MORTM: MoE-Optimized Rhythmic Transformer Model for Symbolic MIDI Generation

Takaaki Nagoshi* and Tetsuro Kitahara†
* Nihon University, Japan
E-mail: nagoshi@kthrlab.jp
† Nihon University, Japan
E-mail: kitahara@kthrlab.jp

***Abstract*—This study proposes MORTM, a new melody generation model that introduces the Mixture of Experts (MoE) architecture to address the challenges of conventional Transformer-based models in handling a diverse array of musical styles. MoE is a framework in which specialized subnetworks (experts) learn to handle distinct musical aspects. Experimental results demonstrate that MORTM equipped with MoE significantly improved predictive accuracy, reducing the perplexity from 4.92 to 2.04 when compared to the model without MoE. It also enhanced music theoretical consistency, with the usage rate of diatonic notes increasing from 76.3% to 81.2%. Furthermore, qualitative analysis implied that the generated melodies were more natural and showed greater variability in melodic features, with effective use of triplets and rests. Based on these findings, this study is one of the first to demonstrate the effectiveness of MoE in the field of symbolic music generation. We conclude that the MoE architecture significantly enhances the expressive power of the generation model, contributing to both musical consistency and diversity.**

I. Introduction

The history of computer-based music generation dates back several decades. Initially, people used rules or numbers to make music. The advent of deep learning subsequently revolutionized the field. New methods like recurrent neural networks (RNNs) [1] and long short-term memory (LSTM) [2] excelled at modeling sequential data, so people used them to make music. These methods could learn and predict what note comes next, thus improving the models' ability to learn longer range structural dependencies. However, this architecture still faced challenges in capturing the full complexity of very long range dependencies in music, which prompted the exploration of new approaches.

In 2017, the Transformer [3] changed how computers understood words, and people began using it to make music. The Music Transformer [4] from 2018 was significant, in that it showed that a mechanism called relative attention could learn the large-scale structure and rhythmic patterns of a song. This has become a standard approach in the field.

Meanwhile, GShard [5] and Switch Transformers [6], which extended the feed forward network (FFN), were proposed in the field of natural language processing. This was intended to give each FFN a specialty and reduce computational costs while significantly increasing the number of parameters. This technique of selecting a subnetwork by means of a gating mechanism is called Mixture of Experts (MoE) [7]. Furthermore, other challenges cited as Transformer issues are being addressed. For instance, FlashAttention2 [8] mitigates the quadratic computational complexity of attention, a bottleneck for long sequences, by optimizing GPU memory I/O. Additionally, methods like attention with linear biases (ALiBi) [9] have been introduced to encode sequence order, providing relative position information by adding a fixed bias to attention scores based on token distance.

Music is inherently composed of diverse elements, including a wide variety of genres, rhythms, and melodic styles. We hypothesize that the MoE architecture, which routes different inputs to specialized "expert" subnetworks, is particularly well suited for this task. This structure suggests a strong potential for effectively modeling these varied musical components, as each expert can learn to specialize in a distinct aspect of the music. This study proposes MORTM, a novel model designed to capture the beat structure of music. Unlike conventional approaches that treat music as a simple linear sequence, MORTM's primary innovation is its "metric-oriented" tokenization. By using a dedicated "Start of Measure" (SME) token, it embeds the music's underlying rhythmic framework directly into the input sequence, allowing the model to develop a deeper understanding of musical time and structure. MORTM introduces three key innovations: (1) a tokenizer that explicitly encodes rhythmic structure by marking measure boundaries and relative note positions, (2) the integration of a Mixture of Experts (MoE) layer into its FFN architecture, and (3) the implementation of FlashAttention2 for computational efficiency and ALiBi for relative positional encoding.

II. Related Research

This section provides an overview of the key technologies and previous studies in symbolic music generation that are fundamental to our research.

A. Transformer-based Melody Generation Models

Music Transformer: Proposed by Cheng-Zhi Anna Huang et al. [4], this model effectively learns long-term rhythmic dependencies over thousands of steps. It optimizes relative position representation (from Shaw et al., 2018) for linear memory using a "skewing" operation.

979-8-3315-7206-8/25/$31.00

Pop Music Transformer: Developed by Yu-Siang Huang and Yi-Hsuan Yang [10], this model uses the musical event representation, REMI, to add detailed metrical information. This greatly improved the rhythmic structure of generated pop piano music .

B. Mixture of Experts (MoE)

The idea of Mixture of Experts began with Jacobs et al.[7]. In natural language processing, the Switch Transformer by Fedus et al. is very well known. It uses "sparse activation" to keep computing costs almost the same while making the T5-base model train seven times faster and scaling it up to trillion parameter size [5], [6], [11].

C. ALiBi (Attention with Linear Biases)

Proposed by Press et al.[9], ALiBi is a method that adds a linear bias to self-attention. This helps the model understand long-range dependencies without needing absolute or relative position encodings. It captures the structure of time series data with minimal extra computational cost.

D. FlashAttention

Created by Tri Dao et al., FlashAttention [8] is an I/O aware redesign of the attention mechanism. It reduces access between the GPU's high bandwidth memory and static random access memory . This made overall training 15% faster for BERT-large and made inference three times faster for GPT-2.

III. Proposed Method

In this section, we describe the technical details of our proposed model, MORTM. The model aims to generate a musically coherent continuation of a melody based on a given melody prompt. Specifically, it seeks to maintain consistency in tonality and rhythm while creating a structurally sound melodic development. To achieve this, MORTM was specifically designed to address key challenges in music generation, such as capturing diverse expressive styles and understanding complex rhythmic structures. We have introduced three core innovations to tackle these challenges: (1) a metric-oriented tokenizer that encodes musical structure, (2) a dynamic Mixture of Experts (MoE) layer to capture diverse patterns, and (3) the use of FlashAttention2 and ALiBi for computational efficiency and to handle long-range dependencies.

A. Preprocessing and Tokenizer

For this study, we utilized the META MIDI Dataset [12], a large scale, publicly available collection of over 430,000 MIDI files. This dataset is notable for its vast size and stylistic diversity, making it a valuable resource for training data hungry deep learning models. However, due to its uncurated nature and our specific focus on monophonic melody, we applied several preprocessing steps. We filtered the dataset to include only the saxophone parts from songs in 4/4 time. To further enrich the dataset and mitigate key bias, we performed data augmentation by transposing each song to all 12 keys. This process resulted in a final dataset of approximately 43,000 songs, comprising around 800,000 measures of music.

TABLE I
Tokens list

Token	Description
Pitch	The MIDI note number of a note
Duration	The length of the note in ticks
ShiftTime	The start time of the note, normalized to 96 ticks
<SME>	Marks the start of a measure
<S_SEQ>	Marks the start of the sequence
<E_SEQ>	Marks the end of the sequence

We cut the MIDI data into 12-measure parts. Next, we used a tokenizer for MORTM to convert the music into tokens. This tokenizer creates tokens for basic musical information as well as for the structure of the music.

Using the <SME> token helps the model understand the timing of the music measure by measure. This process helps the model learn the right time structure. The tokens we created are beneficial for the Transformer model, in that they store information about time and structure, which helps the model to generate music with rhythm that is both structurally sound and variability in melodic features. The converted sequence of tokens is shown in the figure.1: Where each measure begins

$$\texttt{<S\_SEQ>} \rightarrow \overbrace{\texttt{<SME>} \rightarrow (s_1, p_1, d_1) \rightarrow \ldots}^{\text{M.1}} \rightarrow \texttt{<SME>} \rightarrow \cdots \rightarrow \texttt{<E\_SEQ>}$$

Fig. 1. The token sequence structure of MORTM.

with an <SME> token, followed by a series of melodic tokens (s_i, p_i, d_i) representing the Shift Time, Pitch, and Duration for each musical note within that measure. This explicit structure allows the model to effectively learn the metrical hierarchy of the music.

B. Architecture

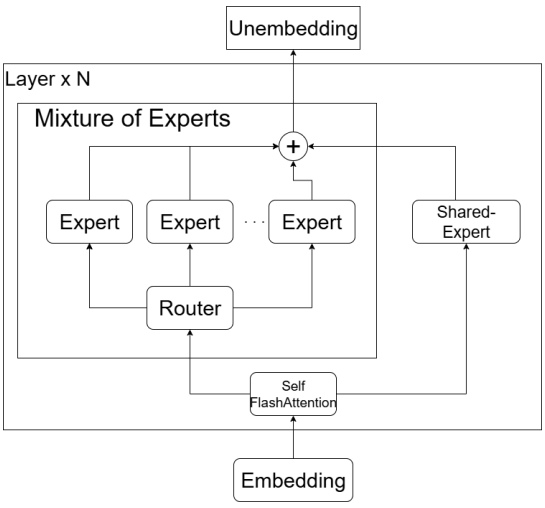

Fig. 2. MORTM Architecture

a) Embedding & Decoder.: We embed token IDs with a learnable table of size $|\mathcal{V}| \times d_{\text{model}}$. Each block is a pre-norm causal Transformer decoder layer (self-attention + feed-forward). We adopt FlashAttention2 for efficient attention

computation, and ALiBi for robust relative position encoding (no absolute position indices are concatenated).

b) Router (gating).: Given the block input $h \in R^{d_{\text{model}}}$, a linear gate produces logits $g = W_{\text{gate}}h+b \in R^E$ for E experts. We select the Top-2 indices $S = \text{top2}(g)$ and compute mixture weights by restricting softmax to S: $\alpha = \text{softmax}(g_S) \in R^2$.

c) Experts (SwiGLU-FFN).: Each independent expert is a position-wise gated FFN using SwiGLU:

$$\text{FFN}_{\text{SwiGLU}}(h) = W_2\Big[(W_{1a}h+b_{1a})\odot\text{Swish}(W_{1b}h+b_{1b})\Big]+b_2,$$

where $\odot$ denotes elementwise product. Dropout is applied after the hidden transform. Inner widths are set to keep the per-token *active* compute comparable to a dense FFN.

d) Shared Expert (dense FFN).: In parallel, a shared (dense) FFN runs for all tokens:

$$\text{FFN}_{\text{dense}}(h) = W_2' \ \text{GELU}(W_1'h + b_1') + b_2'.$$

e) MoE aggregation and residual.: The MoE feed-forward output is the weighted sum of the two selected experts plus the shared pathway:

$$\text{MoE}(h) = \alpha_1 \,\text{FFN}_{e_1}(h) + \alpha_2 \,\text{FFN}_{e_2}(h) + \text{FFN}_{\text{dense}}(h),$$

$$e_1, e_2 \in S.$$

The block output follows the standard residual update with pre-norm: $h \leftarrow h + \text{Dropout}\big(\text{MoE}(\text{LN}(h))\big)$.

The learned router encourages specialization across experts without increasing the per-token active FLOPs. We use a Top-2 routing strategy throughout and keep all other hyper-parameters identical in the non-MoE baseline (where the MoE layer is replaced by a single dense FFN).

C. Melody Generation Procedure

MORTM is trained as an autoregressive Transformer language model over symbolic music tokens (Table I). Each note is represented by a sequence of three successive tokens: Position (within the 96-tick measure), Pitch, and Duration, preceded by a measure-start token <SME>. A melody sequence begins with <S_SEQ> and ends with <E_SEQ>.

During inference, MORTM generates melodies token by token in the same way as text generation. We first feed a short melody prompt (two measures in our experiments). The model then outputs the next token distribution; we sample one token using nucleus sampling (top-p) with temperature and append it to the input. This procedure is repeated, so that the newly generated tokens are fed back into the model to produce further continuations. The process terminates when <E_SEQ> is produced or when a pre-set maximum number of measures (12 in this work) is reached.

IV. Evaluation Experiment

In this section, we compare MORTM (without MoE) and MORTM (with MoE).

A. Input Conditions

We provided each model with the same input (a prompt) to see what it would create. The prompt is a two measure melody based on the C minor scale. This melody has a shuffle rhythm (a swing rhythm), which is often used in jazz. Based on this common prompt, each model generated the melody that comes next. We then compared and evaluated their outputs.

During evaluation, perplexity is computed under teacher forcing on the test set, while qualitative examples and musical-feature analyses are obtained by free autoregressive decoding from the common two-measure prompt.

B. Perplexity

Perplexity is a standard way to measure language models. It shows how well the model can guess the next token in a test set. A lower score is preferred, as it means the model's predictions are more accurate.

C. Comparative Musical Analysis

To assess the musical quality of the generated outputs, we conducted a qualitative comparative analysis. The melodies generated by the two models (MORTM with and without MoE) from the same prompt were transcribed into sheet music. We analyzed these scores, focusing on the differences in their musical characteristics. The analysis centered on three key aspects:(1) rhythmic diversity and complexity, (2) the placement and musical function of rests, and (3) melodic structure and development.

D. Quantitative Evaluation of Musical Features

To check the musical quality of the melodies, which Perplexity alone cannot do, we use the following metrics:

Pitch-Class Histogram Divergence: This checks the extent of difference in the distribution of pitch classes (the 12 notes like C, D, E) in the generated melody compared to the training data. This aids us in determining if the melody maintains a natural sounding scale structure.

Rhythm Pattern Divergence: This checks how different the distribution of rhythm patterns is in the generated melody compared to the training data, helping us evaluate the variety and naturalness of the rhythms.

V. Experimental Results

This section shows the experimental results for the melodies generated by the proposed method.

A. Results of Qualitative Analysis

First, to assess the musical quality of the generated melodies, we conducted a comparative music-theoretical analysis based on their transcriptions into sheet music. When we observed an improvement in the diversity of melodies generated by MORTM (with MoE).

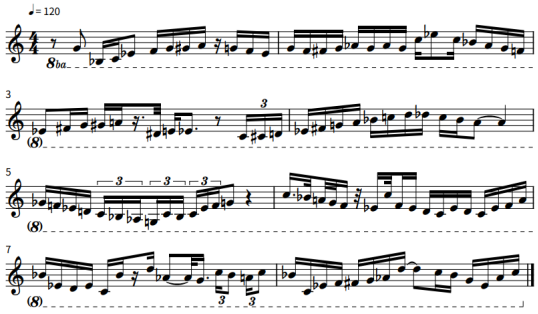

Fig. 3. Result of melody generation using MORTM with MoE

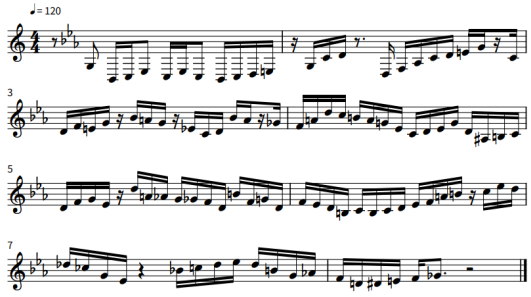

Fig. 4. Result of melody generation using MORTM without MoE

Rhythm Diversity and Understanding of Performance Techniques: MORTM (without MoE) tended to create simple rhythms, mostly using 16th notes. In contrast, MORTM (with MoE) effectively added triplets among the 16th notes, making the rhythm more diverse. This is a technique often used in jazz saxophone playing. It suggests that MORTM (with MoE) may be learning not just the sequence of notes, but also the specific performance styles and techniques of the instrument.

Naturalness of Rests: The melodies from MORTM (without MoE) contain few rests and the notes are continuous, which sounds mechanical, providing no place to "breathe." In contrast, MORTM (with MoE) places rests in appropriate spots, creating space between phrases. This is a more human-like and natural melody structure, considering the breathing of a wind instrument like the saxophone. This "breathing" like expression is an important element that connects to the singable quality (cantabile) of a melody.

Structure and Development: MORTM (without MoE) showed that it can build a consistent and stable melody structure by repeating the pitch and rhythm effectively. This is an important way to compose music and it succeeded in creating melodies that are easy for listeners to follow. In contrast, MORTM (with MoE) exhibited increased rhythmic diversity. While the baseline model primarily generated continuous 16th-note patterns, the MoE-equipped model developed the melody by incorporating a wider range of rhythmic elements, including triplets and rests. This resulted in less predictable melodic structures and a departure from the rhythmic patterns of the baseline model. Based on this score-based analysis, we conclude that the melodies generated by MORTM with MoE are melodically richer and diverse. We believe the MoE architecture worked well in capturing the small details and structural consistency of the melodies.

B. Basic Performance Comparison of Models

Next, we compared the training loss, validation loss, and Perplexity to objectively evaluate the models' prediction performance. The results are shown in Table II.

TABLE II
PERPLEXITY

Model / comparison	Train Loss	Validation Loss	Perplexity
MORTM(without MoE)	1.59	1.60	4.92
MORTM(with MoE)	0.84	1.02	2.04

As seen in Table II, MORTM (with MoE) performed much better than MORTM (without MoE) in all metrics. Looking at Perplexity, MORTM (with MoE)'s score of 2.04 is less than half of MORTM (without MoE)'s 4.92. This shows that MORTM (with MoE) can predict the next token (note or rhythm) with much more confidence, which better captures structural characteristics of the melody.

The training loss dropped from 1.59 for MORTM (without MoE) to 0.84 for MORTM (with MoE). This shows that the improved model expression from the MoE architecture greatly increased its ability to fit the training data. More importantly, the validation loss, which shows how well the model works on new data, was also lower for MORTM (with MoE) (1.02) than for MORTM (without MoE) (1.60). This means the model is not just overfitting to the training data but can also make good predictions for new melodies. This strongly supports that adding MoE was effective.

C. Pitch-Class Histogram

To analyze the tonal features of the generated melodies, we compared the pitch-class histograms.

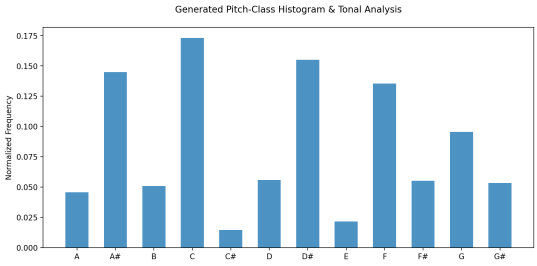

Fig. 5. Pitch-Class Histogram Result: Using MoE

In general, both models' distributions are similar, showing that melodies are created based on the key of the input prompt, which is the C minor scale. Looking closer, MORTM (with MoE) was slightly more faithful to the C minor scale. The use of important notes that define the C minor scale—like the tonic (C), the third (E ♭), and the dominant (G) was higher in MORTM (with MoE) than in MORTM (without MoE). We defined the seven diatonic notes of the C natural minor

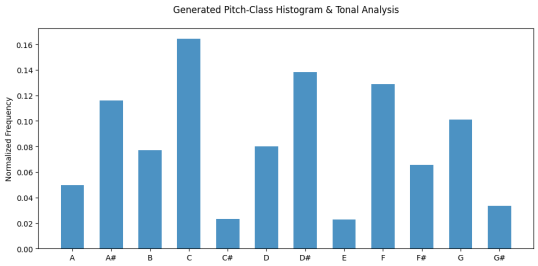

Fig. 6. Pitch-Class Histogram Result: Not Using MoE

scale (C, D, E♭ (D♯), F, G, $A\flat$ (G), $B\flat$ (A♯)) and classified other pitches (like C♯, E, B) as chromatic notes. Below is a comparison of how often they appeared:

MORTM(without MoE):

- Diatonic note rate:76.3%
- Chromatic note rate:23.7%

MORTM(with MoE):

- Diatonic note rate:81.2%
- Chromatic note rate:18.8%

The results show that MORTM (with MoE) used diatonic notes 4.9% more than MORTM (without MoE). The use of chromatic notes like C♯, E, and B was much lower, while the main diatonic notes like C, D♯, F, G, and Bb were used about the same or slightly more.

These findings suggest that MORTM using its MoE layer, improved its ability to stick to the C natural minor scale compared to MORTM (without MoE). This made the generated melodies feel more tonal and unified. In conclusion, MORTM (with MoE) improved its scale adherence by about 5% over MORTM (without MoE). It is a model that has greatly improved its musical theoretical correctness by making it possible to generate melodies mainly with diatonic elements.

D. Rhythm Pattern Divergence

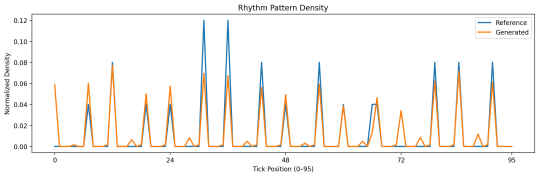

Fig. 7. Rhythm Pattern Divergence Result: Using MoE

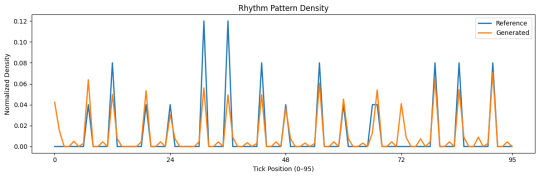

Fig. 8. Rhythm Pattern Divergence Result: Not Using MoE

To evaluate the rhythmic consistency of the generated melodies, we compared the distribution of note start times (tick positions) within a single measure. The blue line represents the rhythm pattern of the input prompt (reference), and the orange line represents the rhythm pattern of the melody generated by the model (generated). In the plot for MORTM (without MoE), the generated rhythm (orange line) often has notes at unexpected times, away from the peaks of the reference rhythm (blue line). The graph fluctuates irregularly, suggesting that the model has not fully captured a stable rhythmic structure, and the timing of the notes is scattered.

In contrast, in the plot for MORTM (with MoE), the generated rhythm is very faithful to the peaks of the reference rhythm. This means the model correctly understands the basic rhythmic structure of the input prompt and can generate a melody that follows it. There are also a few notes at times not present in the reference rhythm. However, this is not just noise but can be seen as creative variation based on the original rhythm—a positive sign of musical "diversity."

Given these results, it is clear that MORTM (with MoE) is better than MORTM (without MoE) at learning and reproducing rhythmic structures, and it has the ability to generate more musically consistent melodies.

VI. Conclusion

In this study, we proposed MORTM (with MoE). This model extends the architecture of the MORTM melody generation model by replacing the traditional static FFN layer with a dynamic Mixture of Experts (MoE) structure. We tested its effectiveness from multiple angles through comparison experiments with MORTM (without MoE), which does not have MoE.

The experimental results show that the proposed model is superior. First, in terms of Perplexity, MORTM (with MoE) achieved a score less than half that of MORTM (without MoE). This confirmed a major improvement in basic prediction performance.

MORTM (with MoE) also showed significant improvements in musical quality. The pitch-class histogram analysis showed that the use of diatonic notes increased by about 5%. This provides quantitative indication that the generated melodies are more consistent with music theory and have a more stable sense of key. The rhythm pattern divergence analysis confirmed a much better adherence to the input rhythm and the ability to add creative variations while keeping the rhythmic foundation. The subjective evaluation also noted the generation of more expressive and natural melodies, with skillful use of triplets and rests.

From these results, we conclude that adding the MoE architecture greatly improves the expressive power of the melody generation model. It makes a major contribution to achieving both musical consistency and diversity. This study is one of the first to show the effectiveness of MoE in the field of symbolic music generation. It provides a new direction for designing future highperformance music generation models.

In future work, we will improve the evaluation in two ways. First, we plan to compare MORTM with other melody generation models, such as Music Transformer and Museformer,

under the same conditions. Second, we plan to run a listening test with human participants to check whether the melodies generated with MoE sound more natural and musical. These steps will provide a clearer and more comprehensive evaluation of the effectiveness of MORTM.

references

REFERENCES

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, A foundational paper for training Recurrent Neural Networks (RNNs) via backpropagation.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, The original paper introducing Long Short-Term Memory (LSTM) networks.

[3] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, The seminal paper that introduced the Transformer architecture., 2017.

[4] C.-Z. A. Huang, A. Vaswani, N. Parma, N. Shazeer, and I. Simon, "Music transformer: Generating music with long-term structure," in *International Conference on Learning Representations*, Introduced the Music Transformer, applying the Transformer architecture to music generation., 2018.

[5] D. Lepikhin, H. Lee, Y. Xu, *et al.*, "Gshard: Scaling giant models with conditional computation and automatic sharding," *arXiv preprint arXiv:2006.16668*, 2020, Introduced GShard, a system for scaling models using MoE.

[6] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: : Scaling to trillion parameter models with simple and efficient sparsity," *arXiv preprint arXiv:2101.03961*, 2021, Proposed the Switch Transformer, a simplified and efficient MoE architecture.

[7] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991, The foundational paper that originally proposed the Mixture of Experts (MoE) concept.

[8] T. Dao, "Flashattention-2: Faster attention with better parallelism and work partitioning," *arXiv preprint: 2307.08691*, 2023.

[9] M. L. Ofir Press Noah A. Smith, "Train short, test long: Attention with linear biases enables input length extrapolation," *arXiv preprint 2108.12409*, 2021.

[10] M. L. Ofir Press Noah A. Smith, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," *arXiv preprint 2002.00212*, 2020.

[11] N. Shazeer, A. Mirhoseini, K. Maziarz, *et al.*, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," in *International Conference on Learning Representations*, A key paper on the modern Mixture of Experts (MoE) layer for large-scale models., 2017.

[12] P. P. Jeff Ens, "Building the metamidi dataset: Linking symbolic and audio musical data," *ISMIR 2021*, 2021.