Fine-Grained Music Plagiarism Detection: Revealing Plagiarists through Bipartite Graph Matching and a Comprehensive Large-Scale Dataset

Wenxuan Liu
Shanghai Key Lab of Digital Media Processing and Transmission, Shanghai Jiao Tong University
Shanghai, China
wenxuanliu@sjtu.edu.cn

Tianyao He
Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai Jiao Tong University
Shanghai, China
hetianyao@sjtu.edu.cn

Chen Gong
Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai Jiao Tong University
Shanghai, China
gongchen@sjtu.edu.cn

Ning Zhang*
Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai Jiao Tong University
Shanghai, China
ningz@sjtu.edu.cn

Hua Yang*
Shanghai Key Lab of Digital Media Processing and Transmission, Shanghai Jiao Tong University
Shanghai, China
hyang@sjtu.edu.cn

Junchi Yan
Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai Jiao Tong University
Shanghai, China
yanjunchi@sjtu.edu.cn

ABSTRACT

Music plagiarism detection is gaining more and more attention due to the popularity of music production and society's emphasis on intellectual property. We aim to find fine-grained plagiarism in music pairs since conventional methods are coarse-grained and cannot match real-life scenarios. Considering that there is no sizeable dataset designed for the music plagiarism task, we establish a large-scale simulated dataset, named Music Plagiarism Detection Dataset (MPD-Set) under the guidance and expertise of researchers from national-level professional institutions in the field of music. MPD-Set considers diverse music plagiarism cases found in real life from the melodic, rhythmic, and tonal levels respectively. Further, we establish a Real-life Dataset for evaluation, where all plagiarism pairs are real cases. To detect the fine-grained plagiarism pairs effectively, we propose a graph-based method called Bipatite Melody Matching Detector (BMM-Det), which formulates the problem as a max matching problem in the bipartite graph. Experimental results on both the simulated and Real-life Datasets demonstrate that BMM-Det outperforms the existing plagiarism detection methods, and is robust to common plagiarism cases like transpositions, pitch shifts, duration variance, and melody change. Datasets and source code are open-sourced at https://github.com/xuan301/BMMDet_MPDSet.

*Corresponding authors.

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3611831

CCS CONCEPTS

• **Applied computing** → **Sound and music computing**; • **Information systems** → **Near-duplicate and plagiarism detection**.

KEYWORDS

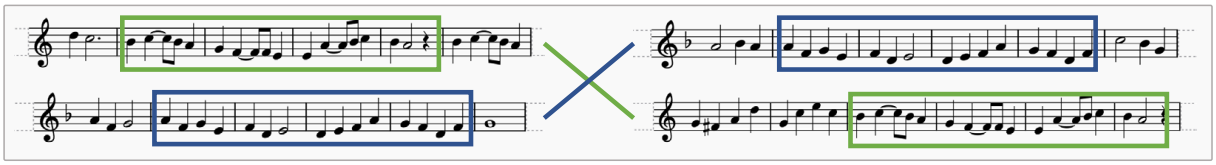
music plagiarism, bipartite graph matching, edit distance

ACM Reference Format:
Wenxuan Liu, Tianyao He, Chen Gong, Ning Zhang, Hua Yang, and Junchi Yan. 2023. Fine-Grained Music Plagiarism Detection: Revealing Plagiarists through Bipartite Graph Matching and a Comprehensive Large-Scale Dataset. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3581783.3611831

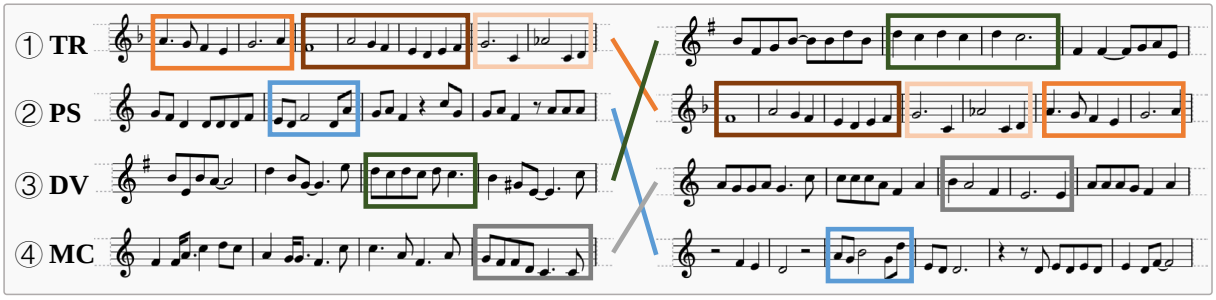
1 INTRODUCTION AND RELATED WORK

Nowadays, the number of music documents on the Internet is increasing rapidly. Each year, over 10 million new albums of recorded music are released and over 100 million new musical pieces are registered for copyright [27]. Easy access to musical content increases the risk of unintentional or intentional plagiarism. The number of lawsuits and revenue losses due to music plagiarism is soaring [8]. However, no concrete rules have seen proposed for music copyright infringement [7].

Melody plagiarism is prominent in the accusation, though sample and rhythm plagiarism are also common [11]. The melody plagiarism problem has been researched a lot over the past decades [9, 10, 21]. Several methods have been proposed to extract melody from musical mixtures [2, 4, 14, 26]. Just as there is now a commitment towards plagiarism detection and copyright protection in the field of image analysis [6, 17], similar efforts are being made in the realm of music. Existing music plagiarism detection methods can be categorized as audio-based methods and sheet-based methods. Audio-based methods inspect music pairs by comparing their time-frequency representations [3, 11, 13]. Sheet-based method measures the symbolic melodic similarity [24, 28]. However, the



(a) Conventional Course-Grained Music Plagiarism Detection



(b) Fine-Grained Music Plagiarism Detection

Figure 1: Comparison between the conventional and fine-grained music plagiarism detection is particularly studied in this paper. (a) Conventional plagiarism detection is usually coarse-grained and not optimized to identify common real-life plagiarism scenarios. (b) BMM-Det can perform fine-grained music plagiarism detection, accurately identify even when the proportion of plagiarised clips is low, and optimize the identification of plagiarism cases such as ① transposition (TR), ② pitch shifts (PS), ③ duration variance (DV), ④ melody change (MC), etc.

current methods are coarse-grained and consider little about music theory. They lack robustness and perform poorly when facing some tricky music changes like transposition, pitch shifts, duration variance, and melody change. Some high-level feature based methods like [22, 23] formulate the problem as a sequence similarity problem solved by edit distance. Other works define the plagiarism degree based on n-gram techniques such as Ukkonen measure, Sum Common measure, and TF-IDF correlation [1, 12, 20]. Nevertheless, these methods cannot perform **fine-grained** detection, which means that plagiarism only exists in a relatively small part of a musical piece, in the form of pitch shifts, duration variance, and melody change, etc.

We propose a novel model, Bipartite Melody Matching Detector (BMM-Det), for fine-grained music plagiarism detection, which can find local melody plagiarism pairs from music datasets. Compared with conventional methods, BMM-Det is able to detect finer-grained plagiarised fragments and identify hard cases like transposition, duration variance, pitch shifts, melody change, etc. The detailed difference is shown in Figure 1. BMM-Det converts two melodies into a bipartite graph and regards the corresponding maximum weight matching as their plagiarism degree. Based on music theory, we represent each melody as a sequence and regard segments of the sequence as vertices. The edge's weight between two vertices is defined by an elaborately designed distance. In our dataset constructed from real-life cases, our method surpasses the baseline plagiarism detection algorithms by a large margin.

Apart from the ineffectiveness of existing plagiarism detection algorithms, there are no large-scale datasets collected and established for music plagiarism detection. Existing datasets on music plagiarism detection like [20], Columbia Law School [10] and MIREX [1] are not public and also out-of-date. Thus, we collect real-life music plagiarism cases on our own, which have been made public for all research use. Also, there is no large-scale dataset on music plagiarism, and court decisions are scarce. Datasets like POP909 [29] contain multiple versions of the same piece of music but are designed for music generation. These multiple versions of music do not take into account any similarities in musical structure, pitch, or duration, but rather are a stylistically consistent re-creation of the original piece. Musical plagiarism is usually a fine-grained copy of another piece, but this type of dataset contains an overall variation. Therefore, we propose Music Plagiarism Detection Dataset (MPD-Set) specifically designed for music plagiarism detection, which is the first large-scale music plagiarism detection dataset and will be covered in detail in Section 2.

Our contributions are summarized as follows:

1. A fine-grained music plagiarism detection model based on bipartite graph matching named as BMM-Det is proposed. BMM-Det can cope with transposition, pitch shifts, duration variance, and melody change, etc. It can also pick out the fine-grained similar regions between two musical pieces with low global similarity.

2. Two new datasets for music plagiarism detection are published. MPD-Set is a large-scale dataset designed under the guidance of researchers from national-level professional institutions in the field of music, and we also collect a dataset consisting of real-life plagiarism pairs. These datasets address the current lack of data in this field and will facilitate research on music plagiarism detection.

3. To evaluate the performance of BMM-Det, the model parameters have been tuned using the training part of MPD-Set, and then tested directly on the testing part of MPD-Set and the whole Real-life Dataset. The experimental results indicate that BMM-Det is efficient in detecting fine-grained plagiarism and that the MPD-Set

[1] https://www.music-ir.org/mirex/wiki/2005:Symbolic_Melodic

is an effective reflection of real-life plagiarism scenarios, highlighting the importance of our research in contributing to fairness within the music industry's copyright landscape.

2 THE SIMULATED MPD-SET

The Music Plagiarism Detection Dataset (MPD-Set) is, to the best of our knowledge, the first publicly available large-scale dataset encompassing 2,000 music pieces designed for the task of music plagiarism detection. We have collaborated with researchers from Central Conservatory of Music to design the dataset under their guidance and expertise[2]. This joint effort ensures MPD-Set accurately reflects the diverse range of music plagiarism cases encountered in real life, fostering the development of more effective tools and methods for protecting intellectual property in the music industry.

The original songs utilized to create the MPD-Set have been sourced from Wikifonia[3], an open-source dataset comprised of real-life human-composed songs. To facilitate the construction of MPD-Set, we extract song fragments from Wikifonia in MusicXML format, convert them into the widely-used MIDI format within academia, and subsequently build the dataset.

The MPD-Set consists of a total of 2,000 music pieces, with each pair exhibiting a copying relationship. The most common method of plagiarism in the real world is often to select the most easily recognized elements (melody, rhythm, tonality) in music for direct replication or subtle revision. Therefore, to reflect real-world plagiarism occurrences and cover the most common plagiarism methods, we have designed four distinct types of plagiarism methods for the dataset from the melodic, rhythmic, and tonal levels respectively: transposition based on the melodic level, pitch shifts based on the tonal level, duration variance based on the rhythmic level, and melody change based on the melodic and rhythmic level, with each type accounting for 25% of the dataset. The specific implications of these four plagiarism types are as follows:

Transposition: In the transposition type of plagiarism, the original song is randomly divided into 3 to 5 segments. After the order of these segments is shuffled, they are reassembled to create a new arrangement of the piece. This process results in a modified version of the original song with the overall structure altered.

Pitch Shifts: In this scenario, a fragment of the original song undergoes pitch shifts and is subsequently added to an entirely unrelated song, creating a new piece with embedded plagiarism. Although the position of the note sequence in the melodic fragment after shift is different from the previous one, the fragment of the original song are exactly the same melodic lines as the new piece, which is actually the presence of a melody in a different tonality. This type of plagiarism is to transfer the tonality of the original song melody, which is a relatively concealed and common method of musical plagiarism.

Duration Variance: This type of plagiarism varies from the process of pitch shifts, instead of shifting the pitch of the notes, the duration of each note in the original piece is altered, modifying the original song at the rhythmic level, and the manipulated fragment is added to a completely unrelated song. It is also a relatively concealed and common method of music plagiarism.

[2]Detailed information will be included in acknowledgements
[3]https://www.wikifonia.org

Melody Change: This type is a much more sophisticated situation of plagiarism, which entails simultaneously altering the melody and rhythm of a fragment of the original song using MuseMorphose [30], a Transformer-based Variational Autoencoder (VAE) model, and integrating the transformed fragment into a completely unrelated song, resulting in a new piece containing concealed plagiarism. Concerning the melody change type of plagiarism, it is important to emphasize that, according to various national laws, determining whether a direct plagiarism relationship exists between the generated music and the original piece can be quite challenging. This difficulty arises because the altered melody and rhythm, despite being derived from the original song, may exhibit significant differences, making it hard to establish a clear connection between the two.

3 METHOD

3.1 Preliminaries: Bipartite Graph Matching

Bipartite Graph is a type of graph where the nodes can be divided into two separate sets, often referred to as "left" and "right." Edges only exist between nodes that are in different sets (i.e. a node in the "left" set can only be connected to a node in the "right" set, and vice versa). Bipartite graph matching refers to the problem of finding pairs of nodes in a bipartite graph such that each node is paired with exactly one other node, and no two pairs share an edge. In other words, we want to find a way to connect every node on the left side of the graph with a node on the right side in such a way that there are no overlaps.

In mathematical formulation, Given a bipartite graph $G = (V, E)$ consisting of two disjoint sets of vertices U and W, where $|U| = n$ and $|V| = m(n \geq m)$, and an edge set $E \subseteq U \times W$. The goal is to find the maximum cardinality matching M in G, where $M \subseteq E$. Formally, let M be a subset of edges that form a matching in G. We can represent M as a binary vector $(x_1, x_2, \ldots, x_m)$, where $x_i = 1$ if edge i belongs to M, and $x_i = 0$ otherwise. The cost of each edge is $S = (s_1, s_2, \ldots, s_m)$. Then, we can formulate it as an optimization problem:

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{i=1}^{m} s_i x_i & & \\ s.t. \quad & x_i \in \{0, 1\} & & \forall i \in \{1, 2, \ldots, m\} \\ & \textstyle\sum_{i \in N(j)} x_i \leq 1 & & \forall i \in \{1, 2, \ldots, n\} \end{aligned} \tag{1}$$

where $N(j)$ denotes the set of nodes adjacent to node j in the bipartite graph G. The first constraint ensures that every edge is either selected or not selected in the matching, and the second constraint ensures that each vertex in U is adjacent to at most one vertex in W in the matching.

3.2 Melody Sequence Representation

Before the detection algorithm, we first process the input data based on music theory. Specifically, given two musical pieces, we represent their melodies with two sequences $S_1 = \{a_1, \ldots, a_n\}$ and $S_2 = \{b_1, \ldots, b_m\}$. Every component a_i is composed of the *pitch*, *duration*, and *downbeat* of the corresponding note. The *pitch* and *duration* are denoted in Musical Instrument Digital Interface(MIDI) protocol, and the binary value *downbeat* denotes whether this note is downbeat.

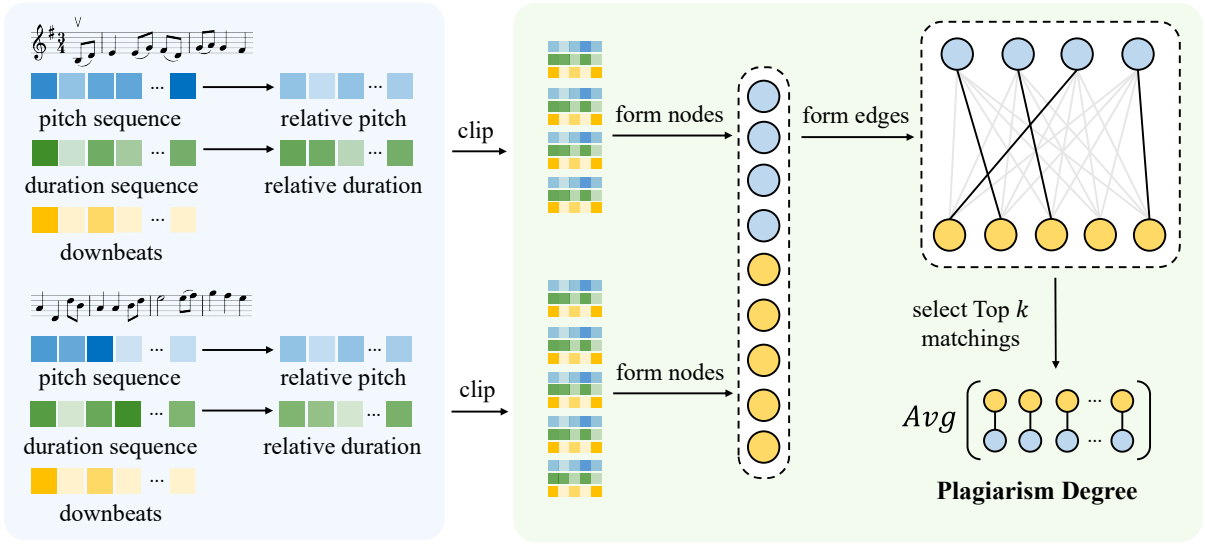


Figure 2: The pipeline of BMM-Det. Given two melodies, we convert them into two sequences, containing robust pitch, duration, and downbeat representations. Next, we cut the sequences into segments and build the bipartite graph. The maximum weight matching algorithm can be performed to get the final plagiarism degree and fine-grained matching results.

The changes of key and speed in music are commonly inspected in music plagiarism. To handle the challenges, we apply the idea of the relative sequence for both *pitch* and *duration*. The relative sequences record the pitch and duration difference between neighboring notes instead of the absolute pitch, which makes our melody representation robust to key and speed modifications.

3.3 Music Plagiarism Detection with Bipartite Graph Matching

With the melody representation, we can formulate music plagiarism detection as a bipartite graph matching problem. We establish a bipartite graph $G = (U \cup W, E)$, where U and W are two disjoint vertex sets of the bipartite graph with $|U| = n, |W| = m$, and E represents the edge set between U and W.

3.3.1 ***Vertex formation.*** We choose to get the vertices from the melody sequence by cutting with overlaps. For two melody sequences S_1 and S_2, we cut them into clips with length l and overlapping rate r to obtain clip lists P_1 and P_2. Each clip in P_1 is considered as a vertex in U and each clip in P_2 is a vertex in W.

3.3.2 ***Edge formation.*** For each vertex $u \in U$ and $w \in W$, we construct an edge $e = (u, w)$. In music plagiarism, high melody similarity is a determinant in the final judgment, while low similarity is acceptable. Therefore, we want our edge cost to satisfy two properties. **First**, the edge cost should reflect the similarity of two melody clips. **Second**, the edge cost should amplify the high similarity and suppress the low similarity, making it more sensitive to plagiarism.

To satisfy the first property, we use the edit distance to reflect the melody similarity. The edit distance between two sequences aims to calculate the minimum operation costs to change one sequence to the other with pre-defined operations. The three basic operations on melody are substitution, insertion, and deletion, whose costs c_{sub}, c_{ins} and c_{del} are detailed in section 3.4.

The edit distance can be calculated with the idea of dynamic programming. We define a two-dimensional dynamic table d, where $d_{i,j}$ records the distance between the first i notes of u and the first j notes of v. Therefore, $d_{l,l}$ means the complete edit distance between u and w. We can define the boundary conditions:

$$d_{i,0} = \sum_{k=1}^{i} c_{\text{del}}(u_k), 1 \le i \le l \tag{2}$$

$$d_{0,j} = \sum_{k=1}^{j} c_{\text{ins}}(w_k), 1 \le j \le l \tag{3}$$

where u_k means the k_{th} note of vertex u (similar for w). With the boundary conditions, we can define the update function under different circumstances.

For $u_i = w_j$, we have:

$$d_{i,j} = d_{i-1,j-1} \tag{4}$$

For $u_i \neq w_j$, we have:

$$d_{i,j} = \min(d_{i,j}^{\text{del}}, d_{i,j}^{\text{ins}}, d_{i,j}^{\text{sub}}) \tag{5}$$

where

$$\begin{aligned} d_{i,j}^{\text{del}} &= d_{i-1,j} + c_{\text{del}}(u_i) \\ d_{i,j}^{\text{ins}} &= d_{i,j-1} + c_{\text{ins}}(w_j) \\ d_{i,j}^{\text{sub}} &= d_{i-1,j-1} + c_{\text{sub}}(u_i, w_j) \end{aligned} \tag{6}$$

Now that we have obtained the edge weight between two nodes, we aim to redesign it to better suit the needs of music plagiarism detection. We observe that in music plagiarism, the local high similarity of two short clips often contributes more to the judgment, while low similarity is acceptable in music creation. To achieve this, we want to satisfy two properties for our edge weight:

1) Amplify high similarity and suppress low similarity.

2) Ensure the edge weight falls within the range of $[0, 1]$.

To satisfy the second property, we use the function described in Equation 7 to transform the original edit distance into our required edge weight. The logarithm calculation in Equation 7 is used to amplify the differences between smaller values and suppress those between larger values.

$$f(d) = \frac{\ln(1 + e^{-d})}{\ln 2} \tag{7}$$

3.3.3 ***Music Plagiarism Degree.*** With the bipartite graph established, we can now analyze the comprehensive melody similarity between two music pieces with fine-grained awareness. To obtain the minimum cost matching of the bipartite graph G, we employ the Kuhn–Munkres (KM) algorithm [16]. The KM algorithm, also known as the Hungarian algorithm, is a combinatorial optimization algorithm that efficiently solves the assignment problem by computing the minimum weight of matching in a weighted bipartite graph.

Let $G = (U \cup W, E)$ be a weighted bipartite graph, where U and W are the sets of vertices on the two sides of the graph, and E is the set of edges connecting the vertices. Each edge $(u, w) \in E$ has an associated non-negative weight $t(u, w)$. The goal of the KM algorithm is to find a perfect matching M of the bipartite graph that minimizes the total weight:

$$M = \arg\min_{M} \sum_{(u,w)\in M} t(u, w). \tag{8}$$

The KM algorithm finds the minimum weight perfect matching by iteratively updating a set of labels, one for each vertex in U and W. The algorithm starts with an initial feasible labeling and a partial matching. It then searches for augmenting paths, which are alternating paths that start and end with unmatched vertices, while respecting the labeling constraints. If an augmenting path is found, the algorithm increases the size of the matching by flipping the matched and unmatched edges along the path. The process repeats until no more augmenting paths can be found, at which point the algorithm has found a minimum weight perfect matching.

By applying the KM algorithm, we can determine the minimum matching scores, which represent the plagiarism degree between the two music pieces. Furthermore, we can locate the plagiarized parts by examining the matched vertex pairs in the bipartite graph. This approach allows for a detailed analysis of the similarities between the two music pieces, identifying and quantifying the extent of plagiarism present in the compositions.

3.4 Score Design of Edit Operations

In this section, we define the scores of substitution, insertion, and deletion. For insertion and deletion, we set their costs to $c_{\text{ins}} = c_{\text{del}} = 1$. For substitution, the computation is more complicated. The common knowledge is that substitution with larger pitch or duration differences will have a greater influence on the music, especially the downbeat notes. Therefore, we need a comprehensive design of the substitution cost to reflect its real impact on the music.

Our substitution cost function is Equation 9:

$$c_{\text{sub}}(u_i, w_j) = c_{\text{downbeat}}(u_i, w_j) \cdot [c_{\text{pitch}}(u_i, w_j) + c_{\text{duration}}(u_i, w_j)], \tag{9}$$

where c_{downbeat} is the downbeat coefficient, c_{pitch} is the pitch difference cost, and c_{duration} is the duration difference cost. c_{pitch} and c_{duration} are defined as:

$$c_{\text{pitch}}(u_i, w_j) = \left| u_i^{\text{pitch}} - w_j^{\text{pitch}} \right| \tag{10}$$

$$c_{\text{duration}}(u_i, w_j) = \left| u_i^{\text{duration}} - w_j^{\text{duration}} \right| \tag{11}$$

where the superscript represents that it is the pitch or duration of the note. Downbeat coefficient c_{downbeat} is defined in Equation 12:

$$c_{\text{downbeat}}(u_i, w_j) = k_{\text{down}} \cdot u_i^{\text{downbeat}} \cdot w_j^{\text{downbeat}} \tag{12}$$

where $u_i^{\text{downbeat}} \in \{0, 1\}$ is the binary downbeat indicator representing whether the i_{th} note of u is a downbeat(similar for v). k_{down} is a hyperparameter controlling the importance of the downbeat.

4 EXPERIMENTS

To demonstrate the effectiveness of our MPD-Set and BMM-Det for fine-grained plagiarism detection, we opt to tune BMM-Det using the MPD-Set and evaluate its performance on two test sets. The first test set is a subset of the MPD-Set, and the second test set consists of real-life plagiarism cases (Real-life Dataset). This approach allows us to assess the algorithm's ability to generalize across different types of data, showing the robustness of our method.

4.1 Datasets

We use two datasets for the experiment. The first is MDP-Set, which has been introduced in Section 2. The second real-life one is collected by us and consists of 29 pairs of songs, where 20 pairs are legally judged as plagiarism by the court [5], and the other 9 pairs of songs are from Ping An Tech's work[4], and all these 29 pairs of songs constitute plagiarism.

4.2 Implementation Details

Experiment Setting: Our primary objective is to assess the capability of our algorithm to detect music plagiarism and locate plagiarized parts, especially in real-world situations. Due to the limited availability of real-life music plagiarism legal cases, we strive to significantly expand our test sets to showcase the robustness and adaptability of our model. To achieve this, we first divide the MPD-Set into an 80-20 proportion. We use 80% of the MPD-Set, consisting of approximately 1600 songs, to tune the BMM-Det algorithm. The remaining 20% of the MPD-Set, comprising approximately 400 songs, serves as the first test set. In addition, we utilize a second test set consisting of real-life plagiarism cases to further evaluate our model's performance in practical scenarios.

By tuning the algorithm on the majority of the MPD-Set and striving to achieve impressive results on the Real-life Dataset, we aim not only to demonstrate the effectiveness of our BMM-Det model in capturing and reflecting the characteristics of plagiarism present in real-world scenarios, but also to highlight the importance of the MPD-Set as a reliable representation of real-life plagiarism cases. Through this approach, we intend to evaluate the algorithm's ability to adapt and generalize across various types of data, while ensuring a comprehensive assessment of its performance in detecting music plagiarism.

Evaluation Metrics: To measure the performance of our algorithm, we mainly focus on two indicators: the Average Ranking Index (ARI) of the plagiarized songs and the accuracy.

[4]https://github.com/andyjhj/MusicPlag_Demo

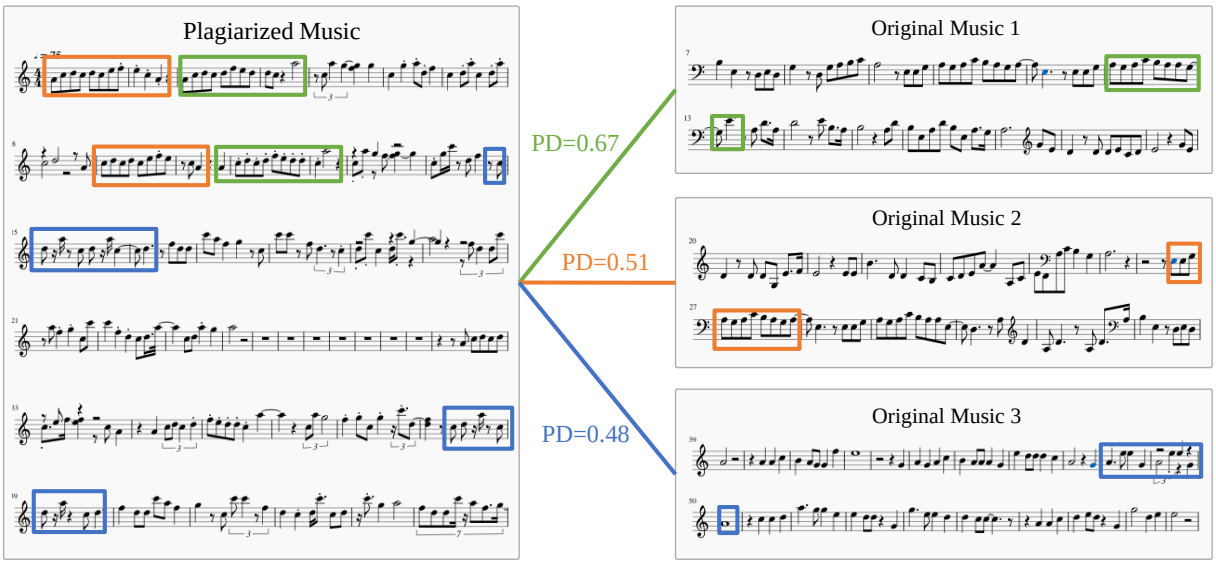


Figure 3: An example of fine-grained plagiarism detection using BMM-Det. The left song has potential plagiarism compared to the three original songs on the right. Musical pieces of the same colour are detected as plagiarised pairs with high plagiarism scores (e.g., 0.67, 0.51, 0.48), indicating a high degree of similarity between the compared sections.

1. **Average Ranking Index (ARI)**: The ARI measures the average ranking of the plagiarized song within a dataset when compared to the remaining original songs. The lower the ARI, the better the performance of the algorithm. Mathematically, it is defined as:

$$ARI = \frac{1}{N}\sum_{i=1}^{N} R_i, \tag{13}$$

where N is the number of song pairs in the test set and R_i is the ranking of the plagiarized song for the i-th test case.

2. **Accuracy**: Accuracy measures the proportion of correctly identified plagiarized songs in the test set. The higher the accuracy, the better the performance of the algorithm. It is defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Identifications}}{\text{Total Number of Test Cases}} \times 100\%. \tag{14}$$

In our experiment, a result is considered correct if the plagiarized song is ranked first when compared to the other songs in the dataset. By evaluating these two metrics, we can obtain a comprehensive understanding of our algorithm's performance in detecting music plagiarism.

4.3 Results and Analysis

On Testing Dataset. We compare BMM-Det with other existing methods, including Sum Common [19], Ukkonen [19], TF-IDF correlation [15], and Tversky-equal [25], on both the MPD-Set and the Real-life Dataset. Ukkonen and Sum Common consider the difference of all n-gram features occurring in either one musical piece [19]. TF-IDF correlation method is widely used for retrieving text documents [15]. In our experiment, we use n-gram features weighted by their frequency in both musical pieces and our dataset, which is measured by inverted document frequency [18]: $\text{IDF}(\tau) = \log(\frac{n}{n_\tau})$ where n is the size of the dataset and n_τ is the number of pieces including τ. Tversky's ratio model originates from [25], which is adapted by inserting IDF.

Table 1: Comparison of ARI and accuracy between different methods on the MPD-Set and the Real-life Dataset.

Method	MPD-Set		Real-life Dataset	
	ARI	Acc.	ARI	Acc.
TF-IDF corr.[15]	10.52	0.695	3.24	0.655
Tversky-equal[25]	16.37	0.685	3.14	0.655
Sum Common[19]	12.00	0.653	3.10	0.724
Ukkonen[19]	10.08	0.675	3.00	0.759
BMM-Det (Ours)	**7.07**	**0.705**	**2.17**	**0.828**

The results are shown in Table 1. For each baseline, we tune the hyperparameter n. It is evident that BMM-Det outperforms all the baselines on both datasets. This indicates that methods based on n-gram features work only when the frequency of common n-gram features is related to the overall plagiarism degree. Upon further analysis of the results in the table, we can observe the following:

1. On the MPD-Set, BMM-Det (our method) outperforms other methods with an ARI of 7.07 and an accuracy of 0.705. This is a significant improvement compared to the second-best method, Ukkonen, which has an ARI of 10.08 and an accuracy of 0.675. This demonstrates the effectiveness of BMM-Det in detecting music plagiarism in the MPD-Set.

2. On the Real-life Dataset, BMM-Det also achieves the best performance with an ARI of 2.17 and an accuracy of 0.828. The second-best method, Ukkonen, has an ARI of 3.00 and an accuracy of 0.759. The superior performance of BMM-Det on the Real-life Dataset further confirms its capability to adapt and generalize across various types of data, including real-life plagiarism cases.

3. Comparing the results between MPD-Set and the Real-life Dataset, BMM-Det maintains consistently high performance in both datasets. This highlights the importance of the MPD-Set as a reliable representation of real-life plagiarism cases and validates the effectiveness of our algorithm in capturing and reflecting the characteristics of plagiarism present in real-world scenarios. The

performance of BMM-Det can be attributed to its ability to adapt and generalize across different types of data, making it a more robust and reliable method for detecting music plagiarism.

In conclusion, the results of our experiment demonstrate the effectiveness of the BMM-Det algorithm in detecting music plagiarism in various testing datasets, including real-life cases. Its superior performance over other methods, both in terms of the ARI and accuracy, highlights its potential for practical applications in the field of music plagiarism detection.

Ablation Study: In our experiment, we conduct ablation studies to test the impact of different combinations of optimization methods for plagiarism degree measurement on the MPD-Set and the Real-life Dataset. The tested optimization methods include RelativeDuration, RelativePitch, MaxMatch, DownBeat, and NoteDistance, where RelativeDuration means using relative duration sequence, RelativePitch means using relative pitch sequence, MaxMatch means segmenting the sequence and computing the maximum weight matching, Downbeat means considering whether the note is downbeat, and NoteDistance means considering the shifts of the pitch when computing the edit distance. The outcomes of these studies are detailed in Table 2, which is sorted by the ARI obtained on the MPD-Set.

From the table, we can draw the following observations:

1. MaxMatch plays a pivotal role in improving the algorithm's performance. This is evident when comparing Row 1 (where MaxMatch is absent) and Row 6 (where it is present). The ARI decreases, and the Accuracy significantly increases when MaxMatch is applied, underscoring its importance in enhancing the algorithm's ability to detect plagiarism.

2. The contributions of each individual module to the model's performance are generally positive. Our model's components are relatively decoupled, which allows us to analyze the role of each part from the results displayed here. For instance, the impact of NoteDistance can be observed from Rows 6, 7, 8, and 9, while the effect of RelativeDuration can be deduced from Rows 4, 5, 8, and 9. Similar trends can be observed for other components. Among these, the addition of the DownBeat module has a relatively minor impact on performance compared to the other modules.

3. The simultaneous use of RelativeDuration, RelativePitch, and MaxMatch results in a substantial enhancement of the model's performance (see Row 6). This performance improvement signals the synergistic effect of these optimization methods when combined. Further augmenting this combination with DownBeat and NoteDistance leads to an even greater improvement, as the ARI decreases further and the Accuracy rises (refer to Row 9). This trend shows the compounded effect of integrating all these compositions, revealing their collective significance in the plagiarism detection task.

In summary, these results emphasize the individual contributions of each optimization method, as well as their synergistic effects when used together. By considering multiple musical features, our algorithm becomes more robust, leading to improved performance on both MPD-Set and Real-life Datasets. This indicates that our BMM-Det algorithm effectively captures and reflects the characteristics of plagiarism present in real-world scenarios and highlights the importance of the MPD-Set as a reliable representation of real-life plagiarism cases.

Considering Different Types of Plagiarism. In this experiment, our primary objective is to assess the effectiveness of various music plagiarism detection methods, including our proposed BMM-Det algorithm, across different types of plagiarism. The MPD-Set takes into account a comprehensive range of real-world plagiarism scenarios, making it an ideal choice for evaluating the performance of these methods. The dataset encompasses four distinct types of plagiarism: transposition, pitch shifts, duration variance, and melody change, with each type constituting 25% of the dataset. By evaluating the performance of different methods in detecting music plagiarism across these four types, we aim to gain a deeper understanding of their effectiveness in various real-life plagiarism scenarios. The experimental results, presented in Table 3, show the performance of BMM-Det in different categories.

1. In the Transposition category, all methods achieve an ARI of 1.00 and an accuracy of 1.00. This indicates that plagiarism involving transposition is relatively easy to detect and locate, as all methods perform equally well in this category.

2. For both Pitch Shifts and Duration Variance categories, BMM-Det surpasses other methods. Specifically, in the Pitch Shifts category, BMM-Det garners an optimal accuracy of 0.98 and an ARI of 1.02. In the Duration Variance category, BMM-Det sustains high performance with an ARI of 1.00 and accuracy of 1.00. These outcomes emphasize BMM-Det's superior capability in identifying music plagiarism involving pitch shifts and duration variance.

3. In the Melody Change category, BMM-Det significantly outperforms other methods, achieving an ARI of 6.89 and an accuracy of 0.33. Although the accuracy is lower compared to other categories of plagiarism, it still demonstrates the potential of BMM-Det in detecting music plagiarism involving melody alterations. The lower accuracy may be attributed to the difficulty in establishing a clear connection between the generated music and the original piece due to significant differences in the altered melody.

Identifying Fine-grained Plagiarized Parts: BMM-Det is capable of identifying fine-grained plagiarized sections in music. Figure 3 displays qualitative results where a song is compared with others for potential plagiarism. The same color in the figure denotes the fine-grained plagiarized parts detected by BMM-Det. Upon listening to these parts, we find they exhibit a high degree of auditory similarity. Despite significant differences in structure, duration, and pitch between these pairs, BMM-Det can successfully detect the plagiarized sections, highlighting our method's robust capabilities.

In our approach, we process and compare potential plagiarized and original songs, calculating the edit distance using the BMM-Det algorithm. By analyzing these distances, we obtain the plagiarism scores (e.g., 0.67, 0.51, 0.48), which are derived from the maximum weight matching of segmented sequences in the compared songs. These scores indicate the degree of similarity between the compared song sections, serving as a valuable indicator to identify and assess the fine-grained plagiarized parts in the analyzed music. This approach demonstrates the effectiveness of BMM-Det in detecting music plagiarism at a detailed level, as it effectively captures the characteristics of plagiarism present in real-world scenarios.

It is crucial to note that the determination of music plagiarism ultimately falls within the realm of legal judgment. Our BMM-Det algorithm serves as an auxiliary tool for music plagiarism detection. Users can decide on the plagiarism threshold according

Table 2: Ablation studies on different compositions of optimization methods for plagiarism degree measurement on MPD-Set and the Real-life Dataset. Here RelativeDuration means using relative duration sequence, RelativePitch means using relative pitch sequence, MaxMatch means segmenting the sequence and computing maximum weight matching, DownBeat means considering whether the note is downbeat, and NoteDistance means considering pitch shifts when computing edit distance.

RelativeDuration	RelativePitch	MaxMatch	DownBeat	NoteDistance	MPD-Set		Real-life Dataset	
					ARI	Acc.	ARI	Acc.
✓	✓				36.32	0.365	3.24	0.482
✓		✓			32.26	0.435	5.00	0.586
✓		✓	✓		32.04	0.445	5.17	0.586
	✓	✓	✓	✓	10.65	0.650	2.28	0.690
	✓	✓		✓	10.01	0.665	2.31	0.689
✓	✓	✓			9.35	0.635	2.93	0.689
✓	✓	✓	✓		9.00	0.630	2.79	0.758
✓	✓	✓		✓	7.07	**0.705**	2.17	**0.828**
✓	✓	✓	✓	✓	**6.98**	0.700	**2.14**	**0.828**

Table 3: Comparison of Music Plagiarism Detection Methods' Performance Across Different Plagiarism Types Using MPD-Set.

Method	Transpostion		Pitch Shifts		Duration Variance		Melody Change	
	ARI	Acc.	ARI	Acc.	ARI	Acc.	ARI	Acc.
TF-IDF corr. [15]	1.00	1.00	1.16	0.92	1.12	0.94	14.54	0.08
Tversky-equal [25]	1.00	1.00	1.34	0.98	1.06	0.96	13.92	0.14
Sum Common [19]	1.00	1.00	2.16	0.82	1.14	0.90	17.02	0.04
Ukkonen [19]	1.00	1.00	2.06	0.86	1.20	0.90	12.44	0.06
BMM-Det (Ours)	**1.00**	**1.00**	**1.02**	**0.98**	**1.00**	**1.00**	**6.89**	**0.33**

to their needs. Once the threshold is determined, users can delve into the results provided by BMM-Det to examine the detected plagiarized sections at a fine-grained level. This flexibility ensures that BMM-Det can cater to different requirements and preferences while effectively identifying music plagiarism.

5 CONCLUSION AND DISCUSSION

Music plagiarism is a widespread issue in the music industry and has become increasingly difficult to detect due to the use of digital tools and the prevalence of online music distribution platforms. To tackle this problem, we propose BMM-Det that can effectively detect fine-grained music plagiarism across different datasets. BMM-Det is based on a bipartite graph and is robust to various forms of musical manipulation that are often used to disguise plagiarized music, such as transposition, duration variance, pitch shifts, and melody change. To evaluate the effectiveness of BMM-Det, we create a simulated large-scale dataset called MPD-Set, incorporating a range of different types of plagiarism generated using a specialized method designed to replicate real-world examples of music plagiarism. Furthermore, we collect a Real-life Dataset encompassing numerous real-life cases of music plagiarism. Experimental results on both MPD-Set and the Real-life Dataset demonstrate BMM-Det's outstanding performance in detecting fine-grained plagiarism. By developing BMM-Det for identifying fine-grained plagiarism and creating MPD-Set that accurately represents real-world plagiarism scenarios, our work contributes to promoting fairness and transparency in the music industry.

We have made efforts to design our method as a transparent "white-box" approach, but we acknowledge that the final determination of music plagiarism is ultimately a legal matter. Therefore, we envision our method as a valuable auxiliary tool for music industry professionals to use for self-evaluation or detecting plagiarism in others' works. BMM-Det can provide detailed information on the fine-grained plagiarized segments and the degree of plagiarism, and its detection process is theoretically well-grounded, making it highly useful for related investigations.

For future work, we plan to improve BMM-Det by considering additional auditory features to enhance its accuracy and adaptability. Furthermore, we aim to expand our dataset to facilitate ongoing research in this field. We also encourage the research community to join our efforts in continually refining and expanding the dataset. Collaborative contributions will help in creating a more robust and comprehensive dataset that accurately reflects the diverse range of music plagiarism cases encountered in real life.

ACKNOWLEDGMENTS

We would like to extend our gratitude to Prof. Zijin Li, Zhaorui Liu from Central Conservatory of Music, for their contributions to the design of the dataset used in this study.

This research received partial support from the National Natural Science Foundation of China (NSFC, Grant No. 62171281) and the Science and Technology Commission of Shanghai Municipality (STCSM, Grant Nos. 20DZ1200203, 2021SHZDZX0102, 22DZ2229005, 22ZR1434900).

REFERENCES

[1] David Bainbridge, Michael Dewsnip, and Ian H Witten. 2005. Searching digital music libraries. *Information processing & management* 41, 1 (2005), 41–56.

[2] Rachel M. Bittner, Avery Wang, and Juan P. Bello. 2017. Pitch contour tracking in music using Harmonic Locked Loops. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 191–195. https://doi.org/10.1109/ICASSP.2017.7952144

[3] Neetish Borkar, Shubhra Patre, Raunak Singh Khalsa, Rohanshhi Kawale, and Priti Chakurkar. 2021. Music Plagiarism Detection using Audio Fingerprinting and Segment Matching. In *2021 Smart Technologies, Communication and Robotics (STCR)*. IEEE, 1–4.

[4] Pritish Chandna, Merlijn Blaauw, Jordi Bonada, and Emilia Gómez. 2020. Content Based Singing Voice Extraction from a Musical Mixture. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 781–785. https://doi.org/10.1109/ICASSP40776.2020.9053024

[5] C Cronin. 2016. Columbia Law School & UCLA Law Copyright Infringement Project. *update* (2016).

[6] Shenglan Cui, Fang Liu, Tongqing Zhou, and Mohan Zhang. 2022. Understanding and Identifying Artwork Plagiarism with the Wisdom of Designers: A Case Study on Poster Artworks. In *Proceedings of the 30th ACM International Conference on Multimedia*. 1117–1127.

[7] Roberto De Prisco, Antonio Esposito, Nicola Lettieri, Delfina Malandrino, Donato Pirozzi, Gianluca Zaccagnino, and Rocco Zaccagnino. 2017. Music plagiarism at a glance: metrics of similarity and visualizations. In *2017 21st International Conference Information Visualisation (IV)*. IEEE, 410–415.

[8] Roberto De Prisco, Nicola Lettieri, Delfina Malandrino, Donato Pirozzi, Gianluca Zaccagnino, and Rocco Zaccagnino. 2016. Visualization of music plagiarism: Analysis and evaluation. In *2016 20th International Conference Information Visualisation (IV)*. IEEE, 177–182.

[9] Roberto De Prisco, Delfina Malandrino, Gianluca Zaccagnino, and Rocco Zaccagnino. 2017. A computational intelligence text-based detection system of music plagiarism. In *2017 4th International Conference on Systems and Informatics (ICSAI)*. IEEE, 519–524.

[10] Roberto De Prisco, Delfina Malandrino, Gianluca Zaccagnino, and Rocco Zaccagnino. 2017. Fuzzy vectorial-based similarity detection of music plagiarism. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 1–6.

[11] Christian Dittmar, Kay F Hildebrand, Daniel Gärtner, Manuel Winges, Florian Müller, and Patrick Aichroth. 2012. Audio forensics meets music information retrieval—a toolbox for inspection of music plagiarism. In *2012 Proceedings of the 20th European signal processing conference (EUSIPCO)*. IEEE, 1249–1253.

[12] Shyamala Doraisamy and Stefan Rüger. 2003. Robust polyphonic music retrieval with n-grams. *Journal of Intelligent Information Systems* 21, 1 (2003), 53–70.

[13] J Stephen Downie, Mert Bay, Andreas F Ehmann, and M Cameron Jones. 2008. Audio Cover Song Identification: MIREX 2006-2007 Results and Analyses.. In *ISMIR*. 468–474.

[14] Xingjian Du, Bilei Zhu, Qiuqiang Kong, and Zejun Ma. 2021. Singing Melody Extraction from Polyphonic Music based on Spectral Correlation Modeling. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 241–245. https://doi.org/10.1109/ICASSP39728.2021.9414190

[15] Daniel Jurafsky and James H Martin. [n. d.]. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.

[16] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.

[17] Yuanjing Luo, Tongqing Zhou, Fang Liu, and Zhiping Cai. 2023. IRWArt: Levering Watermarking Performance for Protecting High-quality Artwork Images. In *Proceedings of the ACM Web Conference 2023*. 2340–2348.

[18] C. D. Manning and H Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Foundations of Statistical Natural Language Processing.

[19] Daniel Müllensiefen, Klaus Frieler, et al. 2004. Cognitive adequacy in the measurement of melodic similarity: Algorithmic vs. human judgments. *Computing in Musicology* 13, 2003 (2004), 147–176.

[20] Daniel Müllensiefen and Marc Pendzich. 2009. Court decisions on music plagiarism and the predictive value of similarity algorithms. *Musicae Scientiae* 13, 1_suppl (2009), 257–295. https://doi.org/10.1177/102986490901300111 arXiv:https://doi.org/10.1177/102986490901300111

[21] François Pachet, Jean-Julien Aucouturier, Amaury La Burthe, Aymeric Zils, and Anthony Beurive. 2006. The cuidado music browser: an end-to-end electronic music distribution system. *Multimedia Tools and Applications* 30, 3 (2006), 331–349.

[22] Matthias Robine, Pierre Hanna, and Pascal Ferraro. 2007. Music Similarity: Improvements of Edit-Based Algorithms by Considering Music Theory *(MIR '07)*. Association for Computing Machinery, New York, NY, USA, 135–142. https://doi.org/10.1145/1290082.1290103

[23] Matthias Robine, Pierre Hanna, Pascal Ferraro, and Julien Allali. 2007. Adaptation of string matching algorithms for identification of near-duplicate music documents. In *Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection (PAN07)*. 37–43.

[24] Mu-Syuan Sie, Cheng-Chin Chiang, Hsiu-Chun Yang, and Yi-Le Liu. 2017. DETECTING AND LOCATING PLAGIARISM OF MUSIC MELODIES BY PATH EXPLORATION OVER ABinary MASK. In *CS & IT Conference Proceedings*, Vol. 7. CS & IT Conference Proceedings.

[25] A. Tversky. 1988. Features of similarity. *Readings in Cognitive Science* 84, 4 (1988), 290–302.

[26] Stefan Uhlich, Franck Giron, and Yuki Mitsufuji. 2015. Deep neural network based instrument extraction from music. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2135–2139. https://doi.org/10.1109/ICASSP.2015.7178348

[27] Alexandra Uitdenbogerd and Justin Zobel. 1999. Melodic matching techniques for large music databases. *Proceedings of the ACM International Multimedia Conference & Exhibition*, 57–66. https://doi.org/10.1145/319463.319470

[28] Valerio Velardo, Mauro Vallati, and Steven Jan. 2016. Symbolic melodic similarity: State of the art and future challenges. *Computer Music Journal* 40, 2 (2016), 70–83.

[29] Ziyu Wang, Ke Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, Xianbin Gu, and Gus Xia. 2020. Pop909: A pop-song dataset for music arrangement generation. *arXiv preprint arXiv:2008.07142* (2020).

[30] Shih-Lun Wu and Yi-Hsuan Yang. 2021. MuseMorphose: Full-Song and Fine-Grained Music Style Transfer with One Transformer VAE. *arXiv preprint arXiv:2105.04090* (2021).

A PERFORMANCE UNDER VARIOUS CONDITIONS

In our research, we further conduct a series of experiments concerning the overlapping rate and piece length. These two parameters are scrutinized to assess their impact on the performance of the proposed algorithm. The performance criteria taken into account are the average ranking index, accuracy, and normalized time, with the latter being an indicator of the relative execution time as compared to a predefined benchmark scenario.

The benchmark for the piece length is established at 7, whereas that for the overlapping rate is defined as 0.2. Our investigations unveil that the algorithm demonstrates optimal performance with a piece length of 7 and an overlapping rate of 0.2, signifying a profound interplay between these parameters and the efficacy of the algorithm.

Table 4: Performance on the Simulated MPD-SET Under Different Piece Lengths

Piece Length	ARI	Accuracy	Normalized Time
3	25.83	0.270	1.06
5	17.99	0.540	1.34
7	**6.98**	**0.700**	1.00
9	21.29	0.495	**0.89**
11	24.10	0.525	0.97

Table 5: Performance on the Real-World Dataset Under Different Piece Lengths

Piece Length	ARI	Accuracy	Normalized Time
3	4.17	0.345	1.11
5	3.31	0.552	1.27
7	**2.14**	**0.828**	1.00
9	3.69	0.586	**0.91**
11	6.00	0.483	1.01

A.1 Performance Under Different Piece Lengths

We first examine the performance of our algorithm under varying piece lengths, as shown in Tables 4 and 5. From the results, it is evident that the optimal performance is achieved when the piece length is set to 7. Both shorter and longer piece lengths negatively impact performance. This is aligned with our understanding of music, where a length of 7 is appropriate for detecting plagiarism. Therefore, we have set the piece length to 7 in this study.

A.2 Performance Under Different Overlapping Rates

We also investigate the performance under varying overlapping rates. As demonstrated in Tables 6 and 7, time complexity escalates significantly with the increase in overlapping rates due to the augmented number of pieces correlated with the rate. However, performance improvement is not directly proportional to the overlapping rate. While a higher overlapping rate allows for more detailed analysis, it also exacerbates the complexity of the problem by increasing the number of pieces. Therefore, after considering the trade-off between performance and time complexity, we have determined the final overlapping rate to be 0.2 in our study.

Table 6: Performance on the Simulated MPD-SET Under Different Overlapping Rates

Overlapping Rate	ARI	Accuracy	Normalized Time
0	19.51	0.495	**0.75**
0.2	**6.98**	0.700	1.00
0.4	15.98	0.615	1.42
0.6	8.72	**0.720**	3.84

Table 7: Performance on the Real-World Dataset Under Different Overlapping Rates

Overlapping Rate	ARI	Accuracy	Normalized Time
0	3.41	0.586	**0.75**
0.2	**2.14**	**0.828**	1.00
0.4	2.86	0.621	1.44
0.6	3.13	0.551	3.90