Why Do Music Models Plagiarize? A Motif-Centric Perspective

Tatsuro Inaba
MBZUAI
tatsuro.inaba@mbzuai.ac.ae

Kentaro Inui
MBZUAI
kentaro.inui@mbzuai.ac.ae

Abstract

This paper examines plagiarism-like behaviors in Transformer-based models for symbolic music generation. While these models can produce musically convincing outputs, they also risk copying fragments from training data. We hypothesize that such plagiarism arises from local overfitting of motifs—short, recurrent patterns within a piece—rather than from global overfitting. To test this hypothesis, we analyze motif repetition in training data and assess motif-level plagiarism through perplexity and the originality of generated samples. Experiments show that frequently repeated motifs are predicted with lower perplexity and are more likely to reappear in generated outputs. We also explore preliminary strategies to mitigate plagiarism—label smoothing, transposition-based data augmentation, and Top-K sampling—and evaluate their effectiveness.

1 Introduction

Symbolic music generation has advanced rapidly with neural sequence models, which can now produce musically coherent compositions [13]. However, this progress comes with a growing ethical concerns due to their tendency to plagiarize training data [15, 21]. As the deployment of generative music systems expands into creative domains, addressing plagiarism is essential for ensuring their ethical reliability and long-term safety.

This raises an important question: why do plagiarism-like behaviors appear? While regularization methods such as early stopping and dropout are generally helpful to prevent overfitting, they do not suffice to prevent plagiarism in symbolic music generation. We hypothesize that plagiarism arises not from global overfitting but from local overfitting of motifs—short, recurrent patterns within a musical piece. Because motifs are repeated multiple times within a single composition, they present highly predictable structures for the model to learn. As a result, the model may memorize and reproduce these fragments with high fidelity despite the impression of overall generalization.

In this work, we investigate this hypothesis by analyzing how motif frequency in training data influences model predictions and generated outputs. Here, we operationalize "motif" at the bar level—that is, we focus on repetitions of single bars as fixed-length units, while acknowledging that motifs in music are generally variable-length phrases. Our findings show that motifs are predicted with low perplexity and disproportionately reproduced in generated samples as illustrated in Figure 1. This suggests that plagiarism risk is closely tied to motif and cannot be explained solely by conventional notions of overfitting. We further explore three strategies to mitigate motif-level memorization; label smoothing, Top-K sampling, and data augmentation via transposition.

Our study contributes to a deeper understanding of how plagiarism arises in symbolic music generation models and provides concrete directions for mitigation. We argue that motif-level analysis is essential for developing ethically reliable and safe generative models, and we hope this work stimulates further discussion at the intersection of creativity and ethics.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI for Music.

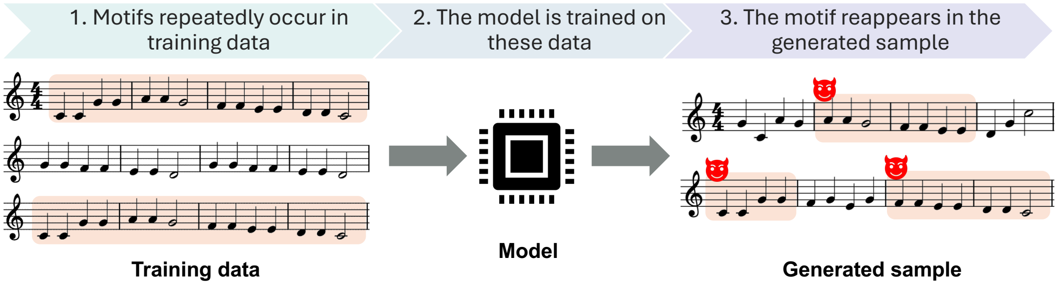
1. Motifs repeatedly occur in training data
2. The model is trained on these data
3. The motif reappears in the generated sample
Training data
Model
Generated sample

Figure 1: Illustration of motif-level copying. The model trained on data containing repeated motifs reproduces these motifs in its generated samples.

2 Motif Repetition and Plagiarism Metrics

In this section, we describe the methods used to measure similarity between musical segments, to quantify motif repetition in the training data, and to define plagiarism-oriented evaluation metrics.

2.1 Measuring bar-level similarity with the jaccard index

To quantify the similarity between symbolic music segments, we adopt the Jaccard index. Given two segments x and y of the same time span (e.g., one bar), let $\mathcal{N}(x)$ and $\mathcal{N}(y)$ denote the sets of notes contained in each segment, where a note is represented by its onset time and pitch (duration is not considered). The Jaccard index is defined as

$$J(x,y) = \frac{|\mathcal{N}(x) \cap \mathcal{N}(y)|}{|\mathcal{N}(x) \cup \mathcal{N}(y)|}. \tag{1}$$

This index ranges from 0 (no common notes) to 1 (identical note content). The Jaccard index thus provides a simple and interpretable measure of bar-level similarity based solely on shared onset–pitch events.

2.2 Motif (bar-level) repetition in training data

To quantify repetition patterns in the training corpus, we compute pairwise similarities between bars using the jaccard index. Let $\mathcal{B}(s) = \{b_1, b_2, \ldots, b_{n_s}\}$ denote the set of bars extracted from a song s. For each pair (b_i, b_j), we calculate $J(b_i, b_j)$ and consider b_i and b_j instances of the same motif if $J(b_i, b_j) \geq \tau$, where $\tau = 0.8$ in all experiments. Based on this rule, we construct clusters of similar bars using a union–find procedure. The repetition count of a bar b is then defined as

$$r(b) = |\mathcal{C}(b)|, \tag{2}$$

where $\mathcal{C}(b)$ denotes the cluster containing b. This results in a motif-level repetition profile for the training set, which characterizes how frequently each bar recurs within a piece.

2.3 Plagiarism-oriented metrics

To assess plagiarism-like behaviors, we employ two complementary metrics that link model behavior to motif repetition.

First, we analyze the model's likelihoods on training data. For each bar b, we compute its token-level perplexity $\text{PPL}(b)$ under the trained model. We then compare motifs, defined as bars with $r(b) \geq 5$, against non-motifs with $r(b) = 1$, by calculating a perplexity ratio

$$\text{PPLr} = \frac{\mathbb{E}_{b:r(b)\geq 5}[\text{PPL}(b)]}{\mathbb{E}_{b:r(b)=1}[\text{PPL}(b)]}. \tag{3}$$

A ratio below 1 indicates that motifs are predicted more confidently than non-motifs, suggesting local memorization.

Table 1: Comparison of plagiarism-oriented (left) and performance-oriented (right) metrics across different settings. For plagiarism metrics, higher originality indicates less plagiarism, a higher PPL-ratio, the ratio of perplexity between motifs and non-motifs, reflects reduced motif memorization, and a lower motif over-representation ratio (MORr), the relative prevalence of motifs in strongly copied samples, indicates less copying. For performance metrics, higher values correspond to better predictive accuracy on the test data.

	Plagiarism				Performance (↑)				
Method	Orig_{avg} (↑)	$\text{Orig}_{10\%}$ (↑)	PPLr (↑)	MORr (↓)	$F1_{note}$	$F1_{pr}$	GS	CS	PRS
Test data	0.665	0.472	–	0.906	1.000	1.000	1.000	1.000	1.000
Baseline	0.590	0.376	0.759	1.463	0.191	0.254	0.787	$\underline{0.570}$	0.882
+ Label smoothing	0.588	0.371	$\underline{0.763}$	1.486	$\underline{0.194}$	$\underline{0.257}$	$\underline{0.795}$	$\underline{0.570}$	$\underline{0.884}$
+ Transpose aug.	$\underline{0.608}$	$\underline{0.405}$	**0.781**	**1.182**	**0.236**	**0.299**	**0.796**	**0.594**	**0.886**
+ Top-K sampling	**0.657**	**0.484**	–	$\underline{1.262}$	0.143	0.212	0.760	0.552	0.882

Second, we examine the originality of generated outputs. From each test piece, we condition on the first 1 bar to generate continuations and randomly sample one bar from each. For a generated bar g, we define originality as

$$\text{Orig}(g) = 1 - \max_{b \in \mathcal{B}_{\text{train}}} J(g, b), \tag{4}$$

where $\mathcal{B}_{\text{train}}$ is the set of all training bars. Lower originality scores indicate closer matches to the training set and thus a higher plagiarism risk. This idea to calculate originality is based on previous approaches [6, 21, 17, 12]. We report both the overall mean originality across all sampled bars (Orig_{avg}) and the mean within the lowest-scoring 10% ($\text{Orig}_{10\%}$), which captures particularly strong copying cases. Implementation details are provided in Appendix B.

Finally, to examine whether strongly copied outputs disproportionately involve repeated motifs, let $\mathcal{B}_{\text{train}}$ be the set of all training bars, and $\mathcal{B}_{\text{copied}} \subseteq \mathcal{B}_{\text{train}}$ the subset identified as strongly copied. We then define the Motif Over-Representation ratio (MORr) as

$$\text{MORr} = \frac{E_{b \sim \mathcal{B}_{\text{copied}}}\big[\mathbf{1}\{r(b) \geq 5\}\big]}{E_{b \sim \mathcal{B}_{\text{train}}}\big[\mathbf{1}\{r(b) \geq 5\}\big]}, \tag{5}$$

where $E_{b \sim \mathcal{S}}[\mathbf{1}\{r(b) \geq 5\}]$ denotes the proportion of bars in a set $\mathcal{S}$ whose repetition count $r(b)$ is at least 5. Thus, MORr compares the relative prevalence of highly repeated motifs between the copied subset and the training corpus as a whole. Values greater than 1 indicate that repeated motifs are overrepresented among copied outputs, reinforcing the link between plagiarism risk and motif recurrence.

3 Experiments

3.1 Experimental Setup

We conducted experiments on the POP909 dataset [19] using an event-based Transformer model with relative attention [14, 9]. Musical sequences were represented in an event-based format following REMI [10], where each note was serialized into tokens of beat, position, track, pitch, and duration. Further details of dataset preprocessing, model configuration, and training procedure are provided in Appendix C.

3.2 Results

The baseline row in Table 1 summarizes the results of this setting. The model exhibits substantially lower originality scores than the test data, both in terms of average (Orig_{avg}) and the lowest 10% ($\text{Orig}_{10\%}$), indicating a strong tendency to copy training fragments. Moreover, the perplexity ratio (PPLr) is below 1.0, suggesting that the model predicts motifs more confidently than non-motifs, supporting the hypothesis of motif-level memorization. The perplexity ratio (PPLr) is below 1.0, which shows that motifs are predicted more confidently than non-motifs and supports the hypothesis of motif-level memorization. Finally, the motif over-representation ratio (MORr) exceeds 1.0. This means that motifs occur disproportionately often in strongly copied samples and further links plagiarism risk to motif recurrence.

4 Exploring Mitigation Strategies

We examine three strategies for reducing plagiarism-like behaviors: (i) label smoothing [16] with smoothing coefficient ϵ=0.1, which redistributes a fraction of probability mass from the target to non-target classes to mitigate overconfidence during training, (ii) data augmentation via transposition, where each training piece is randomly shifted by -5 to $+5$ semitones, and (iii) Top-K sampling, where generation is performed by sampling from the top-K most probable tokens instead of greedy decoding to promote diversity.

4.1 Additional performance evaluation

In addition to the plagiarism-oriented metrics introduced in Section 2.2, we also evaluated the predictive accuracy of the model. Following Inaba et al. [11], we used 15-bar excerpts from the test set as prompts and measured how closely the next predicted bar matched the ground truth. Five complementary metrics were employed: note-level F1 ($F1_{note}$) [8], pianoroll F1 ($F1_{pr}$) [8], grooving similarity (GS) [7, 20], chroma similarity (CS) [7, 20], and pitch range similarity (PRS).

4.2 Results & discussion

The results of mitigation strategies are also shown in Table 1. Label smoothing yields only marginal changes: plagiarism-oriented metrics remain close to the baseline, and performance metrics also show little improvement. In contrast, data augmentation via transposition improves originality scores and reduces motif-level memorization, as reflected in higher PPLr and lower MORr, while also achieving the best performance across all objective metrics. Top-K sampling substantially increases originality and thus suppresses plagiarism, but this comes at the cost of decreased performance according to our evaluation metrics. While these mitigation strategies demonstrate the possibility of enhancing originality without severely degrading performance, the models still exhibit strong tendencies to overfit and directly copy frequent motifs. Addressing this motif-level memorization therefore remains an essential avenue for future research.

5 Conclusion & Future Work

This paper examined plagiarism-like behaviors in symbolic music generation with Transformer-based models. Through motif-level analysis, we showed that plagiarism risk is closely linked to local overfitting: motifs repeated in the training data are predicted with lower perplexity and are disproportionately reproduced in generated outputs. We also examined the effectiveness of preliminary mitigation strategies. These findings highlight the need for motif-based perspectives when evaluating the reliability of generative music systems. Understanding the causes of plagiarism and developing effective ways to suppress it are essential steps toward ethically reliable music generation, and we hope this study provides a foundation for future research in this direction.

At the same time, our work leaves important limitations to be addressed. While we focused on bar-level repetitions as a proxy for motifs, motifs in music theory are inherently variable-length phrases, and thus our analysis does not fully capture motif-level phenomena. In addition, our evaluation of model performance relied solely on objective metrics on the test data; more diverse assessments, including subjective evaluations of the coherence and musicality of generated samples, are needed, as originality scores alone can be inflated by random note sequences.

References

[1] J. Barnett. The ethical implications of generative audio models: A systematic literature review. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society (AIES '23)*, pages 146–161, 2023.

[2] R. Batlle-Roca, W.-H. Liao, X. Serra, Y. Mitsufuji, and E. Gómez. Towards assessing data replication in music generation with music similarity metrics on raw audio. In *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*, pages 1004–1011, 2024.

[3] D. Bralios, G. Wichern, F. G. Germain, Z. Pan, S. Khurana, C. Hori, and J. L. Roux. Generation or replication: Auscultating audio latent diffusion models. In *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1156–1160, 2024.

[4] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations ICLR*, 2023.

[5] K. Chen*, Y. Wu*, H. Liu*, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov. Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[6] T. Collins, S. Böck, F. Krebs, and G. Widmer. Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In *53rd AES Conference on Semantic Audio*, 2014.

[7] S. Dixon, F. Gouyon, and G. Widmer. Towards characterisation of music via rhythmic patterns. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2004.

[8] J. P. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel. MT3: Multi-task multitrack music transcription. In *International Conference on Learning Representations ICLR*, 2022.

[9] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck. Music transformer. In *International Conference on Learning Representations ICLR*, 2019.

[10] Y.-S. Huang and Y.-H. Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *ACM International Conference on Multimedia*, page 1180–1188, 2020.

[11] T. Inaba, K. Yoshii, and E. Nakamura. On the importance of time and pitch relativity for transformer-based symbolic music generation. In *2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–6, 2024.

[12] B. Janssen, T. Collins, and I. Ren. Algorithmic ability to predict the musical future: Datasets and evaluation. In *Proceedings of the 20th International Conference on Music Information Retrieval (ISMIR)*, pages 208–215, 2019.

[13] S. Ji, X. Yang, and J. Luo. A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges. *ACM Comput. Surv.*, 56(1), 2023.

[14] P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, 2018.

[15] B. L. T. Sturm, M. Iglesias, O. Ben-Tal, M. Miron, and E. Gómez. Artificial intelligence and music: Open questions of copyright law and engineering praxis. *Arts*, 8(3), 2019.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[17] E. Ukkonen, K. Lemström, and V. Mäkinen. Geometric algorithms for transposition invariant content based music retrieval. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, 2003.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, pages 5998–6008, 2017.

[19] 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 *International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[20] S.-L. Wu and Y.-H. Yang. MuseMorphose: Full-song and fine-grained piano music style transfer with one Transformer VAE. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[21] Z. Yin, F. Reuben, S. Stepney, and T. Collins. Measuring when a music generation algorithm copies too much: The originality report, cardinality score, and symbolic fingerprinting by geometric hashing. *SN Comput. Sci.*, 3(5), June 2022.

A Related Work

Another active line of research concerns plagiarism and data replication in generative models. Symbolic and audio-based systems alike have been observed to reproduce memorized content, raising concerns about originality [21, 2, 3]. Methods such as embedding-space mixup in MuseNet and beat-synchronous mixup in MusicLDM [5] acknowledge and attempt to mitigate these risks. Similar issues have been studied extensively in language modeling, where models are known to memorize large portions of training data [4]. These observations indicate that memorization is a recurring phenomenon across domains, but the mechanisms that drive it—especially in music—remain underexplored.

These technical concerns connect with ethical debates surrounding generative audio and music systems. Questions of copyright, ownership, and fair use have been discussed from both legal and engineering perspectives [15], with recent surveys highlighting a lack of systematic attention to the societal and ethical risks of generative audio models [1]. Taken together, this body of work underscores the importance of not only mitigating plagiarism but also developing a deeper understanding of why and how it occurs, as a prerequisite for safe human–AI co-creation.

B Implementation Details of Originality

Algorithm 1 BuildIndexFromTrainingBars

```
Require: Training bars $\mathcal{B} = \{b_1, \ldots, b_M\}$, where each $b_i$ is a set of notes (each note encoded as an
    onset–pitch token).
Ensure: TrainingBarSets: $\mathsf{BarSet}[i] = S_i$ (unique note set for bar $i$);
        InvertedIndex: $\mathsf{Postings}[\text{note}] = \{\, i \mid \text{note} \in S_i \,\}$ (note $\to$ list of bar IDs).
BarSet $\leftarrow$ empty array of length $M$
Postings $\leftarrow$ empty hashmap from note to list
for $i \leftarrow 1$ to $M$ do
    $S_i \leftarrow \text{set}(b_i)$                                ▷ deduplicate to a unique note set
    $\mathsf{BarSet}[i] \leftarrow S_i$
    for all note $\in S_i$ do
        append $i$ to $\mathsf{Postings}[\text{note}]$
    end for
end for
return $(\mathsf{BarSet}, \mathsf{Postings})$
```

Algorithm 2 OriginalityScore (Jaccard-based)

```
Require: Query note set $Q \subset \mathbb{Z}$ in a bar, Candidate cap $K$
Require: InvertedIndex Postings, TrainingBarSets BarSet from Alg. 1.
Ensure: Originality score $\in [0, 1]$.
$\mathsf{votes} \leftarrow$ empty hashmap $i \mapsto 0$
for all note $\in Q$ do
    for all $i \in \mathsf{Postings}[\text{note}]$ do
        $\mathsf{votes}[i] \leftarrow \mathsf{votes}[i] + 1$
    end for
end for
$Candidates \leftarrow \textsc{TopK}(\mathsf{votes}, K)$                ▷ indices of the $K$ largest values
$\mathrm{BestJ} \leftarrow 0$
for all $i \in Candidates$ do
    $S \leftarrow \mathsf{BarSet}[i]$
    $\mathrm{Jaccard} \leftarrow |Q \cap S| / |Q \cup S|$
    if $\mathrm{Jaccard} > \mathrm{bestJ}$ then
        $\mathrm{bestJ} \leftarrow \mathrm{Jaccard}$
    end if
end for
return $1 - \mathrm{bestJ}$
```

To efficiently compare generated bars against training data, we first build an index of training bars (Algorithm 1). Each bar is converted into a unique set of onset–pitch tokens, and for every note we record the IDs of bars containing it. This yields two data structures: (i) `BarSet`, which stores the deduplicated note set of each bar, and (ii) `Postings`, an inverted index mapping each note to the list of bar IDs where it occurs.

Given a query bar, we compute its *originality score* using a Jaccard-based similarity measure (Algorithm 2). First, using the inverted index, bars that share notes with Q are retrieved and assigned `votes` proportional to the number of shared notes. Among the top-K candidates, we calculate the Jaccard index between the query set and each bar's set, and take the maximum similarity. The originality score is then defined as $1 - \max(\text{Jaccard})$, yielding values closer to 1 for more novel bars and closer to 0 for more plagiaristic ones.

C Experimental Details

C.1 Dataset

We used the POP909 dataset [19], which contains popular music with a wide variety of repetitive structures. The dataset was filtered and split at the song level into training, validation, and test sets with a ratio of 8:1:1 to avoid overlap between pieces. Each song was then segmented into 64-beat excerpts with a stride of 4 beats for data augmentation, resulting in 47,306, 5,721, and 5,855 segments for training, validation, and test, respectively.

C.2 Representation

Notes were first sorted in ascending order by beat, position, track, and pitch, and then serialized into tokens of beat, position, track, pitch, and duration. Other token types such as tempo and chord annotations were omitted.

C.3 Model and Training

We used a decoder-only Transformer [18] with relative attention [14, 9], configured with a maximum sequence length of 2048, 6 layers, 8 attention heads, an embedding dimension of 256, a dropout rate of 0.1, a batch size of 16, and a learning rate of 0.001. Validation was performed every 200 steps, and the checkpoint achieving the lowest validation loss was selected for evaluation (early stopping). During generation, the temperature was set to 0.0, yielding greedy decoding.