directions may provide a more comprehensive evaluation of the robustness and musical characteristics of neural music generation systems.

References

[1] Y. Zhang, "Using LSTM neural network to generate music," *Theoretical and Natural Science*, vol. 19, no. 1, pp. 111–115, Dec. 2023. doi: 10.54254/2753-8818/19/20230512.

[2] A. Joshi, A. Gokhale, A. Khandekar, A. Kesar, S. Khade, and N. Tarapore, "Music Generation Using Recurrent Neural Networks," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 12, pp. 1352–1358, Dec. 2022. doi: 10.22214/ijraset.2022.48200.

[3] M. Conner, L. Gral, K. Adams, D. Hunger, R. Strelow, and A. Neuwirth, "Music Generation Using an LSTM," *arXiv*, 2022. doi: 10.48550/arXiv.2203.12105.

[4] N. Kotecha and P. Young, "Generating Music using an LSTM Network," *arXiv*, 2018. doi: 10.48550/arXiv.1804.07300.

[5] S. Mangal, R. Modak, and P. Joshi, "LSTM Based Music Generation System," *IARJSET*, vol. 6, no. 5, pp. 47–54, May 2019, doi: 10.17148/IARJSET.2019.6508.

[6] R. V. Uppuluri, "Music Generation Using Recurrent Neural Networks : An LSTM Approach to Melodic and Harmonic Composition," *International Journal of Research Publication and Reviews*, no. 5, pp. 1623–1628, 2024. [Online]. Available: https://ijrpr.com/uploads/V5ISSUE10/IJRPR33995.pdf

[7] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: learning expressive musical performance," *Neural Comput Appl*, vol. 32, no. 4, pp. 955–967, Feb. 2020. doi: 10.1007/s00521-018-3758-9.

[8] S. A. Fathima, S. Hariram, S. M. Kanagalingam, B. M. Nitish, and M. M. Muthu, "Neural Harmony: Advancing Composition with RNN-LSTM in Music Generation," in *2024 IEEE International Conference on Contemporary Computing and Communications (InC4)*, IEEE, Mar. 2024, pp. 1–6. doi: 10.1109/InC460750.2024.10649223.

[9] K. Soundarya, K. R. Pandik, S. Samsher, V. S. Kishore, and A. S. Lavanya, "Music Generation Using Recurrent Neural Network," *Journal of Nonlinear Analysis and Optimization*, vol. 15, no. 02, pp. 913–919, 2024. doi: 10.36893/JNAO.2024.V15I2.142.

[10] G. Keerti, A. Vaishnavi, P. Mukherjee, A. Vidya, G. S. Sreenithya, and D. Nayab, "Attentional networks for music generation," *Multimed Tools Appl*, vol. 81, pp. 5179–5189, 2020. doi: 10.1007/s11042-021-11881-1.

[11] D. Fudholi, D. N. A. Putri, R. B. M. A. A. Wijaya, J. E. Kusnadi, and J. C. Amarissa, "The Application of LSTM in the AI-Based Enhancement of Classical Compositions," *Journal of Informatics Information System Software Engineering and Applications (INISTA)*, 2024. doi: 10.20895/inista.v7i1.1628.

[12] M. K. Jedrzejewska, A. Zjawinski, and B. Stasiak, "Generating Musical Expression of MIDI Music with LSTM Neural Network," in *2018 11th International Conference on Human System Interaction (HSI)*, IEEE, Jul. 2018, pp. 132–138. doi: 10.1109/HSI.2018.8431033.

[13] J. Huang, "Multitrack Symbolic Music Composition with LSTM-GAN," *2022 2nd International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR)*, pp. 35–40, 2022. doi: 10.1109/ICBAR58199.2022.00014.

[14] S. Chikkamath and S. R. Nirmala, "Melody generation using LSTM and BI-LSTM Network," *2021 International Conference on Computational Intelligence and Computing Applications (ICCICA)*, pp. 1–6, 2021. doi: 10.1109/iccica52458.2021.9697286.

[15] B. Nag, A. I. Middya, and S. Roy, "Melody generation based on deep ensemble learning using varying temporal context length," *Multim. Tools Appl.*, vol. 83, pp. 69647–69668, 2024. doi: 10.1007/s11042-024-18270-4.

[16] A. Dhar and A. Victor, "Neural Harmony: Advancing Polyphonic Music Generation and Genre Classification through LSTM-Based Networks," *2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pp. 1–6, 2024. doi: 10.1109/CONECCT62155.2024.10677254.

[17] A. Elmsley, T. Weyde, and N. Armstrong, "Generating Time: Rhythmic Perception, Prediction and Production with Recurrent Neural Networks," vol. 1, 2017. doi: 10.5920/JCMS.2017.04.

[18] A. Ycart and E. Benetos, "Learning and Evaluation Methodologies for Polyphonic Music Sequence Prediction With LSTMs," *IEEE/ACM Trans Audio Speech Lang Process*, vol. 28, pp. 1328–1341, 2020, doi: 10.1109/TASLP.2020.2987130.

[19] S. Chakraborty, S. Kishor, S. Patil, and J. Timoney, "LeaderSTeM-A LSTM model for dynamic leader identification within musical streams," *Zenodo*, 2020. doi: 10.5281/zenodo.4285378.

[20] Y. Huang, X. Huang, and Q. Cai, "Music Generation Based on Convolution-LSTM," *Computer and Information Science*, vol. 11, no. 3, p. 50, Jun. 2018, doi: 10.5539/cis.v11n3p50.

[21] K. Gupta, K. Srivastava, K. Singh, and R. Pillai, "Music Generation and Retrieval Using LSTM and VAE," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, Jul. 2023, pp. 1–6. doi: 10.1109/ICCCNT56998.2023.10307558.

[22] A. Agarwal, "Application of Music Retrieval & Generation," *International Journal Of Scientific Research In Engineering And Management*, vol. 08, no. 05, pp. 1–5, May 2024. doi: 10.55041/IJSREM34003.

[23] J. Liu, "Expressive Music Data Processing and Generation," *arXiv*, 2025. doi: 10.48550/arXiv.2503.11896.

[24] D. Evangeline, M. Bhuvan, A. Parkavi, A. Manikarnike, M. Harikrishnan, and B. S. Adiga, "Automated Music Synthesis Using Recurrent Neural Network and Long Short Term Memory," in *2023 International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS)*, IEEE, Oct. 2023. doi: 10.1109/ICCAMS60113.2023.10525888.

[25] A. A. S. Gunawan, A. P. Iman, and D. Suhartono, "Automatic music generator using recurrent neural network," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 645–654, 2020. doi: 10.2991/ijcis.d.200519.001.

[26] M. Djamaluddin, I. Fathurrahman, and M. N. Wathani, "Pengembangan Model Deep Learning Long Short-Term Memory untuk Generasi Musik Berbasis Data MIDI [Development of a Long Short-Term Memory Deep Learning Model for Music Generation Based on MIDI Data]," *Infotek: Jurnal Informatika dan Teknologi*, vol. 9, no. 1, pp. 197–208, Jan. 2026, doi: 10.29408/jit.v9i1.33165.