Exploring Non-Autoregressive Transformers for Efficient Adaptive Music Composition

Αλέξης Σπηλιωτόπουλος[1,*], Σπύρος Πολυχρονόπουλος[2], Ιωάννης Παναγάκης[1]
[1]Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
[2]Ελληνικό Μεσογειακό Πανεπιστήμιο
[*]alexis.spiliot@gmail.com

ABSTRACT

In this study, we investigate the application of a non-autoregressive Transformer encoder-based model, MusicBERT, for symbolic music generation, focusing on generating musical continuations based on given contexts. We performed a modified fine-tuning of a pre-trained MusicBERT model using different token unmasking techniques, including Sequential Unmasking, Random Sampling, Parallel Unmasking, and Causally Biased Iterative Decoding. We assessed each method for coherence, diversity, and computational efficiency. Preliminary results showed that Causally Biased Iterative Decoding performed best. The methods showed low GPU VRAM utilization and rapid execution speeds, making such models promising for real-time music composition.

Εξερεύνηση μη αυτοπαλινδρομικών Μετασχηματιστών για αποτελεσματική προσαρμοστική σύνθεση μουσικής

ΠΕΡΙΛΗΨΗ

Σε αυτή τη μελέτη, διερευνούμε την εφαρμογή ενός μοντέλου που βασίζεται σε Μετασχηματιστή με κωδικοποιητή, το MusicBERT, για συμβολική παραγωγή μουσικής, εστιάζοντας στη δημιουργία μουσικών συνεχειών με βάση δεδομένο περιεχόμενο. Κάναμε μια τροποποιημένη προσαρμογή ενός προεκπαιδευμένου MusicBERT, χρησιμοποιώντας διαφορετικές τεχνικές συμπλήρωσης μάσκας, όπως Διαδοχική, Τυχαία και Ταυτόχρονη Συμπλήρωση Μάσκας και Επαναληπτική Αποκωδικοποίηση με αιτιώδη μεροληψία, και αξιολογήθηκαν ως προς τη συνοχή, την ποικιλομορφία και την υπολογιστική απόδοση. Πρώιμα αποτελέσματα έδειξαν ότι η επαναληπτική αποκωδικοποίηση με αιτιώδη μεροληψία είχε καλύτερη απόδοση. Οι μέθοδοι είχαν χαμηλή χρήση VRAM και γρήγορες ταχύτητες εκτέλεσης, καθιστώντας αυτά τα μοντέλα υποσχόμενα για σύνθεση μουσικής σε πραγματικό χρόνο.

Introduction

The adoption of artificial intelligence (AI) in music composition is revolutionizing the process of music creation, providing innovative tools for generating new musical ideas and improving arranging and mixing. Since the mid-20th century, and increasingly with recent advances in deep learning, and the availability of large amounts of data, AI has enabled automatic music composition by leveraging large datasets. However, real-time music generation with low latency remains a major challenge, particularly for applications like live performances and dynamic soundtracks in video games or virtual reality (VR). Maintaining coherence and expressiveness in AI-generated music is also important for preserving the emotional impact and artistic quality expected by audiences.

Automated music composition has advanced since the early rule-based systems, such as the Illiac Suite [10], and the probabilistic models like Markov chains. Neural networks, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) [11], were transformative, enabling better handling of longer-term dependencies in music data. Recently, more complex architectures, including Generative Adversarial Networks (GANs) [9] and Transformer models [8], have further transformed the field, with GANs using competing networks for realistic outputs and Transformers surpassing RNNs in capturing long-range dependencies [1][7]. These advancements, supported by greater computational power and access to large, high-quality datasets, allow researchers in AI music to explore more complex models, leading to deeper insights into music generation and improved coherence and expressiveness in the results. However, these systems, particularly Transformer-based ones, require large datasets and significant computational resources for training, and the challenge of fast music generation still remains.

In this paper, we explore the use of non-autoregressive models for music generation, specifically using MusicBERT [2], a Transformer encoder-based architecture. We fine-tuned a pre-trained instance of the MusicBERT model in order to adapt it for generating musical continuations. Given some input musical context, we modified the standard Masked Language Modeling (MLM) task of predicting sequences of tokens for generating musical continuations. Our approach aims to leverage the model's understanding of musical patterns and test various sampling techniques to enhance output quality. This study explores whether non-autoregressive methods can address the limitations of autoregressive models, such as slow generation speeds, while maintaining coherence and creativity, although the results remain preliminary.

1. Related Work

Transformer models have improved symbolic music generation by effectively capturing long-range dependencies and managing the complicated structure of musical compositions. Early approaches, such as the Music Transformer [7], incorporated self-attention mechanisms to produce coherent melodies with longer-term structure. Building on this foundation, models such as Compound Word Transformer [1] improved control over multi-track compositions by integrating hypergraph structures to more effectively represent the relationships between different musical elements. Additional advancements, like PopMAG [17] and Pop Music Transformer [14], focus on multi-track representations and beat-based modeling for improving the expressiveness and structure of generated music. However, these transformer-based approaches primarily concentrate on autoregressive methods, where each token is generated in a sequential manner. This leads to longer inference times, especially for large compositions, and limits their ability to handle specific infilling tasks or precise musical constraints.

More recently, non-autoregressive approaches have gained traction, particularly in the domain of audio generation. Models like SoundStorm [13] and MAGNET [15] employ parallel decoding strategies to efficiently generate high-quality audio. However, these methods are primarily designed for continuous audio representations, such as speech and sound synthesis, rather than symbolic music. While non-autoregressive models offer faster generation, they often struggle with producing long, coherent sequences and maintaining fidelity in complex musical compositions, as observed in methods like VAMPNet [16] and StemGen [6]. Additionally, these approaches require training models from scratch, which can be computationally expensive and harder to scale, especially for masked symbolic music modeling tasks.

2. Methodology

In this section, we outline our approach for generating musical continuations using MusicBERT, a pre-trained transformer model developed for symbolic music understanding. We modified the fine-tuning process to adapt the model specifically for generating continuation sequences given some musical context. We construct pairs of data consisting of a context, which represents the musical input context, and a continuation, which, as the name suggests, represents the continuation that comes after the context. Our fine-tuning procedure extends the standard MLM task by iteratively predicting tokens within a continuation. By experimenting with various unmasking techniques, we aim to evaluate their effects on the model's ability to maintain coherence, diversity, and computational efficiency during the generation process.

2.1 Model Selection

For our experiments, we used MusicBERT, a transformer encoder-based model pre-trained on symbolic music data. It follows the RoBERTa [4] architecture, using attention mechanisms to capture both local and global patterns in the data. MusicBERT has been pre-trained on large symbolic music datasets, enabling learning a rich representation of musical structures such as harmony, rhythm, and instrumentation. Its ability to handle long-range dependencies in sequential data makes it an ideal choice for tasks involving symbolic music generation and analysis. We pre-trained the model on the Masked Language Modeling (MLM) task [12], masking certain tokens in a sequence and teaching it to predict the missing tokens based on the surrounding context. This makes MusicBERT particularly useful for

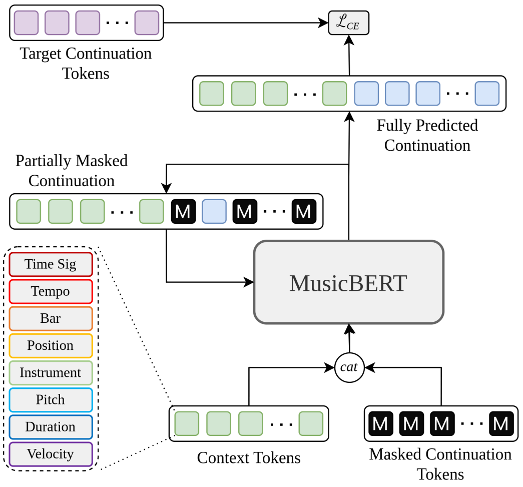
Target Continuation Tokens
$\mathcal{L}_{CE}$
Fully Predicted Continuation
Partially Masked Continuation
M
MusicBERT
Time Sig
Tempo
Bar
Position
Instrument
Pitch
Duration
Velocity
cat
Context Tokens
Masked Continuation Tokens

Figure 2.1 A flowchart illustrating the fine-tuning method, where the model receives a concatenated context and masked continuation. The masked tokens are iteratively unmasked and then evaluated against the target continuation using cross-entropy loss. Parallel Unmasking generates the final continuation in one pass. The OctupleMIDI encoding elements are displayed in the lower left.

generating symbolic music continuations, as it can model complex musical dependencies across time.

2.2 Fine-tuning Process

To adapt MusicBERT for generating musical continuations, we fine-tuned the model using a modification of the standard MLM technique, as shown in Figure 2.1. In this process, the model receives two sequences as input: a *context* sequence that contains unmasked tokens representing the initial musical input, and a masked *continuation* sequence that the model must predict. The context is combined with the masked continuation, and the model progressively unmasks the continuation by predicting a subset of masked tokens in each iteration, which are then incorporated back into the sequence. This iterative process is continued until all the masked tokens are predicted. We experimented with several unmasking techniques (described in Section 2.2) to evaluate their impact on the quality and coherence of the generated music. For each technique, a different model variant was produced.

2.3 Dataset and Music Token Representation

For our fine-tuning musical corpus, we chose the POP909 [3] dataset, a collection of 909 pop songs transcribed to MIDI format. We performed an 80-20 train-test split on the dataset, with 80% of it used for fine-tuning and the remaining 20% for testing. For each song, multiple context-continuation pairs of varying lengths were constructed, ensuring that the model could potentially learn to generate continuations given a variety of musical prompts. To encode the symbolic musical

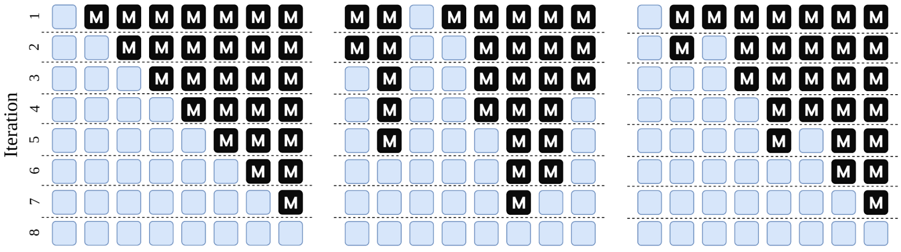

Figure 3.1 The three iterative unmasking methods for an initially masked continuation sequence of 8 tokens (context sequence is not shown).

data, we used the OctupleMIDI [2] encoding, shown in Figure 2.1, which represents each MIDI note as a tuple of eight elements: *bar*, *position*, *instrument*, *pitch*, *duration*, *velocity*, *time signature*, and *tempo*. This method aligns with MusicBERT's pre-training data encoding, allowing the model to better utilize its knowledge more effectively.

3. Experiments

3.1 Experimental Setup

The experiments were conducted using a single NVIDIA RTX 4060 GPU with 8GB of VRAM. For each method, we fine-tuned MusicBERT for 50 epochs. We set the batch size to 64 and optimized the model using the Adam [5] optimizer with β_1=0.9, β_2=0.98, and ε=1e-6. Additionally, we applied L2 weight decay of 0.1 to prevent overfitting. The learning rate was linearly warmed up over the first 50,000 steps to a peak value of 5e-5, followed by a polynomial decay schedule throughout the training process. A dropout rate of 0.1 was applied across all layers.

3.2 Sampling Techniques

During fine-tuning and evaluation, we experimented with four different sampling techniques—three iterative and one non-iterative—to predict the masked tokens in the continuation sequence by sampling from the model's output logits. A brief visualization of the three iterative methods is shown in Figure 3.1.

In *Sequential Unmasking*, tokens are predicted in a left-to-right manner. In each iteration, the model predicts the leftmost masked token in the continuation part of the input, which is then updated with the newly predicted token. This process continues until all tokens in the continuation have been unmasked, while the context remains unchanged throughout the iterations. In *Random Sampling*, tokens are unmasked at random. The continuation is updated with these new tokens, and this process is repeated until all masked tokens have been predicted. *Parallel Unmasking* predicts the entire continuation sequence in one pass, without any iterative steps. This method follows the same MLM procedure used when MusicBERT was pre-trained. Lastly, we used a method similar to the *Causally Biased Iterative Decoding* method, inspired by the StemGen [6], in which tokens

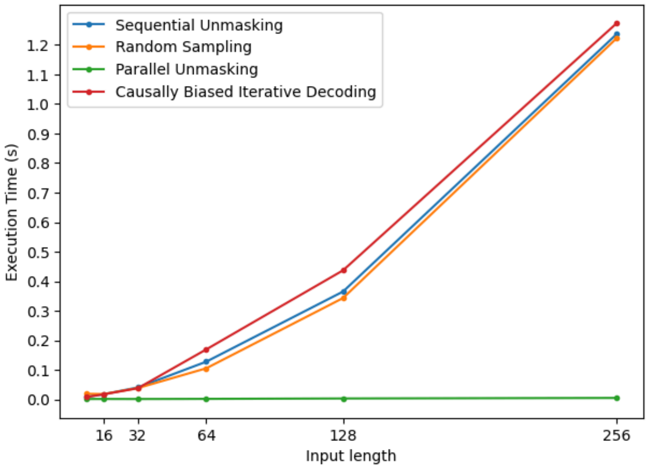

Figure 4.1: Prediction times for different input lengths across each unmasking method. The numbers shown on the x-axis represent the total number of notes in the concatenated context-continuation pairs, while the actual input length is 8 times larger due to the OctupleMIDI format.

earlier in the continuation sequence are prioritized for unmasking, creating a causal structure in which earlier predictions influence subsequent ones. A ranking function combines the model's confidence, a bias toward earlier tokens, and random noise to potentially balance maintaining structure and having diversity in the generated sequence.

For all methods, predictions were made by sampling from the top-k (k=5) probabilities from the model's output distribution. This top-k sampling limits the selection of the next token to the k most likely candidates, promoting diversity while reducing the risk of repetitive predictions.

4. Preliminary Results

We evaluated each sampling technique using random context-continuation pairs from the test set. As shown in Figure 4.1, parallel unmasking maintains near-constant execution time, while the other methods scale linearly with input length. All techniques offer sub-second performance for inputs smaller than 200 tokens, making them ideal for continuous real-time generation. Even with larger inputs, times up to 1.3s remain within a practical range for real-time applications. Also, inference required only around 2GB of GPU memory, making these methods accessible on consumer-grade GPUs.

However, all the methods showed significant flaws in the musical quality of the generated continuations, most likely due to insufficient training. More specifically, the most challenging sub-tokens to predict in the OctupleMIDI encoding were the bar and position sub-tokens, which frequently lead the model to bad note placement behavior, with notes either densely clustered or too sparse across bars. This tendency of the model to place notes far apart was observed across all techniques, resulting in large pauses in the music. Additionally, in the *Sequential Unmasking* method, the pitch sub-token was overly repetitive, reducing the output's diversity and expressiveness. On the contrary, *Random Sampling* led to excessive prediction variability, illustrating that relying too heavily on random selection degrades music quality. The *Causally Biased Iterative Decoding* method, however, demonstrated a better balance between structure and diversity. Although its performance was initially poor during fine-tuning, it eventually achieved a more balanced output, though still not entirely sufficient. Adjusting the factors in the ranking function revealed a clear influence on the quality of the generated music, as expected. At last, the *Parallel Unmasking* method showed the worst results, which we attribute to the lack of iterative refinement.

In general, qualitative evaluations showed that the generated music maintained the key of the input prompt, indicating that the methods captured some structural aspects of the music. As training progressed, the differences between sampling methods became more pronounced, with each method exhibiting distinct characteristics in terms of coherence and variability.

5. Conclusion

In this study, we investigated the use of a non-autoregressive Transformer-based model, MusicBERT, for symbolic music generation. Through fine-tuning the pre-trained MusicBERT, we tested various sampling techniques to assess their impact on generating musical continuations. While we achieved notable efficiency in terms of low GPU VRAM usage and fast execution times, the generated sequences revealed notable issues in musical quality, such as incorrect note placement, large pauses, and variability in structure. The Causally Biased Iterative Decoding method showed the most promise in balancing structure and diversity, though further improvement is needed. Future work will focus on addressing these challenges through extended training and evaluation. While the results are preliminary, our approach demonstrates potential for efficient real-time music generation in resource-constrained environments.

6. References

[1] Hsiao, W.-Y., Liu, J.-Y., Yeh, Y.-C. and Yang, Y.-H. “Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs”, *Proceedings of the AAAI Conference on Artificial Intelligence*, **35(1)**, pp 178-186 (2021)

[2] Zeng, M., Tan, X., Wang, R., Ju, Z., Qin, T., & Liu, T. "MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training" in Findings of the Association for Computational Linguistics: ACL-IJCNLP, pages 791-800 (2021).

[3] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, G. Bin, and G. Xia, “POP909: A Pop-song Dataset for Music Arrangement Generation,” in *ISMIR* (2020)

[4] Liu, Y., Ott, M., & Goyal, N. Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov “Roberta: A robustly

optimized bert pretraining approach" *arXiv preprint arXiv:1907.11692, 1(3.1),* 3-3 (2019).
[5] Kingma, Diederik P. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).
[6] Parker, J., Spijkervet, J., Kosta, K., Yesiler, F., Kuznetsov, B., Wang, J., Avent, M., Chen, J., & Le, D. "STEMGEN: A Music Generation Model That Listens." *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp 1116-1120. (2023).
[7] Huang, C.A., Vaswani, A., Uszkoreit, J., Shazeer, N.M., Simon, I., Hawthorne, C., Dai, A.M., Hoffman, M.D., Dinculescu, M., & Eck, D. "Music Transformer: Generating Music with Long-Term Structure." *International Conference on Learning Representations* (2018).
[8] Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. "Attention is All you Need". *Neural Information Processing Systems* (2017).
[9] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. "Generative adversarial nets" *Advances in neural information processing systems*, ***27*** (2014).
[10] L. Hiller, L. Isaacson. *"Experimental Music: Composition with an Electronic Computer"* McGraw-Hill, New York (1959).
[11] S. Hochreiter and J. Schmidhuber "Long Short-Term Memory" Neural Comput. **9(8)**, pp 1735–1780 (1997)
[12] Devlin, J., Chang, M., Lee, K., & Toutanova, K. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" *North American Chapter of the Association for Computational Linguistics* (2019).
[13] Borsos, Z., Sharifi, M., Vincent, D., Kharitonov, E., Zeghidour, N., & Tagliasacchi, M. "SoundStorm: Efficient Parallel Audio Generation". ArXiv, abs/2305.09636 (2023)
[14] Huang, Y., & Yang, Y. "Pop Music Transformer: Generating Music with Rhythm and Harmony". ArXiv, abs/2002.00212 (2020).
[15] Ziv, A., Gat, I., Lan, G.L., Remez, T., Kreuk, F., D'efossez, A., Copet, J., Synnaeve, G., & Adi, Y. "Masked Audio Generation using a Single Non-Autoregressive Transformer" ArXiv, abs/2401.04577 (2024).
[16] Garcia, H.F., Seetharaman, P., Kumar, R., & Pardo, B. "VampNet: Music Generation via Masked Acoustic Token Modeling" ArXiv, abs/2307.04686 (2023).
[17] Ren, Y., He, J., Tan, X., Qin, T., Zhao, Z., & Liu, T. "PopMAG: Pop Music Accompaniment Generation" *Proceedings of the 28th ACM International Conference on Multimedia* (2020).