Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

EdgeMusicTransformer: A Lightweight Context-Aware Transformer for Efficient Generative AI Deployment under Edge Computing Constraints

Yifan Zhang

YifanZhang22@outlook.com

School of Art and Design, Anqing Normal University, Anqing 246000, Anhui, China

Xiangyu Wang

School of Music and Dance, China West Normal University, Nanchong 637001, Sichuan, China

Article

Keywords: Generative Artificial Intelligence, Edge Computing, Lightweight Transformer, Efficient Deep Learning, Context-Aware Inference, Symbolic Music Generation, Transformer Architecture

Posted Date: October 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11183677/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Scientific Reports on November 22nd, 2026. See the published version at https://doi.org/10.1038/s41598-026-81420-1.

EdgeMusicTransformer: A Lightweight Context-Aware Transformer for Efficient Generative AI Deployment under Edge Computing Constraints

Yifan Zhang[1]*, Xiangyu Wang[2]
[1*]School of Art and Design, Anqing Normal University, Anqing 246000, Anhui, China
[2]School of Music and Dance, China West Normal University, Nanchong 637001, Sichuan, China
Corresponding Author Email: **YifanZhang22@outlook.com**

Abstract
New types of symbolic music creation have been made possible by recent developments in generative artificial intelligence; however, Transformer-based models' computational complexity, memory requirements, and inference overhead continue to be significant obstacles for deployment on edge platforms with limited resources. A lightweight, context-aware Transformer architecture for effective generative AI inference at the edge is presented in this paper: EdgeMusicTransformer. In order to minimise computational cost and preserve the ability to describe sequential dependencies in symbolic music data, the suggested architecture combines lightweight multi-head self-attention with a compact feed-forward transformation module. The created model uses a compact 397-token symbolic representation based on REMI encoding and has 2,443,149 trainable parameters. Four token configurations—128, 256, 384, and 512—are used to methodically examine the effects of inference context length. According to experimental results, the 256-token setup achieves the best throughput of 17,783.53 tokens/s, while increasing context length increases inference latency from 7.809 ms to 33.991 ms. Additionally, peak memory use shows steady computational behaviour across various context settings, staying between 248.148 and 259.699 MB. Pitch distribution, pitch diversity, duration characteristics, note density, and polyphonic structure are among the objective structural descriptors that are used to further analyse generated symbolic sequences. The results show that EdgeMusicTransformer offers a small and computationally effective method for implementing Transformer-based generative models in edge-oriented settings.

Keywords
Generative Artificial Intelligence; Edge Computing; Lightweight Transformer; Efficient Deep Learning; Context-Aware Inference; Symbolic Music Generation; Transformer Architecture

1. Introduction
The creation of digital material has been profoundly changed by recent developments in artificial intelligence (AI), which allow intelligent systems to generate intricate and imaginative outputs in fields like music, language, and vision. However, implementing AI applications on devices with limited resources has become extremely difficult due to the growing computational demands of contemporary deep learning models. By bringing AI capabilities closer to data sources, lowering communication overhead, enhancing reaction latency, and allowing intelligent services in distributed situations, edge intelligence has become a promising paradigm [1,2]. For real-time applications, where low latency, energy efficiency, and computational efficiency are crucial criteria, the integration of AI models with edge computing has grown in significance [3]. The need to create small models that can function well with constrained hardware resources is further highlighted by recent advancements in TinyML and lightweight deep learning [4]. Because of its potential in entertainment, education, and human-AI co-creative systems, automatic music production has drawn increasing study focus among AI-driven creative applications. Deep learning techniques may successfully acquire intricate musical patterns, such as melody, rhythm, harmony, and long-term structural dependencies, according to recent research [5,6]. Because of its interpretability and adaptability for algorithmic composition, symbolic music generation which uses discrete events like notes, durations, and temporal connections to describe musical information has emerged as a significant research area [7, 8]. However, because musical structures are hierarchical and sequential, producing high-quality

symbolic music is still difficult, requiring models that can capture both local patterns and long-range relationships [8,9].

Transformer-based architectures, which offer efficient sequence modeling capabilities, have greatly increased the creation of symbolic music. However, the majority of Transformer-based music production techniques now in use need on massive architectures with significant memory and computational requirements, which restricts their use in edge-oriented settings. Due to costly self-attention procedures and growing inference latency for lengthy input sequences, these models' application on resource-constrained platforms is still difficult, despite their remarkable symbolic music characteristics [9]. Thus, a significant open research topic is creating lightweight generative models that preserve musical coherence while lowering computational overhead.

Transformer-based models can function on low-power devices by optimizing inference methodologies and lowering model complexity, as demonstrated by recent advancements in effective Transformer deployment. For instance, the viability of implementing compact attention-based models in limited hardware environments has been shown via lightweight Transformer optimization techniques [10]. The particular difficulties of effective symbolic music production, such as the trade-off between context length, computing cost, and musical structure preservation, have not yet received enough attention. Instead, the majority of current research focuses on natural language or general AI applications.

The assessment of music produced by AI is another significant difficulty. In contrast to traditional prediction challenges, evaluating generative music models necessitates taking into account both musical qualities and computational performance. According to recent surveys, rather than depending solely on subjective assessment or generation examples, a thorough evaluation should incorporate quantifiable data relating to musical structure, diversity, and consistency [11]. Consequently, a successful edge-oriented music generating system should maintain crucial musical characteristics during effective inference while also lowering computing complexity.

This study suggests an context-aware lightweight Transformer with Context-Aware Edge Optimization for Effective Symbolic Music Generation in order to overcome these difficulties. In order to simulate symbolic music sequences with less computing complexity, the suggested framework presents a compact Transformer architecture based on lightweight self-attention and compact feed-forward blocks. The suggested architecture can learn intricate musical connections with just 2.44 million trainable parameters, in contrast to large-scale Transformer models. Additionally, the relationship between sequence context length, inference latency, and generated music characteristics is examined using a context-aware inference optimization technique. According to experimental findings, reducing the context length from 512 to 256 tokens decreases inference latency from 33.991 ms to 14.395 ms, while maintaining significant musical characteristics. This effectively strikes a compromise between efficiency and symbolic music characteristics.

The following is a summary of this paper's primary contributions:

- For the creation of symbolic music, an context-aware lightweight Transformer architecture is suggested. It combines compact feed-forward and lightweight self-attention components to accomplish effective sequence modeling with a small parameter size.
- In order to enable effective deployment in resource-constrained situations, a context-aware edge inference optimization technique is presented to examine the effects of various context lengths on computing efficiency and musical generation characteristics.
- Together, computational efficiency and symbolic music attributes such as latency, model complexity, pitch distribution, length patterns, diversity, and polyphonic behavior are analyzed to create a thorough evaluation framework.

This is how the rest of the paper is structured. The relevant research on Transformer-based architectures, edge-oriented intelligent systems, and AI-based symbolic music production is reviewed in Section 2. The suggested EdgeMusicTransformer framework, which includes the context-aware inference technique and the compact Transformer architecture, is presented in Section 3. The experimental setup, dataset, implementation specifics, and assessment procedures are all covered in Section 4. The experimental results, including computational efficiency analysis, context-length evaluation, ablation investigations, and

comparative comments, are presented in Section 5. The limits and future research directions are finally covered in Section 6, and the conclusion is then given in Section 7.

2. Related Work

Automatic music generating systems are now much more capable thanks to recent developments in deep learning, especially with the use of Transformer-based architectures. Transformers are now one of the most popular methods for creating symbolic music because of their capacity to represent long-range linkages and sequential dependencies. However, current approaches mostly concentrate on enhancing structural modeling and symbolic music characteristics, while computational efficiency and edge-oriented deployment are still largely unexplored.

The efficiency of attention processes for capturing musical structures was shown by early Transformer-based methods. The Pop Music Transformer, developed by Huang and Yang, used beat-based modeling to enhance expressive piano composition production and rhythmic coherence [12]. While this method produced encouraging outcomes, its design prioritized generating quality over computational efficiency. In a similar vein, Transformer-GAN models improved the diversity of symbolic music creation by combining Transformer topologies with adversarial learning [13]. However, the extra generating elements make the model more complex, which makes deployment difficult on platforms with limited resources. Several studies have suggested hierarchical or organized Transformer topologies as a solution to the problem of simulating long-term musical structures. In order to simultaneously model several musical properties and produce entire songs, the Compound Word Transformer presented a representation based on dynamic directed hypergraphs [14]. Richer instrumental interactions are made possible by Jin et al.'s Transformer generative adversarial network for multi-track music production, while the intricacy of multi-track modeling necessitates more computer resources [15]. By adding both fine-grained and coarse-grained attention techniques for music creation, Museformer further enhanced long-range dependency modeling [16]. These techniques greatly enhance musical coherence, but their bigger attention mechanisms and hierarchical structures make inference more difficult, which restricts their applicability in edge situations. Learning more complex musical structures has been the subject of several research. In order to capture long-term dependencies and enhance structural consistency in created compositions, Qin et al. suggested the Bar Transformer, which models music at the bar level [17]. To create music with deeper thematic connections, Theme Transformer used theme conditioning techniques [18]. Additional semantic controls have been investigated in more modern hierarchical and emotion-aware generation techniques. To enhance the production of emotive music, Ji and Yang created EmoMusicTV, which combines hierarchical Transformer-based generation with emotion conditioning [19]. Although these techniques improve creativity and controllability, they typically add more modules and conditioning mechanisms, which raises the computing demands.

Table 1 summarizes representative Transformer-based symbolic music generation methods and their attributes. It has been noted that while lightweight architectures and adaptive inference optimization for edge-oriented deployment have received less attention, current approaches primarily optimize musical representation, hierarchical modeling, or generation variety.

Table 1. Comparison of Representative Symbolic Music Generation Approaches

Reference	Method	Main Contribution	Representation / Strategy	Computational Efficiency Consideration	Limitation
[12]	Pop Music Transformer	Beat-based Transformer for expressive piano generation	Beat-level symbolic modeling	Not designed for edge deployment	High dependency on Transformer computation
[14]	Compound Word Transformer	Full-song generation using compound representation	Dynamic directed hypergraph representation	No lightweight optimization	Large-scale sequence modeling cost

[16]	Museformer	Long-term structure modeling using multi-scale attention	Fine/coarse-grained attention	Attention complexity remains high	Limited resource-constrained deployment
[17]	Bar Transformer	Hierarchical modeling of musical bars	Bar-level sequence generation	No inference optimization strategy	Increased hierarchical complexity
[18]	Theme Transformer	Theme-conditioned symbolic music generation	Theme-aware Transformer	Additional conditioning overhead	Requires extra modeling components
[19]	EmoMusicTV	Emotion-controlled hierarchical generation	Emotion-conditioned Transformer VAE	Focus on controllability rather than efficiency	Higher model complexity
This work	**context-aware lightweight Transformer**	**Efficient symbolic music generation with edge-aware optimization**	**Lightweight attention + adaptive FFN + context-aware inference**	**2.44M parameters; 7.809–33.991 ms latency range; maximum throughput of 17,783.53 tokens/s**	**Trade-off between context length and symbolic music characteristics**

In addition to algorithmic music production, human-AI cooperation in creative endeavors has also been studied recently. Research has demonstrated that by offering flexible recommendations and supporting human composers throughout the creative process, AI-based music systems can facilitate co-creative workflows [21,22]. Additionally, the significance of creating AI music systems that strike a balance between automation and user involvement is highlighted by recent studies on creative control and user interaction [23,24]. However, rather than computational optimization of generative models, these research primarily concentrate on characteristics of interaction and creativity. There are still three fundamental issues with current research, despite tremendous advancements. First, deployment constraints are ignored in favor of symbolic music characteristics in the majority of Transformer-based music generation models. Second, most current methods rely on predefined sequence lengths, which adds needless computational complexity to the inference process. Third, little research has been done on the connection between inference latency, context length reduction, and musical characteristic preservation.

This paper presents an context-aware lightweight Transformer architecture in conjunction with a context-aware edge optimization technique to overcome these constraints. The suggested strategy focuses on striking an efficient balance between computational efficiency and musical generating capabilities, in contrast to earlier methods that mainly improve model capacity. The experimental findings show that the suggested framework can significantly lower inference latency while preserving significant symbolic music characteristics, making it appropriate for edge-oriented AI music creation applications with limited resources.

3. Proposed context-aware lightweight Transformer Framework

By simultaneously maximizing model compactness and inference efficiency, the suggested framework seeks to overcome the computational difficulties of Transformer-based symbolic music production in resource-constrained edge situations. This work focuses on creating an effective architecture that can preserve important musical structures while lowering computational overhead, in contrast to traditional music generation techniques that mainly increase generation capability by increasing model depth, attention complexity, or sequence length. The four primary parts of the suggested framework are MIDI reconstruction, context-aware edge inference optimization, adaptable lightweight Transformer architecture, and symbolic music representation. To facilitate efficient sequence modeling, MIDI sequences are first

transformed into discrete symbolic tokens. The suggested lightweight Transformer then uses effective self-attention and compact feed-forward modules to learn temporal and structural connections. In order to effectively balance inference latency and musical features, a context-aware optimization technique examines various context lengths. With a compact model architecture appropriate for edge-oriented AI applications, the overall design enables effective symbolic music production.

3.1 Overall Framework

Figure 1 shows the general workflow of the suggested framework. The system starts with the MAESTRO symbolic music dataset, which uses symbolic representation to interpret MIDI files and convert them into discrete event sequences. Note events, timing relationships, velocity values, and length patterns are among the melodic details found in every MIDI sequence. The generation task can be expressed as an autoregressive sequence prediction problem by converting these events into a token-based representation. Following tokenization, the produced token sequences are sent into the suggested context-aware lightweight Transformer. To capture both short-term musical patterns and long-range interdependence, the model uses a compact Transformer structure made up of lightweight self-attention and compact feed-forward modules. The suggested architecture lowers computing complexity by utilizing a small number of Transformer blocks while retaining efficient sequence modeling capabilities, in contrast to traditional Transformer-based music generation models that rely on vast parameter spaces to improve representation capability.

The suggested context-aware inference technique optimizes the generated sequence during the inference stage. Different context configurations are examined to ascertain their impact on computing efficiency and musical features, rather than employing a fixed context length for all generation circumstances. In particular, context lengths of 128, 256, 384, and 512 tokens are assessed. Pitch distribution, duration characteristics, note diversity, and polyphonic behavior are among the symbolic music qualities that are simultaneously analyzed during the optimization process, with inference latency serving as the main computational indicator. The context-aware optimization technique makes it possible to choose a suitable context configuration depending on the intended balance between generation efficiency and melodic consistency, as shown in Figure 4.

The final symbolic music output is created by using the reconstruction module to transform the predicted token sequence back into MIDI format. In order to determine whether the lightweight design and context optimization preserve the fundamental qualities of the generated compositions, the resulting MIDI files are then examined using objective musical metrics. The suggested system offers an effective way to deploy AI-based symbolic music generation models in edge-oriented environments while lowering inference overhead through this integrated pipeline.

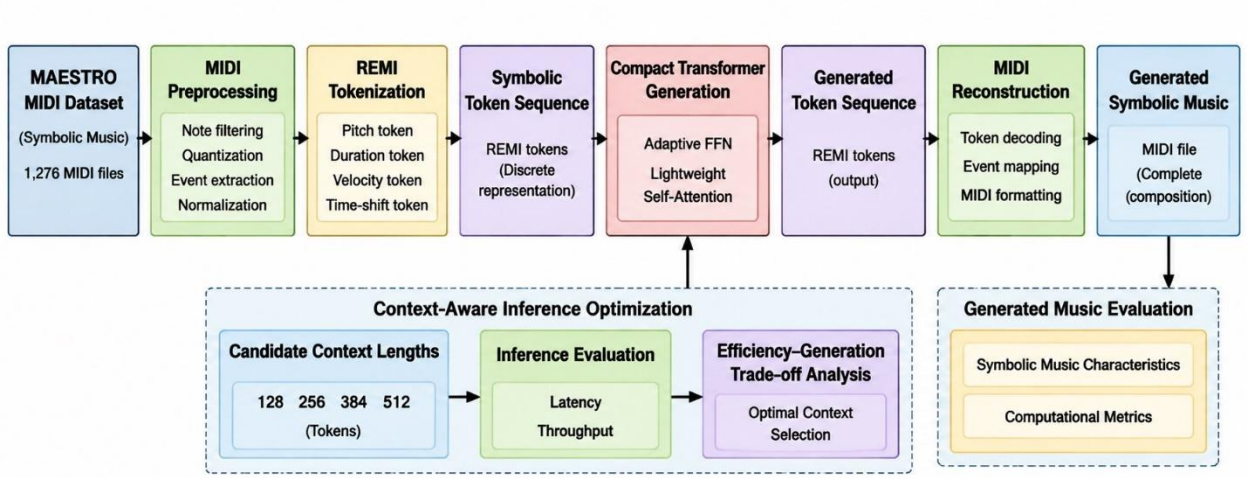

Figure 1. Overall workflow of the proposed framework, including MIDI preprocessing, symbolic token representation, context-aware lightweight Transformer-based generation, context-aware inference optimization, and MIDI reconstruction.

3.2 Symbolic Music Representation and REMI-based Tokenization

In order for deep learning models to successfully learn musical structures, symbolic music representation is essential. Transformer-based models can directly capture temporal dependencies and structural links among musical elements since symbolic representations characterize music as a sequence of discrete events, in contrast to audio-based representations that necessitate processing high-dimensional continuous signals. MIDI files are used as the main symbolic representation in this work because they offer a compact representation appropriate for effective sequence modeling while preserving crucial historical information such as note pitch, length, velocity, and temporal position. An ordered series of musical occurrences can be used to depict a MIDI composition:

$$M = \{m_1, m_2, \ldots, m_N\} \tag{1}$$

where M stands for the original MIDI sequence, N for the total number of musical events, and m_i for the pitch, velocity, duration, and temporal properties of each event. Even though MIDI offers a structured representation, it is difficult to directly apply raw MIDI events to Transformer topologies due to the varied durations of event sequences and the lack of clear organization of the temporal links between musical components for sequential learning.

The suggested system uses a REMI-based tokenization technique to convert MIDI signals into a discrete token sequence in order to overcome this restriction. While maintaining both temporal and musical links, REMI representation makes it possible to break down musical occurrences into structured tokens. The original MIDI data is first retrieved and broken down into basic elements, such as note events, velocity values, duration information, and temporal positions, as shown in Figure 2. The suggested context-aware lightweight Transformer can then process these components after they have been transformed into matching symbolic tokens. The tokenization procedure can be expressed as follows:

$$T = \Phi(M) = \{t_1, t_2, \ldots, t_L\} \tag{2}$$

where $\Phi(.)$ indicates the REMI encoding function, T is the generated token sequence, and L is the token sequence's length. A particular musical event or attribute, like Note-On, Duration, Velocity, Position, or Time-Shift, is associated with each token t_i. The model can differentiate between various musical elements and understand intricate relationships between notes, rhythms, and temporal structures thanks to this event-based representation.

The training sequences in the suggested framework are created by applying the REMI tokenizer to the MAESTRO v3 MIDI dataset. The generated vocabulary has 397 distinct musical tokens after preprocessing and token extraction. This little vocabulary helps the suggested Transformer architecture achieve its lightweight design goal by lowering the embedding layer's complexity. Additionally, the maximum sequence length is set to 512 tokens, which maintains computational efficiency during training and inference while offering enough contextual information for modeling musical connections. The context-aware lightweight Transformer's embedding layer receives the resulting token sequences and maps each discrete token into a continuous latent representation. Through effective attention-based processing, this transformation allows the model to acquire long-range relationships and high-level musical patterns. The suggested representation strategy focuses on striking an effective balance between expressive capability and computational efficiency, which is crucial for edge-oriented symbolic music generation, in contrast to traditional Transformer-based music generation approaches that frequently rely on large vocabularies and extended sequences to improve symbolic music characteristics.

Therefore, by offering a structured, compact, and Transformer-compatible encoding of symbolic music, the REMI-based representation plays a significant role in the suggested framework. The context-aware inference optimization strategy presented later in Section 3.4, where various context lengths are assessed to achieve an ideal trade-off between symbolic music characteristics and edge inference latency, is supported by this representation in addition to facilitating efficient sequence learning.

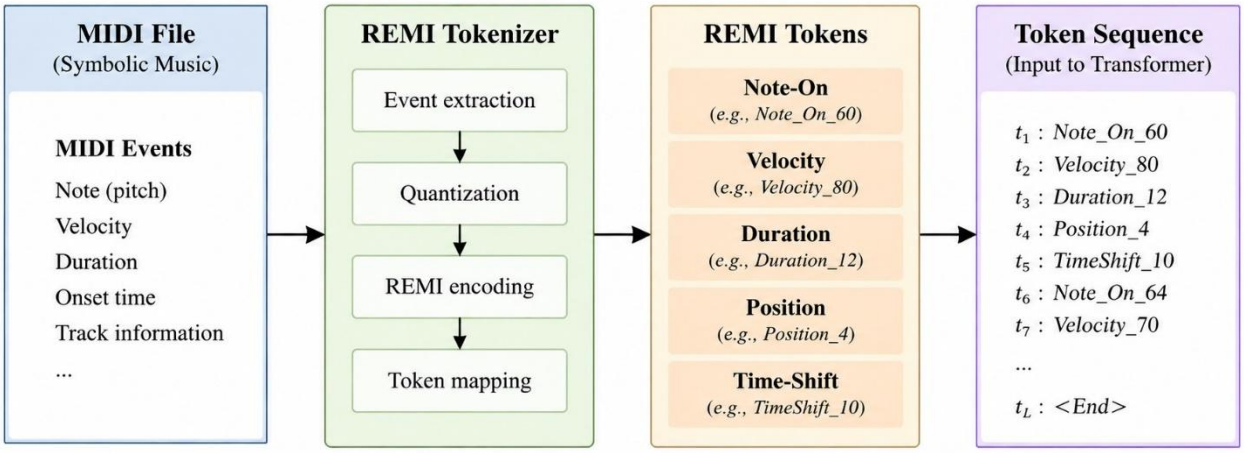

Figure 2. REMI-based symbolic music tokenization process, converting MIDI events into structured token sequences containing note, velocity, duration, and temporal information for context-aware lightweight Transformer learning.

3.3 Context-aware lightweight Transformer Architecture

The core generating module of the framework is the suggested context-aware lightweight Transformer. This architecture's primary goal is to offer an effective sequence modeling mechanism that can understand intricate symbolic music connections while keeping a small computational footprint appropriate for edge-oriented deployment. The suggested architecture adheres to a lightweight design philosophy by minimizing needless computational overhead while maintaining the crucial temporal and structural features of musical sequences, in contrast to traditional Transformer-based music generation techniques that primarily enhance performance by increasing model depth, attention complexity, or hidden dimensions. Figure 3 shows the internal architecture of the suggested model. Token embedding, learnable positional encoding, lightweight self-attention modules, Compact Feed-Forward Networks, and output token prediction are the five primary components of the compact Transformer architecture that the framework uses to process the REMI-based token sequence produced by the symbolic representation module. Four stacked Transformer blocks make up the entire architecture. Each block uses effective attention-based feature extraction and nonlinear transformation to gradually acquire high-level musical representations.

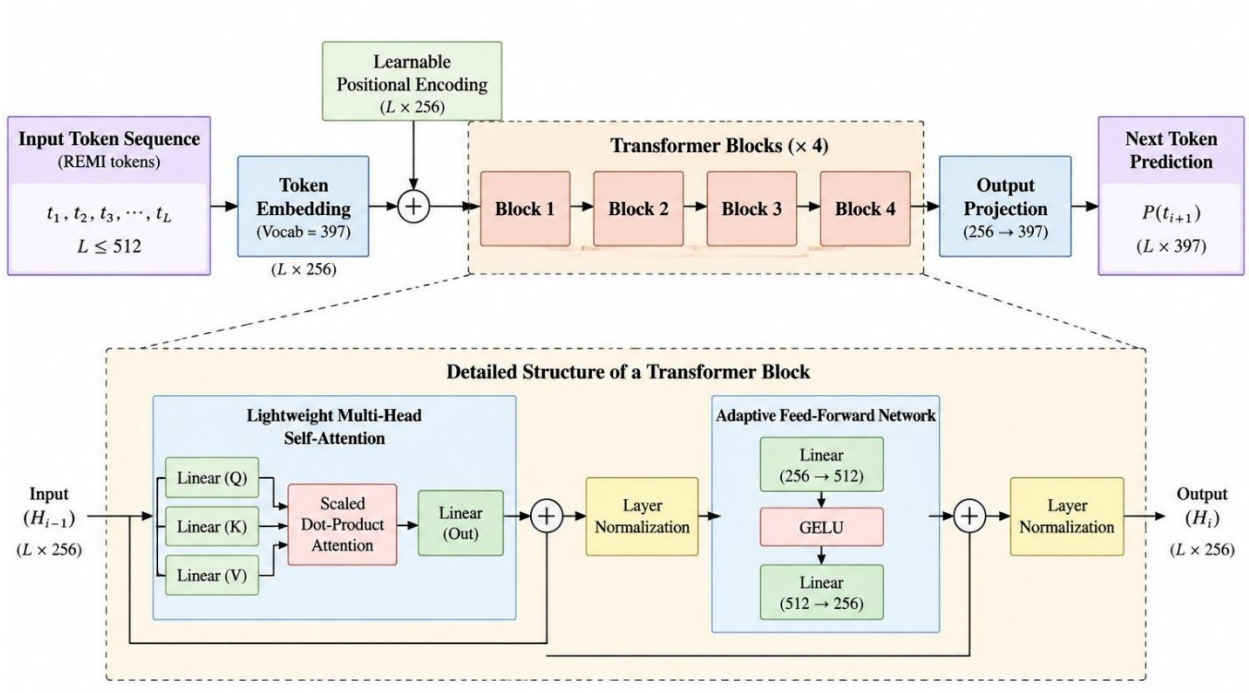

Figure 3. Architecture of the proposed context-aware lightweight Transformer, including token embedding, learnable positional encoding, lightweight self-attention modules, Compact Feed-Forward Networks, Transformer blocks, and output token prediction.

The sequence of input tokens can be shown as:

$$T = \{t_1, t_2, \ldots, t_L\} \tag{3}$$

where T stands for the REMI token sequence and L for the length of the sequence. A particular musical event, such as note information, length, velocity, or temporal position, is associated with each token. Learning the conditional probability distribution of the upcoming musical event is the Transformer's goal:

$$P(t_i | t_1, t_2, \ldots, t_{i-1}) \tag{4}$$

This permits the creation of autoregressive symbolic music.

The embedding layer first transforms the token sequence into dense latent vectors, as illustrated in Figure 3. After that, the representation is propagated through the lightweight Transformer blocks and enhanced with positional information. Lastly, the output projection layer creates the probability distribution of potential musical tokens by mapping the learnt representation into the vocabulary space.

3.3.1 Token Embedding and Learnable Positional Encoding

Maintaining the sequential sequence of tokens is crucial since symbolic music creation significantly depends on temporal linkages among musical events. Positional information must be deliberately included in the input representation since transformer designs do not naturally store sequential information. A trainable embedding layer in the suggested architecture transforms each input token into a continuous vector representation:

$$X_e = E(T) \tag{5}$$

where X_e indicates the input sequence's embedded representation and $E(.)$ represents the embedding procedure. A learnable positional embedding matrix is introduced in order to maintain the temporal ordering of musical events:

$$X_0 = X_e + P \tag{6}$$

where the trainable positional embedding matrix is denoted by P. The highest supported sequence length in the developed model is 512 tokens, which enables the model to capture adequate musical context without incurring undue processing costs. For symbolic music modeling, a compact latent representation is provided by setting the embedding dimension to 256. This design decision helps to preserve adequate representation capability for learning musical structures while lowering the total number of trainable parameters.

3.3.2 Lightweight Self-Attention Module

Relationships between various positions within a musical sequence are captured by self-attention. This feature is crucial for symbolic music generation since both local transitions and long-range interdependence are frequently necessary for musical coherence. However, typical self-attention's computational complexity grows quadratically with sequence length, making edge deployment difficult. The suggested approach uses a lightweight multi-head self-attention module with a constrained attention configuration to solve this problem. The query, key, and value matrices are computed as follows given the input representation X:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V \tag{7}$$

Next, the attention operation is calculated as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{8}$$

where the key vector dimension is represented by d_k. The outputs of each attention head are concatenated for multi-head attention:

$$MHA(X) = Concat(head_1, head_2, \ldots, head_h)W_O \tag{9}$$

where the number of attention heads is represented by h. Four attention heads with an embedding dimension of 256 are used in the suggested model. Compared to large-scale Transformer designs, this compact structure minimizes memory needs and computing operations while maintaining the capacity to describe dependencies between musical occurrences.

3.3.3 Compact Feed-Forward Network

Each Transformer block has a Compact Feed-Forward Network to carry out nonlinear transformation and feature refining after the attention process. Two fully connected layers with a GELU activation function make up the suggested feed-forward module. The following is the mathematical formulation:

$$FFN(X) = W_2\big(GELU(W_1X + b_1)\big) + b_2 \tag{10}$$

where b_1 and b_2 are bias terms and W_1 and W_2 are learnable weight matrices. The input and output dimensions of the feed-forward layer stay at 256, while the hidden dimension is set to 512. With this setup, the model's capacity for feature transformation can be increased without causing undue parameter growth. As a result, the compact feed-forward design strikes a compromise between computing economy and expressive capacity.

3.3.4 *Lightweight Transformer Block Design*

As shown in Figure 3, the suggested model is made up of four identical Transformer blocks. Every block adheres to a residual learning framework that integrates compact feed-forward transformation, layer normalization, and self-attention. Each block begins with the application of the residual connection attention mechanism:

$$Z_i = LN\big(H_{i-1} + Attention(H_{i-1})\big) \tag{11}$$

where H_{i-1} is the output of the previous block and $LN(.)$ indicates layer normalization. Nonlinear feature transformation is carried out in the second stage:

$$H_i = LN\big(Z_i + FFN(Z_i)\big) \tag{12}$$

And the output representation of the i-th Transformer block is represented by H_i. The model may gradually learn musical patterns while retaining consistent optimization and lower computing complexity thanks to this lightweight block design.

3.3.5 *Output Projection and Token Generation*

To forecast the next musical token, the final hidden representation is projected into the vocabulary space after going through the stacked Transformer blocks. The formulation of the output transformation is:

$$Y = WH_N + b \tag{13}$$

where the output projection layer's parameters are denoted by W and b. The softmax function is used to obtain the probability distribution over the entire token vocabulary:

$$P(t_i) = softmax(Y) \tag{14}$$

The final symbolic music output is created by reconstructing the generated token sequence into MIDI format.

3.3.6 *Architectural Configuration and Lightweight Design Analysis*

Table 2 summarizes the primary configuration parameters of the suggested context-aware lightweight Transformer. In order to accomplish effective inference while maintaining the capacity to acquire significant musical structures, the architecture is purposefully built with a small number of layers, attention heads, and hidden dimensions.

Table 2. Configuration of the proposed context-aware lightweight Transformer architecture

Component	**Configuration**
Vocabulary size	397
Maximum sequence length	512
Embedding dimension	256
Number of Transformer blocks	4
Number of attention heads	4
Feed-forward hidden dimension	512
Activation function	GELU
Positional encoding	Learnable embedding
Total trainable parameters	2,443,149

Table 2 shows that the suggested architecture is compact in comparison to traditional Transformer-based generation models, with just 2.44 million trainable parameters. The suggested model is more appropriate for edge-oriented symbolic music production because of its lightweight configuration, which lowers memory needs and computational overhead. Additionally, the compact architecture serves as the basis for the context-aware inference optimization technique covered in the section that follows, where various context durations are examined in order to successfully strike a compromise between musical qualities and generation efficiency.

3.4 Context-Aware Edge Inference Optimization

The quantity of contextual information taken into account during sequence production still has an impact on the inference efficiency of symbolic music synthesis, even though the suggested context-aware lightweight Transformer lowers the computing load through a compact architecture. The number of prior musical events that can be used to forecast the next token in transformer-based architectures is determined by the context length. Longer contexts allow the model to capture more broad temporal dependencies and structural links, but they also need more memory and computing complexity, which can be a major drawback in edge-oriented systems. The self-attention mechanism, in which the computational complexity grows quadratically with respect to the length of the input sequence, is the primary cause. The attention operation's complexity can be stated as follows:

$$O(L^2) \tag{15}$$

where the sequence length is denoted by L. As a result, choosing the right context length is crucial to striking a successful compromise between inference efficiency and musical dependency modeling capacity. The suggested approach presents a context-aware inference optimization strategy that assesses various context configurations throughout the generation process in order to overcome this difficulty. The suggested method takes into account a range of alternative context sizes and examines their impact on both computing efficiency and symbolic music qualities, as opposed to using a single context length for all generating circumstances. The definition of the candidate context set is:

$$C = \{c_1, c_2, \ldots, c_n\} \tag{16}$$

where the available context configurations are represented by C. The proposed context lengths in this study are 128, 256, 384, and 512 tokens.

Figure 4 shows the general process of the suggested context-aware optimization technique. First, the input is the REMI-based token sequence that was acquired from the symbolic representation module. The context-aware lightweight Transformer uses the corresponding historical token information to perform autoregressive inference for each candidate context length in order to forecast future musical events. For additional examination, the resultant token sequence is subsequently rebuilt in MIDI format. There are two complementing goals in the evaluation process. The first goal takes into account memory constraints, generating throughput, inference latency, and computing efficiency. Given the restricted computational resources available for edge deployment, these characteristics are especially crucial. By examining structural aspects including pitch distribution, note diversity, length patterns, and polyphonic behavior, the second goal aims to maintain the fundamental elements of symbolic music generation. One way to formulate the context selection problem is as a multi-objective optimization problem:

$$C^* = argmin_{c \in C}\big(\alpha E(c) - \beta M(c)\big) \tag{17}$$

where $E(c)$ is the computational cost related to context c, $M(c)$ is the extracted musical qualities, and C^* is the chosen context duration. The relative relevance of computational efficiency and preservation of symbolic music characteristics is determined by the parameters α and β. The suggested approach does not

separately seek to minimize computational costs or maximize context length. Rather, it offers an adaptive technique to choose an appropriate operating point based on the target edge environment's requirements. The suggested symbolic music creation framework can be used flexibly under various computational situations thanks to this design. The context-aware optimization module is incorporated between the final MIDI reconstruction phase and the context-aware lightweight Transformer inference stage, as illustrated in Figure 4. Section 5 presents the experimental assessment of various context configurations and their effects on generated music qualities and inference efficiency.

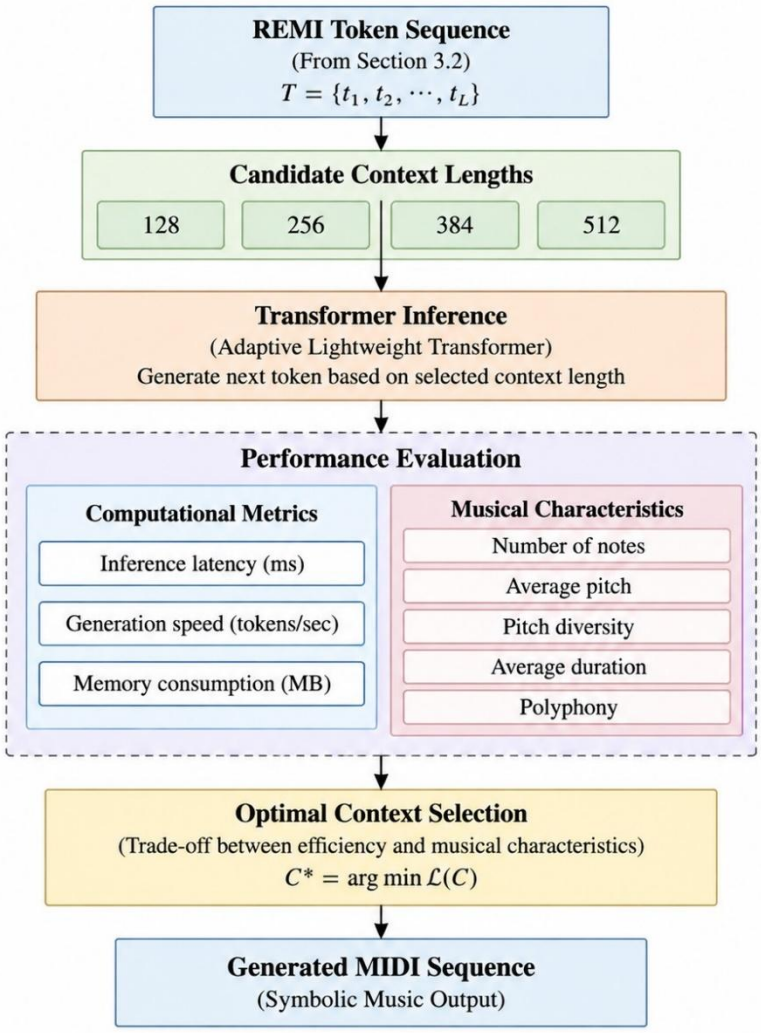

Figure 4. Context-aware inference optimization strategy for edge-oriented symbolic music generation.

4. Experimental Setup

The experimental approach utilized to assess the efficacy of the suggested context-aware lightweight Transformer framework for symbolic music production is presented in this section. The assessment is intended to look into the suggested architecture's computational effectiveness as well as its capacity to maintain crucial musical qualities during generation. The compilation of the dataset, implementation specifics, assessment criteria, and the development of several evaluation situations comprise the experimental setup. Table 3 summarizes the proposed framework's full setup.

Table 3. Experimental configuration and implementation details of the proposed context-aware lightweight Transformer framework.

Category	Parameter	Configuration
Dataset	Dataset name	MAESTRO v3.0.0
Dataset	Data type	Symbolic MIDI files
Dataset	Number of MIDI files	1276
Representation	Encoding method	REMI-based tokenization
Representation	Vocabulary size	397
Representation	Maximum sequence length	512 tokens
Model Architecture	Transformer type	context-aware lightweight Transformer
Model Architecture	Number of Transformer blocks	4
Model Architecture	Attention mechanism	Multi-head self-attention
Model Architecture	Number of attention heads	4
Model Architecture	Embedding dimension	256
Model Architecture	Feed-forward hidden dimension	512
Model Architecture	Activation function	GELU
Model Architecture	Positional encoding	Learnable positional embedding
Model Complexity	Trainable parameters	2,443,149
Implementation	Programming language	Python 3.10
Implementation	Deep learning framework	PyTorch
Implementation	MIDI processing libraries	symusic, music21
Evaluation	Context lengths	128, 256, 384, 512
Evaluation	Generated samples	Representative generated MIDI sequence
Evaluation	Real-seed evaluation	MAESTRO MIDI seed sequence

4.1 Dataset Description and Preprocessing

The MAESTRO v3.0.0 dataset, a popular benchmark for symbolic piano music production, is utilized to assess the suggested system. Aligned MIDI sequences with comprehensive performance data, such as note events, velocity values, durations, and temporal relationships, make up the dataset. The input data source for training and assessment in this study is 1276 MIDI files. The MIDI files go through a preprocessing pipeline to transform unstructured musical events into a structured symbolic representation before being supplied to the Transformer model. MIDI parsing, event extraction, REMI-based tokenization, vocabulary building, and sequence segmentation make up the preprocessing process. Each MIDI composition is converted into a discrete token sequence appropriate for autoregressive Transformer learning through the tokenization procedure. 397 distinct musical tokens that indicate various musical events, such as note information, timing relationships, velocity, and duration properties, are included in the created lexicon. The maximum sequence length is set to 512 tokens in order to preserve adequate musical context while maintaining computing efficiency. The context-aware lightweight Transformer then uses the final token sequences as inputs to create new symbolic music sequences by learning the probabilistic correlations between successive musical events.

4.2 Implementation Details and Model Configuration

Python 3.10 and PyTorch, a deep learning framework, are used to implement the suggested framework. Symbolic music processing libraries, such as symusic and music21, are used for MIDI processing and analysis. Table 3 summarizes the primary architectural and implementation parameters. Four stacked transformer blocks make up the suggested context-aware lightweight Transformer. Each block has a Compact Feed-Forward Network and a lightweight multi-head self-attention module. The feed-forward hidden dimension is set to 512, and the embedding dimension is set to 256. To maintain temporal links between symbolic music tokens, the model uses learnable positional embeddings and four attention heads. A total of 2,443,149 trainable parameters are produced by the compact design. This design was chosen on purpose to minimize computational cost while preserving enough capacity to learn musical dependencies. In edge-oriented systems, such a lightweight design serves as the basis for effective inference optimization.

4.3 Evaluation Metrics

Two types of metrics are taken into consideration in order to thoroughly assess the suggested framework: symbolic music features and computational efficiency metrics.

4.3.1 Computational Efficiency Metrics

The following measures are used to assess the suggested model's computational efficiency:

Inference Latency:
the amount of time needed during inference to produce musical tokens. The model's suitability for real-time edge applications is directly reflected in this statistic.

Generation Throughput:
The quantity of tokens produced every second, computed as:

$$TPS = \frac{N_{tokens}}{T_{generation}} \tag{18}$$

where $T_{generation}$ is the entire generating time and N_{tokens} is the number of tokens generated.

Memory Variation:
the variation in memory usage during model inference, which shows how many resources various generation setups require.

4.3.2 Symbolic Music Characteristics Metrics

Since a single numerical indicator cannot adequately capture the quality of music creation, a number of objective symbolic traits are examined:

Number of Notes:
Calculates the quantity of musical stuff produced.

Pitch Distribution:
Examines the resulting sequences' average pitch, pitch range, and pitch variance.

Pitch Diversity:
Counts the number of distinct pitch values that show up in compositions that are generated.

Duration Characteristics:
Examines average note duration to assess temporal patterns.

Polyphonic Ratio:
determines the percentage of notes that occur concurrently, which reflects the harmonic complexity of the music that is produced.

4.4 Experimental Scenarios

Four experimental scenarios are taken into consideration in order to evaluate various parts of the suggested framework. To assess the architecture's lightweight features in terms of parameter count and computational configuration, model complexity analysis is first carried out. Second, four context lengths of 128, 256, 384, and 512 tokens are used for context-aware inference assessment in order to examine the connection between generated sequence properties and inference efficiency. Third, structural characteristics such as pitch

distribution, length patterns, note diversity, and polyphonic behavior are assessed using objective symbolic music metrics on a typical generated MIDI sequence. Lastly, the input seed for context-conditioned continuation evaluation under various context configurations is an actual MAESTRO MIDI sequence.

4.5 Baseline Comparison Protocol

Representative Transformer-based techniques, such as Pop Music Transformer [12], Compound Word Transformer [14], Museformer [16], and Bar Transformer [17], are taken into consideration for comparison with current symbolic music creation methodologies. These techniques beat-based representation, compound event modeling, hierarchical attention, and bar-level generation represent various methods for enhancing musical structure modeling. The comparison focuses on architectural features, representation schemes, and computational considerations because the main goal of this work is efficient edge-oriented generation rather than just maximizing generation complexity. The following section's experimental findings show how the suggested lightweight design successfully strikes a balance between inference efficiency and generation capability.

5. Results and Discussion

This section presents and analyzes the experimental findings of the suggested context-aware lightweight Transformer framework. The evaluation's main goal is to confirm the suggested method's efficacy from the standpoints of computation and music production. First, the model complexity and computational configuration of the suggested architecture are examined in order to look at its lightweight features. The real-seed production capabilities, context-aware inference performance, generated symbolic music characteristics, and the contribution of various framework components are then assessed. The obtained results shed light on the trade-off between effective symbolic music production and efficient edge-oriented deployment.

5.1 Model Complexity and Computational Efficiency Analysis

In the first experiment, the suggested EdgeMusicTransformer's computational properties and applicability for effective symbolic music creation in edge-oriented environments are assessed. The suggested framework concentrates on preserving a compact design with lower computational cost, in contrast to large-scale Transformer-based models that enhance sequence modeling capabilities by expanding model depth and parameter size. The implemented model has four Transformer blocks with four attention heads, an embedding dimension of 256, and a feed-forward hidden dimension of 512, as shown in Table 2. Only 2,443,149 trainable parameters are produced by this arrangement, indicating how lightweight the suggested design is. The model is better suited for environments with limited resources because of its compact parameter size, which lowers memory requirements and inference complexity. Additionally, this lightweight design serves as the basis for the context-aware inference analysis discussed in the sections that follow, where various context lengths are examined to successfully strike a balance between generated symbolic music features and inference efficiency.

5.2 Generated Symbolic Music Characteristics Analysis

This section examines the proposed context-aware lightweight Transformer 's capacity to produce organized symbolic music sequences after confirming its computational efficiency. This analysis's goal is to determine whether the lightweight architectural design can maintain key symbolic music characteristics including note diversity, temporal organization, pitch distribution, and polyphonic structure. Objective symbolic music descriptors are used to analyze a representative generated MIDI sequence produced by the suggested system. These descriptors enable the assessment of musical structure beyond mere token prediction accuracy and offer quantitative information about the features of the generated piece. The number of generated notes, average pitch, pitch variation, pitch range, pitch diversity, average note duration, and polyphonic ratio are among the extracted characteristics.

Table 4 summarizes the statistical properties of the developed symbolic music sequence. According to reports, the developed composition has 232 musical notes with a pitch standard deviation of 8.74 and an

average pitch of 63.64. The created sequence covers a relatively wide musical register rather than being confined around a small number of notes, as evidenced by the acquired pitch range from 37 to 91. Additionally, the created sequence has enough melodic diversity, as evidenced by the 41 distinct pitch values.

Table 4. Statistical characteristics of a representative generated symbolic music sequence.

Metric	Value
Number of notes	232
Average pitch	63.64
Pitch standard deviation	8.74
Pitch range	37–91
Unique pitches	41
Average duration	1.53
Polyphony ratio (%)	70.88

Average note duration and polyphonic behavior are also used to assess the temporal properties of the created sequence. Instead of generating consistent note patterns, the average duration value of 1.53 shows that the generated composition has a mixture of brief and prolonged musical occurrences, enabling the model to depict temporal changes. Additionally, the polyphony ratio of 70.88% shows that there are simultaneous note activations, indicating that the suggested model can produce multi-note structures that are represented in symbolic piano compositions. The piano-roll representation is shown in Fig. 5 to further examine the temporal and pitch organization of the created sequence. The piano-roll visualization offers an intuitive depiction of the link between pitch and time, in contrast to numerical statistical analysis. The resulting sequence includes many simultaneous note events, pitch distribution fluctuations, and continuous note transitions, as shown in Figure 5. These findings show that organized symbolic sequences with significant temporal organization can be produced by the suggested framework.

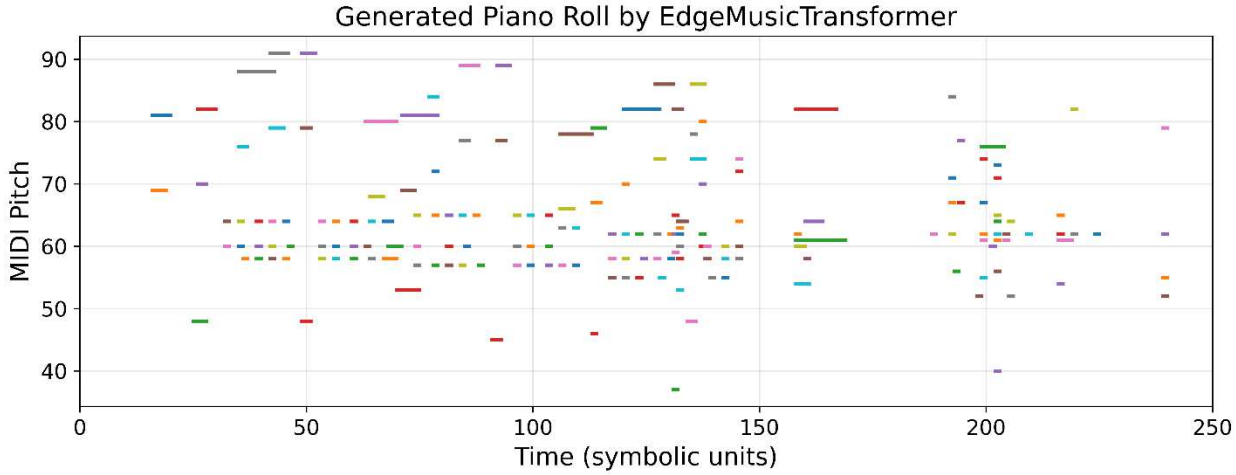

Figure 5. Piano-roll visualization of a representative generated symbolic music sequence produced by the proposed context-aware lightweight Transformer.

To give a human-interpretable depiction of the generated composition, the generated MIDI sequence is reconstructed into a standard musical score format in addition to the piano-roll representation. Figure 6 shows the reconstructed eight-bar snippet. The score representation shows that note ordering, rhythmic structure, and simultaneous musical occurrences may all be preserved when converting the resulting symbolic sequence into traditional musical notation. By showing that the output of the suggested model can be converted into a legible musical representation rather than just abstract token sequences, the score

visualization offers supporting evidence to the piano-roll analysis. For symbolic music generation systems, this feature is crucial since the produced sequences should preserve significant connections between distinct musical occurrences.

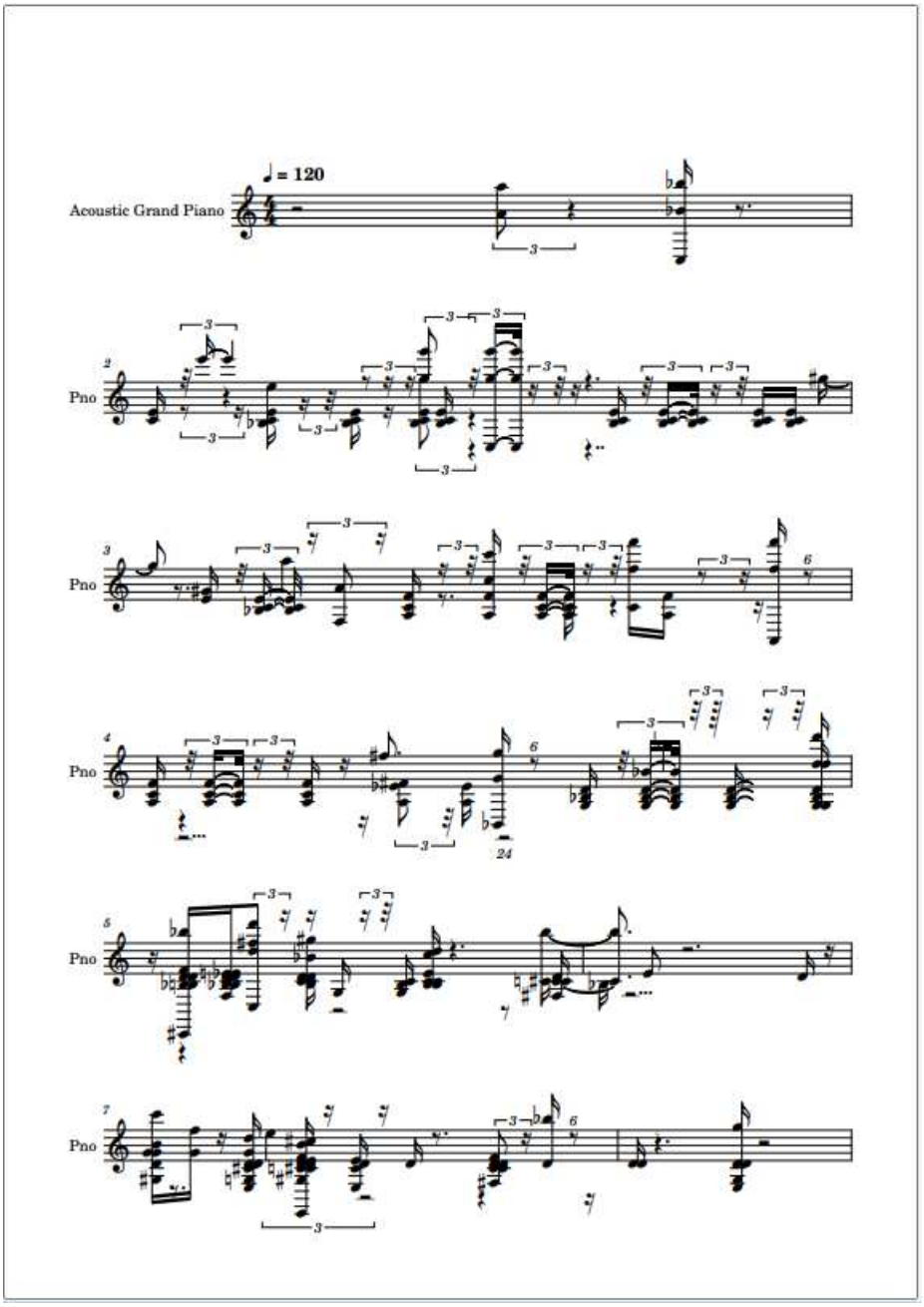

Figure 6. Eight-bar score representation reconstructed from a representative generated MIDI sequence produced by the proposed framework.

Overall, despite its small size, the suggested context-aware lightweight Transformer can produce structured symbolic music sequences, according to statistical analysis and visual observations. The results gained indicate that learning basic musical features, such as pitch fluctuation, temporal structure, and polyphonic organization, is not hindered by decreasing model complexity. These results corroborate the suggested

lightweight design's efficacy as a workable strategy for effective symbolic music production in edge-oriented AI settings.

5.3 Context Length Impact on Edge Inference Performance

Because it influences the quantity of previously generated historical information available during autoregressive inference, the context length is crucial to Transformer-based symbolic music production. The model can access more previous musical occurrences and possibly capture longer-range dependencies with a bigger context. However, lengthening the context also raises the self-attention mechanism's computing cost, which may have a detrimental effect on inference efficiency in settings with limited resources. Using the suggested context-aware lightweight Transformer, four distinct context lengths are assessed in order to examine the connection between contextual information and computing efficiency:

$$C = \{128, 256, 384, 512\}$$

The training model and inference settings remain the same across the tests, with the exception of the context length. The impact of context length on inference efficiency is assessed using three computing metrics: inference delay, generation throughput, and peak memory use. Table 5 summarizes the outcomes for each of the four context configurations.

Table 5. Effect of different context lengths on inference efficiency of the proposed context-aware lightweight Transformer.

Context Length (tokens)	Inference Latency (ms)	Throughput (tokens/s)	Peak Memory Usage (MB)
128	7.809	16,391.13	248.148
256	14.395	17,783.53	259.699
384	24.799	15,484.23	258.832
512	33.991	15,062.75	257.785

The findings demonstrate that a longer context results in a higher inference overhead. Because processing longer sequences through the Transformer self-attention mechanism requires more computing power, latency rises from 7.809 ms at 128 tokens to 33.991 ms at 512 tokens. The 256-token setup, on the other hand, attains the maximum throughput of 17,783.53 tokens/s, suggesting a good balance between computing efficiency and contextual processing. Longer contexts gradually increase inference latency and decrease throughput, whereas the 128-token setup offers the lowest latency but does not reach the maximum throughput. Across the assessed setups, peak memory utilization stays within a very small range of 248.148–259.699 MB, suggesting a rather constant memory footprint under the measured inference settings. Overall, the findings show that for the suggested symbolic music production framework, an intermediate context length specifically, 256 tokens offers an efficient computational trade-off.

Figure 7 provides additional illustration of the latency variation across various context setups. The figure shows a roughly increasing trend between inference delay and context duration. Specifically, the model processes the most historical data at each generation step in the 512-token configuration, which results in the highest computational cost.

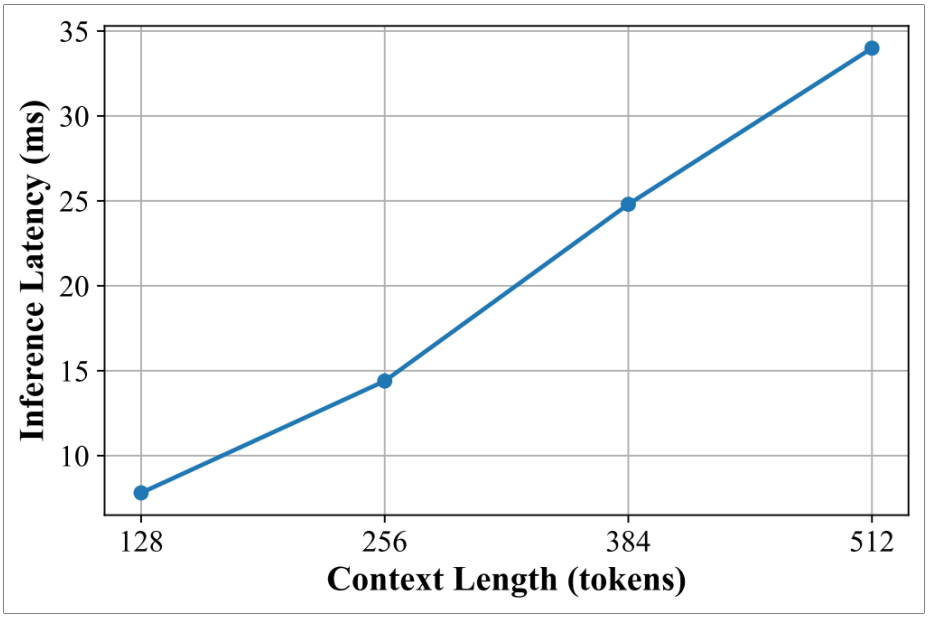

Figure 7. Impact of context length on inference latency of the proposed context-aware lightweight Transformer.

Figure 8 shows how context length affects generating throughput. The findings show that the throughput varies based on how much contextual information is given to the model. The resulting throughput numbers show that the relationship between context length and generation speed is not strictly linear because of the interplay between sequence processing and implementation overhead, even though the shortest context lowers the computing workload per inference step.

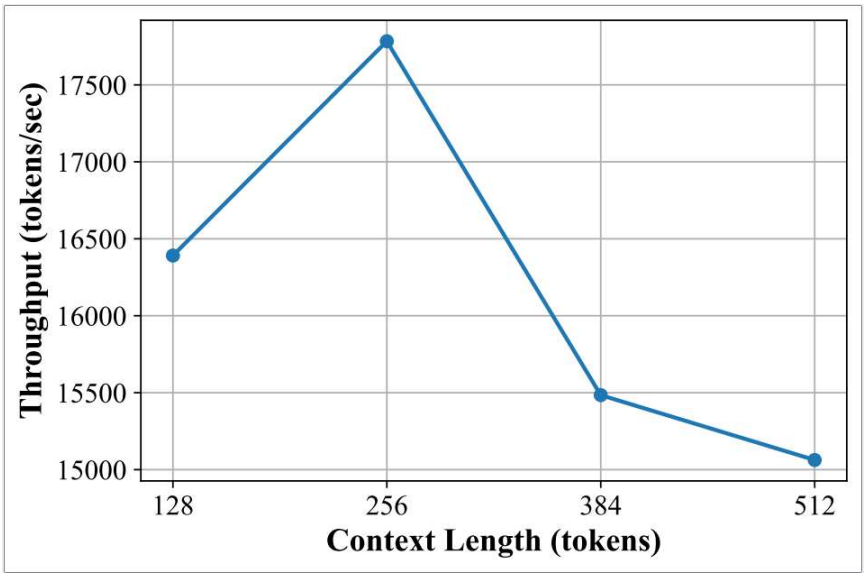

Figure 8. Impact of context length on inference throughput of the proposed Adaptive Lightweight Transformer.

Context length influences the quantity of historical information available during production in addition to computational efficiency. A longer context gives the model a more comprehensive historical representation, which can affect the produced musical patterns, such as polyphonic structure, duration distribution, and pitch transitions. Therefore, rather than focusing only on inference speed, choosing an acceptable context

length necessitates taking into account both computational restrictions and symbolic music characteristics. The logic behind the suggested context-aware inference optimization technique is validated by the obtained results. The framework allows contextual information to be flexibly adjusted in accordance with the computing requirements of the target deployment environment rather than depending on a fixed context configuration. For edge-oriented symbolic music production scenarios, where striking the right balance between inference efficiency and generation capability is crucial, this capability is especially helpful.

5.4 Real Seed Context Evaluation

The ability of the suggested context-aware lightweight Transformer to carry out context-conditioned symbolic music continuation is further examined in this section, whereas the earlier experiments examined the impact of context length on inference efficiency and generated sequences under general generation settings. Continuation-based generation necessitates that the model comprehend and maintain the features of an existing musical context, in contrast to unconditional generation, where the model creates sequences without a baseline musical structure. The initial seed sequence for this purpose is a real MIDI sequence that was chosen from the MAESTRO dataset. The trained model receives the seed MIDI file for continuation generation once it has been transformed into a REMI-based token representation. Four distinct context lengths 128, 256, 384, and 512 tokens—are assessed in order to examine the impact of historical musical knowledge.

Table 6 summarizes the statistical properties of the generated continuations under various context setups. The findings show that altering the context length has an impact on the generated sequences' structural qualities, such as the quantity of notes produced, pitch distribution, duration traits, and polyphonic behavior.

Table 6. Statistical analysis of generated continuations using different context lengths with a real MIDI seed.

Context Length	Notes	Average Pitch	Pitch Std	Unique Pitches	Average Duration	Average Velocity	Polyphony (%)
128	175	68.806	7.284	17	2.122	59.394	23.429
256	330	40.355	14.354	28	2.629	60.094	8.788
384	244	47.160	16.069	26	1.044	72.619	11.066
512	243	68.267	7.379	25	0.354	68.136	10.700

The features of the generated continuation are directly impacted by the context duration, as shown in Table 6. The 256-token setup generates the greatest number of created notes (330) and distinct pitches (28), suggesting that the model can provide a more varied sequence in terms of musical events when a considerable quantity of history information is provided. The 512-token setup, on the other hand, produces 243 notes with 25 distinct pitches, indicating that a lengthier context does not always result in greater musical diversity. This suggests that since generation behavior depends on the learnt distribution and the available historical representation, increasing contextual knowledge does not always optimize symbolic output diversity.

Differences across context configurations are also shown by the duration-related study. For the 512-token option, the average duration drops to 0.354 as the context length increases. This suggests that by giving the model a more thorough historical representation of earlier musical patterns, longer contextual inputs may affect the temporal density of generated events. The piano-roll representations of the four context configurations are shown in Figure 9 to offer a qualitative examination of the generated continuations. The image shows how the temporal organization and pitch distribution of the created sequences are affected by varying levels of historical information. Instead of isolated or random token sequences, all combinations result in structured musical patterns with continuous note events, as illustrated in Figure 9. Additionally, differing context lengths exhibit differences in note density, pitch transitions, and temporal organization.

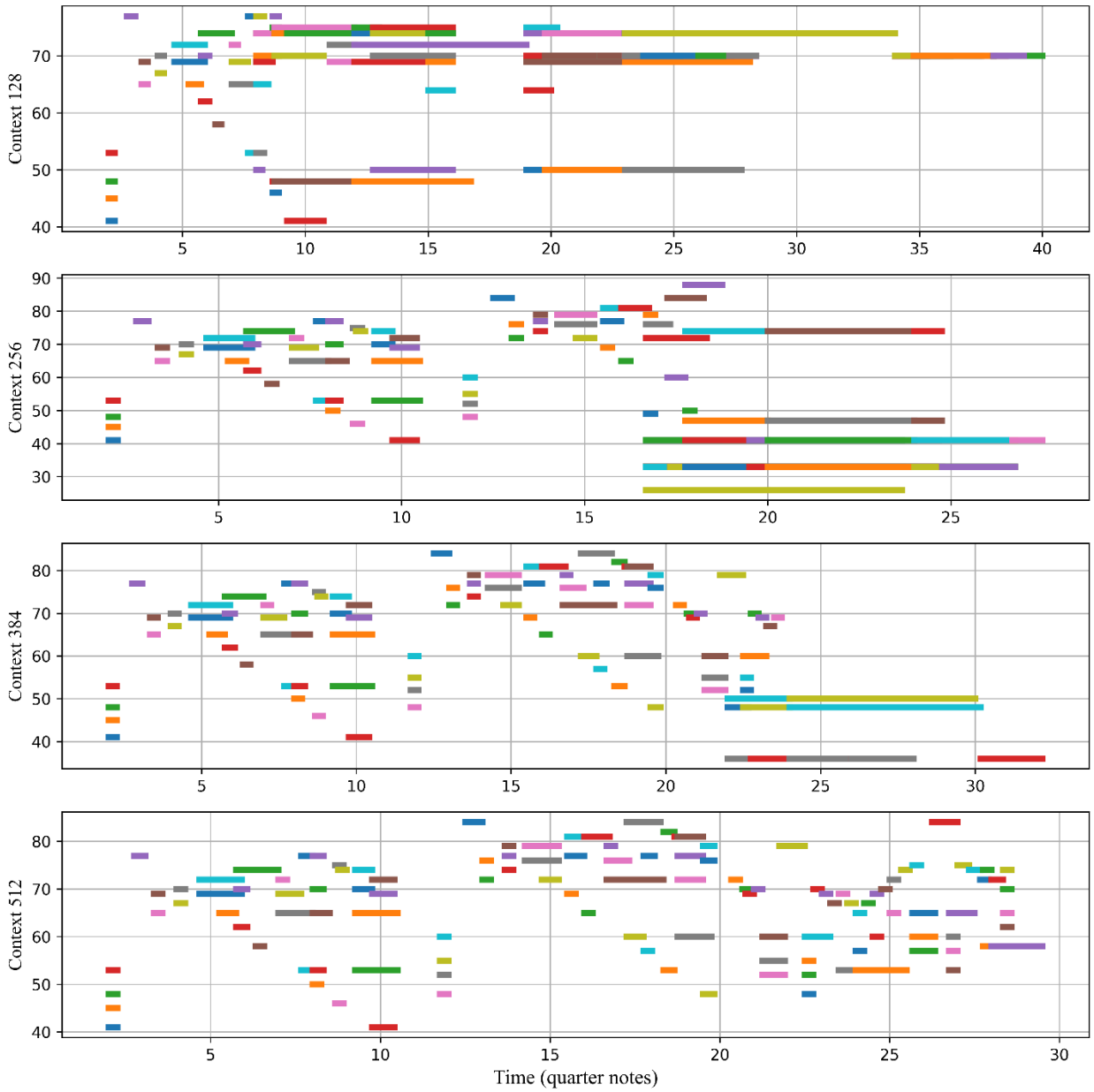

Figure 9. Piano-roll comparison of generated musical continuations using different context lengths with a real MAESTRO seed sequence.

The findings show that the suggested framework can produce logical symbolic continuations by making use of the current musical environment. Additionally, because context duration affects both computing requirements and musical output characteristics, the experiment validates that it should be regarded as a significant generation parameter. As a result, the context-aware approach presented in Section 3.4 offers a versatile method for modifying the generating process in accordance with the demands of various deployment scenarios. Overall, the real-seed evaluation shows that the suggested context-aware lightweight Transformer can execute context-conditioned continuation while preserving structured symbolic representations, in addition to autonomous music creation. This feature strengthens the suggested lightweight framework's suitability for effective edge-oriented music production applications.

5.5 Ablation Study of Transformer Components

Using the same trained model checkpoint, an inference-time ablation investigation was carried out to examine the contribution of the primary computational pathways in the suggested EdgeMusicTransformer. The goal of this research is to directly quantify the influence of individual Transformer components during inference while maintaining all learnt parameters, in contrast to traditional ablation procedures that necessitate retraining independent models after eliminating specific components. This approach offers a fair evaluation of the functional significance and computational contribution of every element in the suggested architecture. The self-attention mechanism and the feed-forward transformation module are the two primary processing pathways that underpin the Transformer design. The self-attention route is in charge of capturing contextual linkages within the input sequence and modeling dependencies among symbolic music tokens. In the meantime, nonlinear feature transformation is carried out by the feed-forward module to improve the learnt musical patterns' capacity for representation. As a result, two inference-time variations were produced by turning off each pathway separately while keeping the rest of the original model. Table 7 displays the setups that were evaluated.

Table 7. Configuration of the inference-time ablation study.

Variant	Self-Attention Pathway	Feed-Forward Pathway	Description
A0: Full Model	Enabled	Enabled	Complete EdgeMusicTransformer
A1: Without Self-Attention	Disabled	Enabled	Evaluates the contribution of contextual dependency modeling
A2: Without Feed-Forward	Enabled	Disabled	Evaluates the contribution of nonlinear feature transformation

Computational Impact Analysis

Using inference latency and throughput under the same input context length of 256 tokens, the computational impact of turning off each Transformer pathway was initially assessed. Table 8 provides a summary of the findings. Because the ablation measurements were carried out under a different controlled inference run, runtime measurement variability may be the cause of small variations from Table 5.

Table 8. Computational performance of different ablation configurations.

Variant	Latency (ms)	Tokens/sec
A0: Full Model	13.651	18,753.31
A1: Without Self-Attention	5.588	45,812.57
A2: Without Feed-Forward	6.966	36,749.19

The whole EdgeMusicTransformer achieves an inference latency of 13.651 ms and a throughput of 18,753.31 tokens/s, as shown in Table 8. By turning off the self-attention channel, the throughput rises to 45,812.57 tokens/s and the latency drops to 5.588 ms. In a similar vein, turning off the feed-forward method increases throughput to 36,749.19 tokens/s while reducing latency to 6.966 ms. These findings suggest that in the assessed design, the self-attention pathway adds more computational overhead than the feed-forward pathway. This finding aligns with the computational features of Transformer-based models, where sequence-level dependency modeling typically necessitates significant computing for attention operations. However, eliminating these elements results in a lower computing cost at the risk of changing the features of the resulting symbolic music. To determine how each element affects the produced MIDI sequences, a further analysis was carried out.

Impact on Generated Symbolic Music Characteristics

Each ablation setup used the same REMI-based tokenization and decoding pipeline to produce a MIDI sequence for this analysis. Note density, pitch distribution, temporal features, velocity, and polyphonic

structure were among the structural and statistical music characteristics that were used to examine the generated MIDI files. Table 9 displays the results that were obtained.

Table 9. Effect of Transformer components on generated symbolic music characteristics.

Variant	Notes	Average Pitch	Pitch Std	Unique Pitches	Average Duration	Average Velocity	Polyphony (%)
A0: Full Model	93	53.70	15.85	33	0.314	84.16	7.97
A1: Without Self-Attention	83	60.02	17.17	38	0.447	82.37	10.23
A2: Without Feed-Forward	60	64.03	13.02	29	0.488	82.47	5.03

Table 9 shows that the whole model produces the greatest number of notes (93) with a polyphony ratio of 7.97% and a balanced pitch distribution with 33 distinct pitches. The number of notes produced drops to 83 and the pitch standard deviation rises from 15.85 to 17.17 when the self-attention pathway is turned off (A1). The greater pitch variation indicates that the model's ability to maintain contextual links among created musical events is impacted by the removal of attention, even if this configuration generates a little higher number of distinct pitches (38). In a similar vein, turning off the feed-forward pathway (A2) reduces pitch diversity to 29 distinct pitches and further reduces the number of notes created (60). Additionally, the average duration rises from 0.314 in the whole model to 0.488, suggesting that the generated sequences have less temporal variety. Additionally, the reduced polyphony ratio (5.03%) indicates that the removal of the feed-forward pathway is associated with changes in structural characteristics of the generated sequences.

Overall, the ablation results show that both Transformer pathways help strike a balance between the ability to represent symbolic music and computational efficiency. While the feed-forward module helps with feature transformation and temporal-pitch variety, the self-attention mechanism mainly affects contextual modeling and sequence dependence representation. The full EdgeMusicTransformer validates the efficacy of the suggested lightweight Transformer design for edge-oriented symbolic music generation by offering the most balanced trade-off between inference efficiency and generated music qualities.

5.6 Comparison with Existing Music Generation Approaches

Previous research on Transformer-based music production has mostly concentrated on enhancing structural organization, multi-track generation, long-range dependency modeling, and symbolic music representation [12–19]. Transformer designs are useful for modeling intricate musical sequences, as shown by methods like Pop Music Transformer [12], Compound Word Transformer [14], Museformer [16], and Bar Transformer [17]. However, rather than focusing on inference in resource-constrained edge situations, their experimental goals primarily highlight generation capabilities and musical modeling. The suggested EdgeMusicTransformer, on the other hand, concentrates on the useful trade-off between computational efficiency and musical sequence modeling. With four Transformer blocks and four attention heads, the built model has just 2,443,149 trainable parameters. Without depending on the very larger architectures often associated with long-context music modeling, this tiny arrangement offers a foundation for effective symbolic music creation.

The outcomes of the experiment show how crucial customizable context length is for effective inference. The maximum measured throughput was 17,783.53 tokens/s at 256 tokens, while latency increased from 7.809 ms to 33.991 ms when the context length increased from 128 to 512 tokens. Additionally, the real-seed evaluation demonstrated that the statistical properties of generated continuations, such as note density, pitch diversity, duration, and polyphony, are influenced by context length. The ablation experiments offer more proof that the behavior of the entire model is influenced by both feed-forward pathways and self-attention. Latency was lowered to 5.588 ms by eliminating the self-attention pathway and 6.966 ms by eliminating the feed-forward pathway. Efficiency advantages should be taken into account in conjunction with the ensuing generation behavior, as these computational reductions were followed by modifications in

the generated symbolic qualities. Overall, EdgeMusicTransformer's edge-oriented design perspective which takes into account model compactness, customizable context length, inference efficiency, and interactive generation is what sets it apart from other music generating techniques. Since a true cross-study comparison is impossible due to variations in datasets, tokenization, hardware, and evaluation methodologies, direct numerical comparisons with earlier approaches are avoided.

5.7 Discussion

All of the experimental findings show that the suggested EdgeMusicTransformer strikes a useful balance between computing efficiency and symbolic music modeling. The suggested method uses a compact Transformer configuration and actively investigates the impact of inference context on computational cost and generated musical features instead of optimizing model capacity.

5.7.1 Edge Efficiency and Deployment Implications

With 2,443,149 trainable parameters, the model exhibits a comparatively small architecture for Transformer-based symbolic music production. The inference tests also demonstrate that context length has a significant impact on computing cost. Longer contextual processing adds a quantifiable inference burden, as evidenced by the latency increasing from 7.809 ms at 128 tokens to 33.991 ms at 512 tokens. The maximum recorded throughput of 17,783.53 tokens/s was simultaneously attained by the 256-token setup. These findings imply that, in contexts with limited resources, context length can be a useful operational parameter for striking a balance between contextual information and inference efficiency.

5.7.2 Context-Length and Generation Trade-off

The real-seed experiments show that context duration influences both the statistical characteristics of generated continuations and computational performance. Note density, pitch distribution, length, and polyphony varied depending on the circumstance. Crucially, compared to shorter contexts, the 512-token context did not consistently yield more musical diversity. This result suggests that enhancing symbolic music characteristics may not always follow from expanding the accessible context. Alternatively, an intermediate context in the current tests, 256 tokens can offer a workable balance between generated sequence features and computing cost.

5.7.3 Architectural Contribution and Limitations

The ablation study also shows that feed-forward circuits and self-attention both affect how the entire model behaves. While eliminating these circuits significantly decreased inference latency, it also altered the properties of the MIDI sequences that were produced. Disabling the feed-forward pathway decreased these values to 60 notes and 29 unique pitches, while the whole model produced 93 notes with 33 distinct pitches. These findings reinforce the need of keeping the primary Transformer pathways when taking into account both computing efficiency and generation behavior. However, the results should be viewed as component-sensitivity analysis rather than independent architectural comparisons because the ablation variations were assessed by blocking pathways during inference without retraining.

Overall, the results are consistent with EdgeMusicTransformer's primary design goal, which is to provide interactive edge-oriented symbolic music production by striking a suitable balance between model compactness, contextual modeling, generation behavior, and inference efficiency.

6. Limitations and Future Work

Despite the encouraging outcomes, there are a few things to be aware of. First, human listening-based evaluation of symbolic music characteristics and user satisfaction has not yet been carried out; instead, the current evaluation mainly concentrates on symbolic MIDI features and inference efficiency. Second, thorough deployment across many resource-constrained edge hardware platforms needs to be investigated, as the computational experiments were conducted in the current assessment setting. Third, a truly dynamic architectural adaptation mechanism is not used in the current implementation; instead, the adaptive aspect is mostly investigated through adjustable context lengths and the efficiency-generation trade-offs that go along with it. Lastly, the ablation study should be viewed as component-sensitivity analysis rather than

independent model comparison since it inhibits specific Transformer pathways during inference without retraining the resulting variants.

The methodology will be expanded for actual edge-device deployment in subsequent work, which will include a methodical assessment of latency, memory usage, and energy consumption on embedded platforms. To further lower computing requirements, model compression methods like quantization, pruning, and knowledge distillation will be examined. Furthermore, human hearing studies, emotion- and style-conditioned generation, and dynamic context selection will be integrated to enhance interactive and customized music creation. The viability of real-time human-AI co-creative music generation in edge situations with limited resources can be further demonstrated by these extensions.

7. Conclusion

EdgeMusicTransformer, an edge-oriented Transformer framework for interactive symbolic music production with a focus on adjustable contextual modeling and computational efficiency, was introduced in this study. Symbolic music creation is made possible by the implemented model's small architecture, which has 2,443,149 trainable parameters and a comparatively cheap inference cost. With latency increasing from 7.809 ms at 128 tokens to 33.991 ms at 512 tokens and the maximum recorded throughput of 17,783.53 tokens/s at 256 tokens, the experimental results showed that context length directly affects inference performance.

The statistical properties of generated music, such as note density, pitch distribution, duration, and polyphony, are influenced by context length, according to evaluation using both created sequences and an actual MAESTRO MIDI seed. Additionally, the inference-time ablation investigation showed that eliminating the feed-forward or self-attention pathway significantly lowers computational latency but alters the properties of the resulting symbolic sequences. These results emphasize the value of taking generation behavior and computing efficiency into account together rather than addressing each factor separately. Overall, the study shows that a compact, context-configurable Transformer architecture for edge-oriented symbolic music production is feasible. In order to enable effective real-time human-AI co-creative music creation on resource-constrained platforms, the results lay the groundwork for future real-device deployment, dynamic context adaption, model compression, and human-centered evaluation.

Declarations

Ethical Approval: Since the study involves only simulation experiments with synthetic data and no human or animal subjects, ethical approval was not required.

Competing interests: The authors have no competing interests, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Authors' contributions: All authors, contributed to the study conception and design. Yifan Zhang: Conceptualization, Methodology, Software implementation, Data curation, Formal analysis, Investigation, Visualization, Writing – original draft preparation, Corresponding author. Xiangyu Wang: Supervision, Validation, Theoretical guidance, Writing – review and editing, Project administration. Both authors: Contributed to the interpretation of results, manuscript revision, and approved the final version of the manuscript.

Funding: The authors received no specific funding for this work.

Availability of data and materials: The data used in the paper will be available upon request by the corresponding author.

References

[1] Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, *107*(8), 1738-1762.

[2] Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, *7*(8), 7457-7469.

[3] Foukalas, F., & Tziouvaras, A. (2021). Edge artificial intelligence for industrial internet of things applications: An industrial edge intelligence solution. *IEEE Industrial Electronics Magazine*, *15*(2), 28-36.

[4] Ray, P. P. (2022). A review on TinyML: State-of-the-art and prospects. *Journal of King Saud University Computer and Information Sciences*, *34*(4), 1595-1623.

[5] Mycka, J., & Mańdziuk, J. (2025). Artificial intelligence in music: recent trends and challenges. *Neural Computing and Applications*, *37*(2), 801-839.

[6] Civit, M., Civit-Masot, J., Cuadrado, F., & Escalona, M. J. (2022). A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends. *Expert Systems with Applications*, *209*, 118190.

[7] Yin, Z., Reuben, F., Stepney, S., & Collins, T. (2023). Deep learning's shallow gains: A comparative evaluation of algorithms for automatic music generation. *Machine Learning*, *112*(5), 1785-1822.

[8] Ji, S., Yang, X., & Luo, J. (2023). A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges. *ACM Computing Surveys*, *56*(1), 1-39.

[9] Duan, W., Yu, Y., & Briot, J. P. (2026). Recent advances in music generation: methods, evaluation, and challenges. *Artificial Intelligence Review*.

[10] Jung, V. J. B., Burrello, A., Scherer, M., Conti, F., & Benini, L. (2024). Optimizing the deployment of tiny transformers on low-power MCUs. *IEEE Transactions on Computers*, *74*(2), 526-541.

[11] Lerch, A., Arthur, C., Bryan-Kinns, N., Ford, C., Sun, Q., & Vinay, A. (2025). Survey on the evaluation of generative models in music. *ACM Computing Surveys*, *58*(4), 1-36.

[12] Huang, Y. S., & Yang, Y. H. (2020, October). Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM international conference on multimedia* (pp. 1180-1188).

[13] Muhamed, A., Li, L., Shi, X., Yaddanapudi, S., Chi, W., Jackson, D., ... & Smola, A. J. (2021, May). Symbolic music generation with transformer-gans. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 35, No. 1, pp. 408-417).

[14] Hsiao, W. Y., Liu, J. Y., Yeh, Y. C., & Yang, Y. H. (2021, May). Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 1, pp. 178-186).

[15] Jin, C., Wang, T., Li, X., Tie, C. J. J., Tie, Y., Liu, S., ... & Huang, S. (2022). A transformer generative adversarial network for multi-track music generation. *CAAI Transactions on Intelligence Technology*, *7*(3), 369-380.

[16] Yu, B., Lu, P., Wang, R., Hu, W., Tan, X., Ye, W., ... & Liu, T. Y. (2022). Museformer: Transformer with fine-and coarse-grained attention for music generation. *Advances in neural information processing systems*, *35*, 1376-1388.

[17] Qin, Y., Xie, H., Ding, S., Tan, B., Li, Y., Zhao, B., & Ye, M. (2023). Bar transformer: a hierarchical model for learning long-term structure and generating impressive pop music. *Applied Intelligence*, *53*(9), 10130-10148.

[18] Shih, Y. J., Wu, S. L., Zalkow, F., Müller, M., & Yang, Y. H. (2022). Theme transformer: Symbolic music generation with theme-conditioned transformer. *IEEE Transactions on Multimedia*, *25*, 3495-3508.

[19] Ji, S., & Yang, X. (2023). EmoMusicTV: Emotion-conditioned symbolic music generation with hierarchical transformer VAE. *IEEE Transactions on Multimedia*, *26*, 1076-1088.

[20] Tie, Y., Wang, T., Jin, C., Li, X., & Yang, P. (2022). Wireless Communications and Mobile Computing Multitrack Music Generation Network Based on Music Rules. *Wireless Communications and Mobile Computing*, *2022*(1), 5845689.

[21] Micchi, G., Bigo, L., Giraud, M., Groult, R., & Levé, F. (2021). I keep counting: An experiment in human/AI co-creative songwriting. *Transactions of the international society for music information retrieval*, *4*(1), 263-275.

[22] Uitdenbogerd, A. L., Bown, O., Hill, C., Pegram, C., Shave, J., & Wright, B. (2023). Raging with the machine in the uncanny valley: Human–ai cocreativity in the eurovision-themed ai song contest. *Computer Music Journal*, *47*(1), 44-63.

[23] Weaver, A. (2026). Sisters, not twins: exploring artistic control and anthropomorphism through composing with a bespoke generative AI. *AI & SOCIETY*, *41*(2), 1609-1621.

[24] Cotton, K., Kaila, A. K., Jääskeläinen, P., Holzapfel, A., & Tatar, K. (2025). Imploding between the facts and concerns: analysing human–AI musical interaction. *Humanities and Social Sciences Communications*, *12*(1), 754.