APSIPA Transactions on Signal and Information Processing, 2025, 14, e303

Original Paper

Music Similarity Representation Learning Focusing on Individual Instruments with Source Separation and Human Preference

Takehiro Imamura*, Yuka Hashizume, Wen-Chin Huang and Tomoki Toda

Nagoya University, Aichi, Japan

ABSTRACT

This paper proposes music similarity representation learning (MSRL) based on individual instruments (InMSRL) utilizing music source separation (MSS) and human preference without requiring clean instrument stems during inference. We propose three methods that effectively improve performance. First, we introduce end-to-end fine-tuning (E2E-FT) for the *Cascade* approach that sequentially performs MSS and music similarity feature extraction. E2E-FT allows the model to minimize the adverse effects of a separation error on the feature extraction. Second, we propose multi-task learning for the *Direct* approach that directly extracts disentangled music similarity features using a single music similarity feature extractor. Multi-task learning, which is based on the disentangled music similarity feature extraction and MSS based on reconstruction with disentangled music similarity features, further enhances instrument feature disentanglement. Third, we employ perception-aware fine-tuning (PAFT). PAFT utilizes human preference, allowing the model to perform InMSRL aligned with

*Corresponding author: imamura.takehiro@g.sp.m.is.nagoya-u.ac.jp. This work was partly supported by JST CREST JPMJCR19A3 and JST AIP Acceleration Research JPMJCR25U5, Japan.

Received 15 March 2025; revised 10 July 2025; accepted 23 July 2025

ISSN 2048-7703; DOI 10.1561/116.20250016

human perceptual similarity. We conduct experimental evaluations and demonstrate that 1) E2E-FT for *Cascade* significantly improves objective InMSRL performance, 2) the multi-task learning for *Direct* is also helpful to improve disentanglement performance in the feature extraction, 3) PAFT significantly enhances the perceptual InMSRL performance, and 4) *Cascade* with E2E-FT and PAFT outperforms *Direct* with the multi-task learning and PAFT.

Keywords: Music information retrieval, music similarity representation, music source separation

1 Introduction

Recently, the number of musical pieces available online has already exceeded 1 billion[1] and further market expansion is expected.[2] Therefore, the demand for music recommendation and retrieval systems has been increasing. Methods utilizing listening histories of users [3, 50] have been widely used in these systems although these methods cause several limitations, for example, it is hard to handle musical pieces with fewer listening records. One approach to avoid this problem is to extract content features from a musical piece and utilize them for music recommendation and retrieval. Music feature extraction models based on classical methods [51, 20, 29], such as using features like mel-frequency cepstral coefficients [13] and fluctuation patterns [41, 43], modeled using methods like k-means clustering [31] and Gaussian mixture modeling [30, 1], have been investigated. Recently, methods based on deep learning have attracted attention due to their high precision of music feature extraction. Many studies utilize a convolutional neural network (CNN) [40, 7, 32, 11, 5, 44] and have demonstrated their high performance on their tasks.

In particular, music similarity representation learning (MSRL) with unsupervised, self-supervised or semi-supervised learning methods have gained popularity since it can handle previously unseen data and can be applied to various downstream tasks such as music tagging, genre classification, and key detection [27, 28, 38]. Artist labels [42] or music tags [6] have been used for training with triplet loss [14, 12], and their positive impact on a downstream music classification task or zero-shot performance has been demonstrated. Furthermore, contrastive learning approaches that assume "segments within the same

[1] https://go.pardot.com/l/52662/2023-10-23/ljk7xt/52662/169805013966KGzgtB/Spotify_2023_Culture_Next_Report_JP_v3.pdf.

[2] https://www.ifpi.org/wp-content/uploads/2020/03/Global_Music_Report_2023_State_of_the_Industry.pdf.

song are similar to each other" [36, 38], which is called S4 in this paper, or utilize data augmentation [48, 37] have demonstrated the strong effectiveness on several tasks [53], despite not requiring any labels. Moreover, HuBERT-style [21] masked language modeling [33], which estimates the tokens corresponding to masked parts of the input, has also achieved outstanding effectiveness on many downstream tasks [53] by utilizing teacher labels obtained from k-means clustering [27], the EnCodec model [9] or the data2vec-styled [2] approach [28]. Additionally, approaches utilizing the intermediate layer outputs of music generation models [10, 4] and text-music contrastive learning methods [22] have been established as techniques for acquiring music representations.

Although MSRL produces a single, general-purpose feature representation for each musical piece, which is sufficient for many downstream tasks, music recommendation and retrieval scenarios often require multiple, complementary descriptors in order to accommodate the diversity of individual user preferences. MSRL based on individual instruments (InMSRL) [17, 15] addresses this limitation by producing distinct feature vectors for each instrument within a musical piece, thereby furnishing a multi-dimensional representation of the musical piece. These instrument-specific embeddings (e.g., piano-only, drum-only, guitar-only) enable users to steer recommendation and retrieval results based on individual instruments. If an initially recommended musical piece is perceived as dissimilar to the query musical piece, the listener can, for example, request alternatives that more closely match the querys piano sound. Conventional InMSRL models have been trained with the similarity assumption S4, which has also been demonstrated to be effective in MSRL [36, 38]. This approach is advantageous in that it does not require any labels for training, especially since such labels for individual instruments are rarely available. To further obtain the music similarity representation for each instrument from the musical pieces that multiple instrument sounds are mixed, we have proposed three main approaches: *Clean* [17], *Cascade* [17], and *Direct* [15, 16]. *Clean*, which inputs clean individual instrument stems into the corresponding music similarity feature extractors, has been demonstrated its high performance in the music similarity feature extraction per each instrument stem. While *Clean* requires clean individual instrument stems as a searching query during inference, such stems are generally not publicly available, making it practically impossible to utilize it in general-purpose music recommendation and retrieval systems. Therefore, research on the InMSRL model that can input musical pieces themselves during inference and accurately obtain the music similarity representation per individual instruments has progressed, leading to the proposal of *Cascade* and *Direct*. *Cascade* sequentially performs music source separation (MSS) [19] and music similarity feature extractions. With this feature extraction strategy, the model is expected to clearly disentangle the individual instrument features from musical pieces. On the other hand, *Direct* extracts disentangled music similarity features using a single music

similarity feature extractor, with the goal of learning a disentangled feature space consisting of different subspaces for the individual instruments, similar to [52, 26]. With this architecture, the model can reduce the computational costs for feature extraction compared to *Cascade* and avoid artifacts from the preceding model (e.g., MSS models in *Cascade*) in feature extraction.

However, in *Cascade*, since the MSS model and music similarity feature extractors are independently trained, separation errors are likely to cause adverse effects on the music similarity feature extraction. Furthermore, in *Direct*, learning disentangled feature representation is not straightforward, and InMSRL performance tends to degrade for certain instruments, such as bass, piano, and guitar. Additionally, although *Clean*, *Cascade*, and *Direct* employ the S4-based training and successfully earn the music similarity representation between the same musical pieces, there is no guarantee that this training approach captures similarity between different musical piecesor corresponds to human perceptual similarity. Indeed, the previous study [18] has shown that while the music similarity representation from S4-based models between the same musical pieces has exhibited a strong correlation with human perception, the music similarity representation from models between different musical pieces has shown an insufficient correlation with human perception.

In this paper, we propose InMSRL methods utilizing music source separation and human preference aiming to construct a universally applicable InMSRL model and acquire a music similarity representation reflecting human perceptual similarity. For *Cascade*, we propose *Cascade-FT* that performs end-to-end fine-tuning (E2E-FT) of the MSS model and the music similarity feature extractors using an auxiliary separation loss. For *Direct*, we propose *Direct-Reconst* that uses multi-task learning based on the disentangled music similarity feature extraction and MSS based on reconstruction (Reconst) with the disentangled music similarity features. Furthermore, to allow the model to perform InMSRL aligned with human perceptual similarity, we introduce perception-aware fine-tuning (PAFT) utilizing a small amount of human preference labels. We conduct experimental evaluations and demonstrate that 1) the E2E-FT for *Cascade* significantly improves objective InMSRL performance, 2) the multi-task learning for *Direct* is also helpful to improve disentanglement performance in the feature extraction, 3) PAFT significantly enhances the perceptual InMSRL performance, and 4) *Cascade* with the E2E-FT and PAFT outperforms *Direct* with the multi-task learning and PAFT.

The rest of this paper is organized as follows: In Section 2, we describe the previously proposed InMSRL methods. In Section 3, we provide the details of our proposed method. In Section 4, we evaluate the proposed method through experimental evaluation. Finally, in Section 5, we present our conclusion.

2 Related Works

2.1 Conventional InMSRL Methods: Clean and Cascade

Hashizume *et al.* [17] proposed two InMSRL methods: one inputting clean individual instrument stems into the corresponding music similarity feature extractors (*Clean*) and the other inputting individual instrument stems separated by the pre-trained MSS model of Spleeter [19] into those extractors (*Cascade*).

The music similarity feature extractors of *Clean* and *Cascade* are trained using a triplet loss [14, 12]. In the i-th triplet, three types of sample segments, anchor $\mathbf{x}_i^{(a)}$ that serves as the basis, positive $\mathbf{x}_i^{(p)}$ defined as similar to the anchor and negative $\mathbf{x}_i^{(n)}$ defined as dissimilar to the anchor, are used. By denoting a distance function as $d(\cdot)$, a loss function can be formulated as follows:

$$\mathcal{L}_{\text{triplet}} = \max\{0, d(\mathbf{x}_i^{(a)}, \mathbf{x}_i^{(p)}) - d(\mathbf{x}_i^{(a)}, \mathbf{x}_i^{(n)}) + \delta\} \tag{1}$$

where δ is a margin that defines the minimum distance between the anchor-positive and anchor-negative pairs. To perform label-free learning, assuming S4 condition, a triplet is constructed as follows:

- Anchor: Extracted from a randomly selected musical piece
- Positive: Extracted from the same musical piece as that of the anchor
- Negative: Extracted from a different musical piece from that of the anchor.

In *Cascade*, it is inevitable to cause separation errors in MSS. The previous studies [17] have confirmed that the performance of *Cascade* significantly degrades compared with *Clean*. Therefore, it is crucial to optimize the MSS model for the instrument-dependent music similarity feature extractors.

2.2 A Conventional InMSRL Method: Direct

Hashizume *et al.* [15] also proposed the other InMSRL method to extract a disentangled music similarity feature with a single feature extractor, where the disentangled music similarity feature consists of subspaces for individual instrument-dependent music similarity features, e.g., the first to 128-th dimensional components of the 640-dimensional feature vector are used to represent the music similarity focusing on drums.

The training process first involves pre-training. In this training, the single disentangled music similarity feature extractor is trained using a target disentangled feature formed by concatenating the instrument-dependent music

similarity features extracted by *Clean*. Next, similar to *Cascade*, the disentangled music similarity feature extractor is further updated by using the triplet loss as shown in Equation 1.

However, unlike *Cascade*, it is not straightforward to train such a feature extractor. To develop the disentangled music similarity feature extractor working reasonably, the following two approaches are used.

- Conditioning the output of the disentangled music similarity feature extractor
- Using pseudo-musical-pieces as inputs.

Conditioning process conducts a masking operation inspired by other disentangled representation learning methods [52, 26]. For example, when focusing on the bass feature, we leave only the dimensional components corresponding to a subspace for the bass feature and masks the other dimensional components to 0. By partially masking the feature vector, each subspace can model the music similarity feature depending on a specific instrument.

The use of pseudo-musical-pieces is intended to prompt the model to extract only the target instrument features from the musical piece. Figure 1 shows an overview of the pseudo-musical-pieces. In Figure 1, a musical piece α and a musical piece β are similar to each other in drums but dissimilar in the other instruments. In contrast, the musical piece α and a musical piece γ are dissimilar in drums but similar in the other instruments. In the triplet loss-based learning, by using the musical piece α as the anchor, the musical piece β as the positive, and the musical piece γ as the negative, the model can focus only on the drum features. By treating this triplet setting as the basic triplet data, a previous study [15] further introduced additional triplet data. Specifically, the additional triplet data are constructed by swapping the positive and negative samples in the basic triplet data, and, during training with the additional triplet data, a different instrument from that in the basic triplet data is targeted.

However, it is still challenging to accurately disentangle a musical piece into the instrument-dependent subspace features, because the signals of different instruments overlap within the mix, making it hard to isolate the characteristics of single instrument. Consequently, the performance of InMSRL based on *Direct* tends to be insufficient.

2.3 Perceptual Music Similarity Representation Performance of a Conventional Method

Hashizume *et al.* have collected a large-scale dataset of human preference labels and have analyzed the human perception of similarity between individual instrumental stems within musical pieces [18]. Specifically, they have

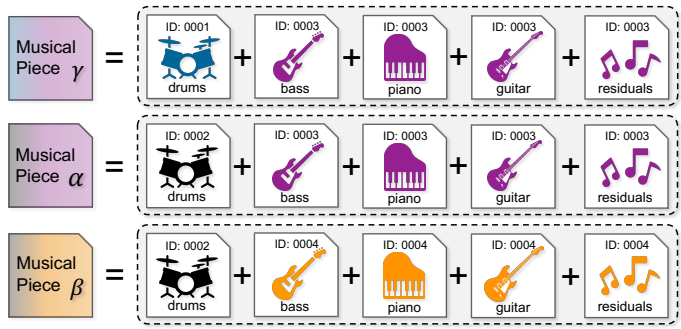

Figure 1: Overview of pseudo-musical-pieces. Instruments of the same color and the same ID indicate sample segments extracted from the same musical piece. This figure illustrates an example of the pseudo-musical-pieces created for learning focusing on drums.

conducted an ABX test with 586 participants, where each participant was asked to answer the question: "Which of A or B is more similar to X?", given three segments of musical pieces (X, A, and B). In constructing the ABX dataset, they defined the following two conditions for comparison:

- All-Diff: X, A, and B are extracted from entirely different songs
- One-Shared: Either A or B is extracted from the same song as X

The ABX dataset included drums, bass, piano, guitar, residuals, and mix tracks, with 240 pairs for each, totaling 480 pairs of ABX data. Here, residuals refer to all sounds in a musical piece except for drums, bass, piano, and guitar, while mix represents the full audio mix of the musical piece. Furthermore, in the experiment, each participant is given the clean stems of the target instruments from the musical pieces. Each pair of ABX data was evaluated by at least three participants, resulting in a total of 26,898 valid responses.

In addition, an experimental evaluation about the perceptual InMSRL performance of the conventional InMSRL model trained in S4 condition [15] have been conducted in [18]. The evaluation results showed that for the One-Shared condition, there was a strong correlation between human perceptual music similarity and music similarity by the S4-based model. Since, in the One-Shared condition, either A or B can satisfy S4 assumption, it is inferred that the S4 criterion aligns to some extent with human perceptual music similarity, and the conventional InMSRL model can represent the similarity between the same musical piece. However, in the All-Diff condition, only a weak correlation has been observed between the human perceptual music similarity and the music similarity by the S4-based model. This indicates that the conventional InMSRL model is inadequate to represent the music

similarity between segments from different musical pieces. Given that S4-based training successfully captures similarity aligned with S4, it is possible that this approach is not effective in learning similarity between different musical pieces that do not conform to the S4 criterion.

2.4 Research Using Human Preference Labels for Learning

In various research domains, efforts have been made to incorporate human preference labels into training in order to align learned embedding spaces with human subjectivity. In the field of speech quality prediction, contrastive learning using human-assigned Mean Opinion Scores has been introduced [46, 23]. In the domain of speaker embeddings, a learning method has been proposed in which a similarity matrix is constructed based on collected perceptual similarity ratings, and the model is trained to predict a dimensionality-reduced representation of that matrix [47]. Furthermore, in the field of music information retrieval, approaches that construct similar/dissimilar triplets based on human perceptual judgments of artist similarity and apply them to learning [39] have been proposed in the area of artist similarity representation. The application of human preference labels to training is expected to be also an effective approach for achieving InMSRL that reflects human perceptual similarity.

3 Proposed InMSRL Methods Leveraging Multi-task Learning and Human Preference

3.1 Cascade-FT

To address the issue of *Cascade*, we propose *Cascade-FT* to optimize the MSS model by performing end-to-end fine-tuning (E2E-FT).

3.1.1 Network Architecture

The network architecture of *Cascade-FT* consists of the MSS model and the instrument-dependent music similarity feature extractor connected in series as shown in Figure 2. The MSS model is based on the U-Net [45, 24] structure, similar to the Spleeter [19] used in *Cascade*. The network outputs a separation mask and the separated instrumental stem is generated by Hadamard product of the input music spectrogram and the separation mask. In this paper, we develop the instrument-dependent MSS models to separately estimate the separation masks for individual instrument stems. The instrument-dependent music similarity feature extractor is based on the U-Net encoder structure additionally using time-averaging and flattening operations and a

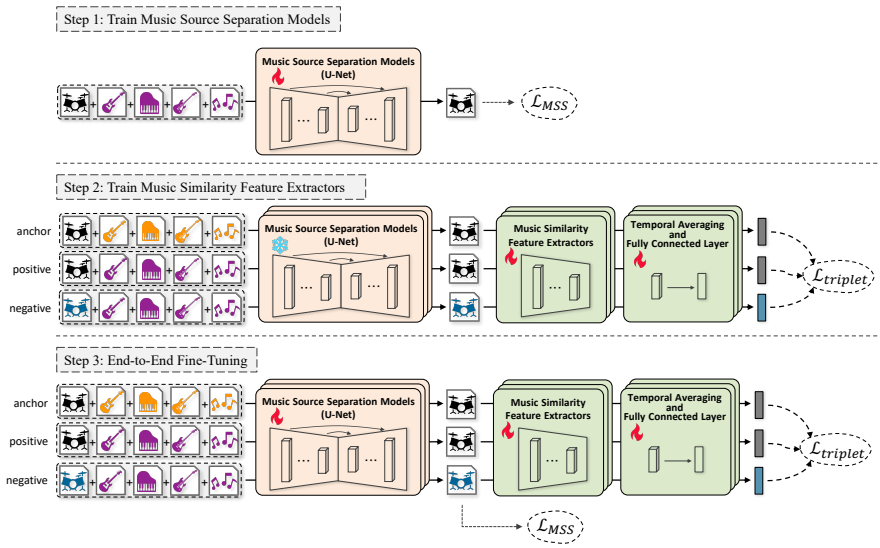

Figure 2: Overview of *Cascade-FT* model.

fully-connected layer to output a 128-dimensional feature vector for each instrument.

3.1.2 Training

The training procedure consists of three stages: training of the MSS models, training of the instrument-dependent music similarity feature extractors and E2E-FT. First, the MSS models are trained in the same manner as proposed by Jansson *et al.* [24]. The separation loss for each instrument stem (denoted as $\mathcal{L}_{MSS}$ in Figure 2) is calculated as the L1 loss between the output separated instrument amplitude spectrogram and a clean target instrument amplitude spectrogram. Next, the music similarity feature extractors are trained using the triplet loss given by Equation 1 (denoted as $\mathcal{L}_{triplet}$ in Figure 2) in the same manner as in *Cascade*. During the training, the MSS models are frozen and their parameters are not updated. The L2 norm is employed as the distance function $d(\cdot)$ in the triplet loss. Finally, in the E2E-FT stage, all parameters of the cascaded network consisting of the MSS models and the instrument-dependent music similarity feature extractors are updated by using a combined loss function given by the triplet loss for the instrument-dependent music similarity feature extractors and the separation loss for the MSS models as an auxiliary loss. Note that three inputs (anchor, positive, and negative) are required to compute the triplet loss, the auxiliary separation loss

for the MSS models during fine-tuning is calculated for all three inputs. In the training, the pseudo-musical-pieces segments (shown in Figure 1) are also used as in *Direct*. Besides, we implement the data augmentation as described in Section 3.2.3.

3.2 Direct-Reconst

To address the issue of *Direct*, we propose *Direct-Reconst* incorporating MSS based on the reconstruction (Reconst) with the disentangled music similarity features for the training of the disentangled feature extractors.

3.2.1 Network Architecture

Figure 3 shows the network architecture of *Direct-Reconst*. The *Direct-Reconst* network consists of three parts: the disentangled music similarity feature extractor, a reconstruction network to reconstruct each instrument stem from output sequences of the disentangled music similarity feature extractor, and time-averaging and flattening operations and fully-connected layer to generate the disentangled music similarity feature vector from the output sequences. The disentangled music similarity feature extractor has a similar structure to the encoder of U-Net [24], and the reconstruction network has a similar structure to the decoder of U-Net [24]. Each layer of the disentangled music similarity feature extractor and those of the reconstruction network are connected by skip connections. The instrument-dependent reconstruction networks are developed for individual instruments. As in the MSS models, the reconstructed instrument stem is generated by Hadamard product of the input music source spectrogram and the output separation mask.

3.2.2 Training

The training procedure consists of two stages: pre-training of the music similarity feature extractor and multi-task learning of the music similarity feature extractor and the instrument-dependent reconstruction network. In the pre-training of the music similarity feature extractor, we follow the same training procedure as in *Direct* [15]. We use 31 out of the 2^5 possible combinations of 5 musical instrument sources (drums, bass, piano, guitar, and residuals) as input, excluding the silent pattern. The training loss for the multi-task learning is a combination of the triplet loss given by Equation 1 (denoted as $\mathcal{L}_{triplet}$ in Figure 3) for the disentangled music similarity features and the reconstruction loss (denoted as $\mathcal{L}_{MSS}$ in Figure 3) for the output reconstructed instrument stems. The distance function $d(\cdot)$ in the triplet loss for the disentangled music similarity features is defined as the L2 norm. The reconstruction loss is

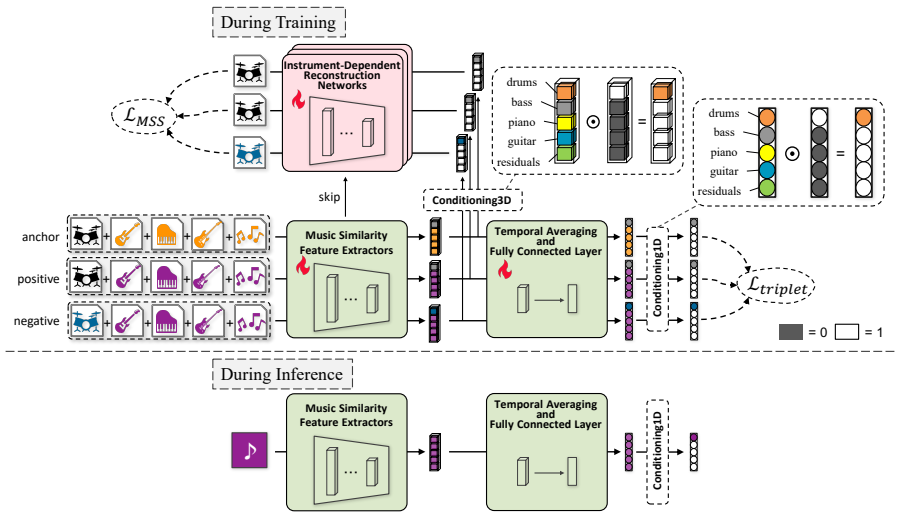

Figure 3: Overview of *Direct-Reconst* model. The same color of inputs and outputs of the networks indicate the segments extracted from the same musical pieces.

defined as the L1 loss between the output instrument amplitude spectrogram from the reconstruction network and the clean instrument amplitude spectrogram in the same manner proposed by Jansson *et al.* [24]. As in *Direct*, we use the conditioning operation and the pseudo-musical-pieces.

3.2.3 Disentanglement Enhancement

To enhance the disentangled music similarity feature extractor, we modify the conditioning process and utilize pseudo-musical-pieces. The modified conditioning process applies the masking operation to not only the output of the time-averaging and flattening operations and fully-connected layer (Conditioning1D in Figure 3) but also the input of the reconstruction network (Conditioning3D in Figure 3). Conditioning1D is the same as the conditioning process used in *Direct*. Conditioning3D is its extension to apply the masking operation to a feature sequence. By Conditioning3D, the reconstruction network can focus only on the features corresponding to each target instrument. For the pseudo-musical-pieces, we further introduce data augmentation (DA). While *Direct* generates a fixed set of triplet data of the pseudo-musical-pieces beforehand and use it in the training, *Direct-Reconst* introduces a process of randomly generating triplet data of the pseudo-musical-pieces each time to construct a mini-batch during training.

3.3 Fine-tuning Utilizing Human Preference

For improving the perceptual InMSRL performance, we propose a Perception-Aware Fine-Tuning (PAFT) which utilizes few human preference labels obtained from ABX test [18] as described in Section 2.3. The training with PAFT follows a two-step process. First, the training with triplet loss of each InMSRL model (described in Section 2.1, 2.2, 3.1.2, and 3.2.2) is conducted as pre-training. Then, PAFT fine-tunes these models using the human preference data obtained from ABX test. For the loss function for PAFT, the triplet loss given by Equation 1 is used. The data setting for the triplet loss is defined as follows:

- Anchor: The reference data in ABX test (denoted as X in Section 2.3)
- Positive: The segment (A or B) that is determined to be more similar to X in the ABX data
- Negative: The segment (A or B) that is determined to be less similar to X in ABX data

For *Clean*, the clean individual instrument stems are used for the inputs of models during PAFT. For *Cascade* and *Direct* approach, the pseudo-musical-pieces are used during PAFT. Here, the pseudo-musical-pieces during PAFT are defined using only the ABX data for the target instrument, while the non-target instruments are defined in the same manner as described in Section 2.2. During PAFT of *Cascade*, *Cascade-FT*, *Direct*, and *Direct-Reconst*, only the feature extractors are trained and the other parts of the model are frozen. With PAFT, the model can acquire the music similarity representation that aligns with human perceptual music similarity.

4 Experimental Evaluations

4.1 Dataset

The dataset used for evaluation was Slakh [35], which was also used in the previous study [17, 15]. The dataset consisted of MIDI-generated musical pieces and their instrument stems. Following previous studies [17, 15], we focused on four instrument classes: drums, bass, piano, and guitar. Since Slakh dataset provided finer-grained stems within these broad instruments (e.g., *Electric Guitar (clean)*), we followed the official Slakh recipe to mix all stems corresponding to each class, treating the resulting mixtures as the "drums," "bass," "piano," and "guitar" sources, respectively. All remaining stems that did not fit into these four instruments were combined into a single track and labeled "residuals."

The dataset consisted of 2100 musical pieces containing multiple groups of musical pieces generated from the same MIDI file. In this experiment, we excluded musical pieces generated from the same MIDI file, resulting in 1200 musical pieces used for training, 270 musical pieces used for validation, and 136 musical pieces used for evaluation.

4.2 Evaluation Metrics

As an evaluation metric, we used music ID estimation accuracy and perceptual similarity agreement as used in the previous studies [17, 15, 18]. Here, for music ID estimation accuracy, we used the following two metrics, a music ID estimation score on normal-test-musical-pieces (MES-Normal) and a music ID estimation score on pseudo-test-musical-pieces (MES-Pseudo). Additionally, to facilitate a deeper discussion of the models behavior, we introduce visualization of the output music similarity feature vectors.

4.2.1 Music Estimation Score on Normal-test-musical-pieces (MES-Normal)

To evaluate the performance of the feature representation, we used the accuracy of the music ID estimation with a simple method using the feature representation, following the approach of previous studies [17, 15]. Specifically, assuming that all test segments were embedded into the learned feature space beforehand and the music IDs of all segments were known except for a test segment to be estimated, we used the 5-nearest neighbors (5NN) method to estimate the music IDs of the test segments. In the evaluation for each instrument, we only used feature dimensions corresponding to the target instrument in 5NN distance calculation while masking the other feature dimensions. The entire test dataset (136 musical pieces) was used to calculate the music ID estimation accuracy.

In MES-Normal, high accuracy indicates that the feature vectors of segments from the same musical piece are closely clustered around each other. In other words, MES-Normal evaluates the representation performance of S4-based music similarity.

4.2.2 Music Estimation Score on Pseudo-test-musical-pieces (MES-Pseudo)

The proposed method aimed to learn the music similarity feature representations focusing on individual instruments. However, in MES-Normal, the ground truth label for the 5NN method was the same over all instruments as shown in the top part of Figure 4. Therefore, it was essentially hard to evaluate the disentanglement performance of the learned representations by MES-Normal. To investigate the disentanglement performance, we used

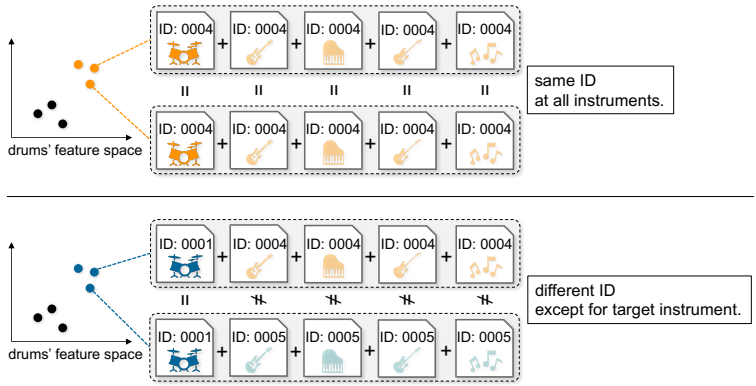

Figure 4: Difference between MES-Normal and MES-Pseudo. The top part of the figure shows MES-Normal, and the bottom part shows MES-Pseudo. This is the example of evaluation for the drums. Instruments of the same color and the same ID indicate segments extracted from the same musical piece.

pseudo-test-musical-pieces in MES-Pseudo. In MES-Pseudo, the ground truth label was different between the target instrument and the others; e.g., the label of the target instrument (i.e., drums label) was different from the others as shown in the bottom part of Figure 4.

The pseudo-musical-pieces used for the test were generated as follows (the process is also showed in Figure 5): 1) 10 musical pieces were selected from the test dataset to be used for the target instrument stems, 2) those 10 musical pieces were removed from the test dataset, 3) for each of those 10 target musical pieces, 3 musical pieces were further selected from the remaining test dataset, and they were used for the non-target instrument stems, and 4) each of the 10 target musical pieces and the corresponding 3 non-target musical pieces were mixed to generate 30 pseudo-musical-pieces in total. The entire test data for MES-Pseudo was constructed by using those 30 pseudo-musical-pieces as well as the 10 normal musical pieces used for the target musical pieces, consisting of 40 musical pieces in total. In the test, we excluded all segments extracted from the same musical-piece as that of each test segment to prevent the music ID estimation focusing on the non-target instruments; i.e., when using a segment within one of the 30 pseudo-musical pieces as a test segment, only segments within 2 pseudo-musical pieces and 1 normal musical piece had correct music ID labels; on the other hand, when using a segment within one of the 10 normal musical pieces as a test segment, only segments within 3 pseudo-musical-pieces had correct musical ID labels.

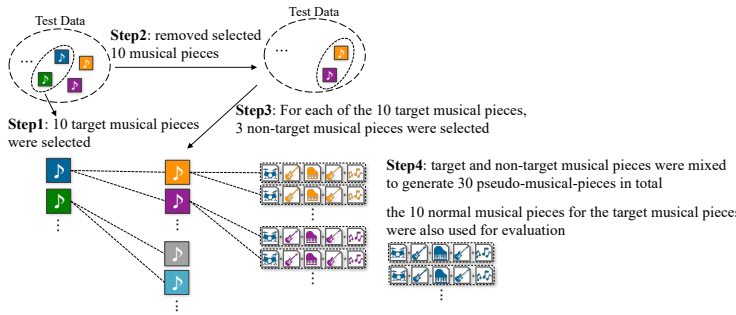

Figure 5: Data preparation process for MES-Pseudo. The same color indicates segments extracted from the same musical piece.

4.2.3 Perceptual Similarity Agreement

Although MES-Normal and MES-Pseudo had been used as evaluation metrics for assessing the performance of *Cascade* and *Direct*, they were limited to evaluating the representation performance of S4-based music similarity. To evaluate similarity not only between same musical pieces but also between different musical pieces, we used a perceptual similarity agreement utilizing ABX data obtained from previous research [18]. Specifically, the accuracy was calculated by comparing human responses and model predictions for the question, "Which of A or B is more similar to X?", given three segments of musical pieces (X, A, and B). The details of ABX data are described in Section 2.3. The ABX data used for evaluation included only the data where more than 75% of participants consistently selected one of the two options, either A or B, as the segment most similar to the reference segment X. The number of ABX data samples and subject responses are shown in Table 1.

Table 1: The number of ABX data and subject responses.

Method	drums	bass	piano	guitar
The number of ABX data				
all data (All-Diff)	240	240	240	240
all data (One-Shared)	240	240	240	240
above 75% (All-Diff)	115	112	106	105
above 75% (One-Shared)	214	194	215	213
The number of subject responses				
all (All-Diff)	2421	2429	2424	2425
all (One-Shared)	2408	2422	2425	2412
above 75% (All-Diff)	1163	1128	1083	1085
above 75% (One-Shared)	2163	1959	2159	2134

4.2.4 Visualization of Music Similarity Feature Vectors

To gain a deeper insight into the performance of each InMSRL method, particularly its disentangling performance, we introduced the visualization of music similarity features. During visualization, pseudo-musical-pieces were used as inputs of models. The used pseudo-musical-pieces were constructed as follows (see Figure 6). First, 10 musical pieces were selected. They were used for both the target instrument and non-target instruments to determine the pseudo musical pieces. Consequently, 100 pairs of the target musical piece and the non-target musical pieces were used in total to generate the pseudo musical pieces. Next, 10 segments were retrieved from each musical piece. Finally, 10 pseudo-segments were constructed for each of those 100 pseudo-musical-pieces by randomly selecting target and non-target segments from those retrieved 10 segments of the corresponding musical pieces and mixing them. Note that although 10 out of those 100 pseudo musical pieces were equivalent to the normal musical pieces, the constructed pseudo-segments were usually different from normal ones because of these mixing process using the randomly selected segments. In total, 1,000 pseudo-segments were used to visually investigate which instruments the model focused on.

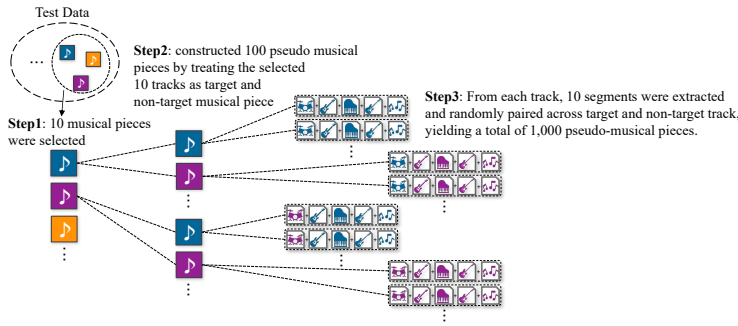

Figure 6: Data preparation process for visualization of music similarity feature vectors. The same color indicates segments extracted from the same musical piece.

To further deepen the analysis of the visualized distributions, MES-Pseudo was calculated for both the target and non-target music IDs. A higher MES-Pseudo value for a target music ID suggests that the model is successfully focusing on the features of the target instruments. Conversely, a lower MES-Pseudo value for a non-target music ID indicates that the model is effectively suppressing features associated with non-target instruments.

4.3 Experimental Conditions

Music segments used in the experiments were cut into 3-second segments for training based on S4 similarity, 5-second for PAFT and perceptual similarity agreement, and 10-second segments for validating and MES-Normal and MES-Pseudo. The training data and test data for PAFT were split from the ABX data obtained in ABX test [18] at a 7 : 3 ratio. The music segments where the target instrument was silent were excluded. The sampling rate was set to 44,100 Hz, and a window size of 2,048 and a frame shift of 512 were used for the short-time Fourier transform (STFT). The number of mel-frequency bins for the log mel-spectrogram used as input to the *Cascade-FT* music similarity feature extractors was set to 259.

Cascade and *Direct-Reconst*, when the encoder and decoder were regarded together as a complete source-separation model, share the same U-Net-style architecture as the network proposed in [24]. Both the encoder and the decoder comprised six convolutional layers. Each convolution was followed by batch normalization and a leaky ReLU activation with a negative slope of 0.2. In the decoder, a dropout layer with a probability of 0.5 was appended to the end of each of the first three layers. The feature extractors used in *Cascade*, and *Direct* (the encoder in *Direct-Reconst*) adopted an architecture identical to the encoder of the U-Net in [24]; thus, their structure was equivalent to the encoder detailed above. In these feature extractors, the final layer omitted batch normalization and leaky ReLU activation applied elsewhere. The convolutional output of the feature extractor was temporally averaged, smoothing the channel and frequency axes, and subsequently passed through a single linear layer that projects it to 128 dimensions for *Cascade* approach, and to 640 dimensions for *Direct* approach.

The learning rate in *Cascade-FT* was set to 5×10^{-5} for training based on S4, 1×10^{-5} for fine-tuning, and 5×10^{-5} for PAFT. The learning rate for the pre-training of *Direct-Reconst* and the multi-task training of the disentangled music similarity feature extractor and the reconstruction network was set to 1×10^{-4}. The batch size was set to 64 for both the *Cascade* and *Direct* approaches. Adam [25] was used to train both models. The maximum number of epochs was set to 400 except for PAFT, and training was terminated if the minimum value of the loss function on the validation data was not updated over 100 epochs. In PAFT, the number of epochs was set to 100. The model was trained and evaluated with one NVIDIA TITAN RTX, RTX 3090, RTX 2080 Ti, or RTX 3090.

4.4 Experimental Results

Evaluation results of MES-Normal and MES-Pseudo are shown in Table 2 and Table 3, respectively. We also show an evaluation result of MSS accuracy for

Table 2: Evaluation results of MES-Normal (%). The evaluation scores of *Clean* [17], *Cascade* w/ Spleeter [17] and *Direct* [15] are respectively quoted from [17] and [15]. In w/o pseudo-musical-pieces, the music similarity feature extractors are simply trained with normal musical pieces. Excluding ablation, the best results are highlighted in **bold**. "Gray" text indicates the scores used in the ablation study.

Method	drums	bass	piano	guitar	residuals
Clean [17]	98.04	94.60	98.14	96.35	-
Cascade w/ Spleeter [17]	88.91	63.87	50.34	-	-
Cascade w/o pre-trained MSS	90.98	73.39	80.77	79.53	-
w/o pseudo-musical-pieces	92.71	90.20	93.62	90.90	-
w/ E2E-FT (*Cascade-FT*)	**93.03**	**74.96**	**81.96**	**82.78**	-
w/o pseudo-musical-pieces	94.89	95.63	96.21	94.40	-
Direct [15]	89.69	**84.45**	**85.70**	**86.27**	84.86
w/ DA	89.33	71.09	79.74	81.75	85.67
w/ DA, Reconst (*Direct-Reconst*)	**91.14**	81.30	84.76	85.17	**88.84**

Table 3: Evaluation results of MES-Pseudo (%). The evaluation scores of *Direct* [15] are quoted from the previous study [17]. In w/o pseudo-musical-pieces, the music similarity feature extractors are simply trained with normal musical pieces. Excluding ablation, the best results are highlighted in **bold**."Gray" text indicates the scores used in the ablation study.

Method	drums	bass	piano	guitar	residuals
Cascade w/o pre-trained MSS	98.68	93.02	91.73	92.19	-
w/o pseudo-musical-pieces	95.09	77.30	81.02	77.60	-
w/ E2E-FT (*Cascade-FT*)	**98.91**	**94.80**	**93.55**	**93.89**	-
w/o pseudo-musical-pieces	95.96	71.54	69.59	77.40	-
Direct [15]	85.5	37.1	31.3	44.7	74.7
w/ DA	97.93	68.22	69.22	63.24	89.99
w/ DA, Reconst (*Direct-Reconst*)	**98.25**	**77.74**	**79.20**	**82.47**	**94.62**

Table 4: Evaluation results of the MSS accuracy for the output separated stem in *Cascade*. SDR (Signal-to-Distortion Ratio) was used for the evaluation. The results of *Cascade* w/ Spleeter [17] are quoted from the previous study [17]. Museval [49] was used for calculation of SDR. The best results are highlighted in **bold**

	SDR				
Method	drums	bass	piano	guitar	residuals
Cascade w/ Spleeter [17]	-13.7	-15.5	-14.7	–	
Cascade w/o pre-trained MSS	**15.50**	10.54	**7.81**	6.94	–
Cascade-FT	15.44	**10.88**	7.62	**7.08**	–
Direct-Reconst	14.17	9.10	5.45	6.26	**8.51**

the output separated stems in *Cascade* approach and *Direct-Reconst* in Table 4. Additionally, the results of perceptual similarity agreement are shown in Table 5.

Table 5: Evaluation results of perceptual similarity agreement. The "w/o PAFT" columns show the scores of models only trained with S4 similarity, and The "w/ PAFT" columns show the scores of models trained with PAFT. $[*, *]$ represents the 95% confidence interval calculated using the Clopper-Pearson method [8], and $\underset{\pm *}{*}$ represents the mean and standard deviation from three training runs, highlighting the instability in the models behavior caused by the limited data available for PAFT. The "mean" columns indicates the average evaluation score of drums, bass, piano, and guitar. The "PAFT data" column indicates the input data of models during PAFT. **Bold** indicates the highest value, while the underline represents the second-highest value. "Gray" text indicates the scores used in the ablation study.

Method	PAFT data	w/o PAFT drums	bass	piano	guitar	residuals	mean	w/ PAFT drums	bass	piano	guitar	residuals	mean
All-Diff													
Clean	clean	63.11 [60.27,65.89]	55.14 [52.18,58.07]	61.59 [58.62,64.50]	65.53 [62.62,68.36]	–	61.34	61.81 ±5.62	71.89 ±17.20	58.36 ±2.64	66.10 ±3.67	–	64.54
Cascade	pseudo	**69.56** [66.83,72.20]	62.32 [59.42,65.16]	57.34 [54.33,60.31]	**66.73** [63.84,69.53]	–	**63.99**	**71.09** ±0.75	75.50 ±2.58	60.45 ±5.52	**76.55** ±4.62	–	70.90
	normal							69.53 ±3.07	73.81 ±5.42	64.22 ±0.86	62.99 ±2.72	–	67.64
Cascade-PAFT	pseudo							70.54 ±3.49	**76.30** ±4.60	**64.65** ±4.62	75.80 ±0.16	–	**71.83**
Cascade-FT	pseudo	62.25 [59.40,65.05]	61.44 [58.52,64.29]	62.42 [59.46,65.31]	58.53 [55.53,61.48]	–	61.16	69.69 ±2.91	68.75 ±3.70	59.75 ±3.68	73.45 ±4.90	–	67.91
	normal							69.77 ±4.26	71.65 ±2.77	60.19 ±2.64	64.59 ±0.59	–	66.55
Direct	pseudo	56.74 [53.41,59.19]	**65.78** [62.93,68.55]	**69.64** [57.12,63.04]	62.20 [59.72,65.56]	57.63 [62.63,68.29]	61.55	63.95 ±0.80	68.51 ±2.24	62.64 ±1.67	64.50 ±2.37	57.11 ±3.14	64.90
Direct-Reconst	pseudo	56.32 [53.41,59.19]	**65.78** [62.93,68.55]	60.11 [57.12,63.04]	62.67 [59.72,65.56]	**65.50** [62.63,68.29]	61.47	55.27 ±1.64	72.93 ±3.69	56.08 ±1.06	71.94 ±2.55	**62.79** ±3.39	64.06
One-Shared													
Clean	clean	95.33 [94.35,96.18]	**92.39** [91.13,93.53]	94.30 [93.24,95.24]	**94.14** [93.06,95.10]	–	**94.04**	**95.90** ±0.00	**92.43** ±1.20	92.66 ±0.64	93.84 ±0.77	–	**93.71**
Cascade	pseudo	**96.26** [95.37,97.02]	91.17 [89.82,92.39]	**94.49** [93.44,95.41]	93.11 [91.95,94.15]	–	93.76	95.26 ±1.11	89.71 ±3.05	92.49 ±2.46	92.13 ±1.38	–	92.39
	normal							95.58 ±0.55	92.30 ±1.67	94.56 ±0.99	92.80 ±1.22	–	93.81
Cascade-PAFT	pseudo							**95.90** ±0.00	90.13 ±0.40	93.13 ±1.57	91.18 ±1.32	–	92.59
Cascade-FT	pseudo	**96.26** [95.37,97.02]	92.04 [90.75,93.20]	94.12 [93.04,95.07]	93.30 [92.15,94.32]	–	93.93	**95.90** ±0.00	90.93 ±1.17	**92.70** ±2.11	**93.97** ±0.68	–	93.37
	normal							95.58 ±0.55	92.01 ±0.40	93.35 ±1.57	90.33 ±1.32	–	92.82
Direct	pseudo	95.90 [94.19,97.21]	91.40 [89.09,93.36]	87.44 [84.89,89.69]	89.20 [86.74,91.35]	**93.04** [90.83,94.86]	90.99	**95.90** ±0.00	92.10 ±0.48	86.10 ±1.05	90.96 ±0.94	90.62 ±0.00	91.26
Direct-Reconst	pseudo	95.93 [95.01,96.72]	89.18 [87.72,90.52]	91.43 [90.17,92.58]	92.55 [91.35,93.63]	92.09 [90.80,93.24]	92.24	93.57 ±1.31	84.91 ±2.82	86.40 ±2.59	88.98 ±0.90	86.18 ±1.03	88.46

4.4.1 Evaluation of Cascade-FT on Music Estimation Scores

It can be observed from Table 2 that *Cascade-FT* achieves higher evaluation scores than the previous method [17] for all instruments. This suggests that the proposed methods achieve higher S4-based InMSRL performance compared to the previous method. From a comparison between *Cascade* w/o pre-trained MSS, w/o pseudo-musical-pieces and *Cascade* w/ Spleeter [17], the performance improvements can be seen in former. This is caused by the insufficient separation accuracy of Spleeter, as shown in Table 4. This poor performance of Spleeter is likely due to the fact that Spleeter is trained on music with raw-audio-songs, while the experiments in this paper and [17] use musical pieces generated from MIDI. Moreover, we can also observe that E2E-FT in the proposed method is effective for further InMSRL performance improvements under S4-based condition from a comparison between *Cascade* w/o pre-trained MSS and *Cascade-FT*. Separation accuracy does not show consistent improvement with E2E-FT. This suggests that E2E-FT optimizes non-target instrument sounds, treated as noise in the separated outputs, for InMSRL task. These results demonstrate that the performance of the MSS model in *Cascade* methods strongly affects the performance of InMSRL.

The disentanglement performance of each InMSRL method can be compared in Table 3. All evaluation scores of *Cascade-FT* exceed 90%. We also observe that E2E-FT is helpful to further improve the performance.

These results suggest that the proposed method *Cascade-FT* can learn high-quality S4-based music similarity feature representations focusing on individual instruments.

4.4.2 Evaluation of Direct-Reconst on Music Estimation Scores

Table 2 shows that *Direct-Reconst* does not outperform the previous method [15] for some instruments, i.e., bass, piano, and guitar. On the other hand, *Direct-Reconst* significantly outperforms previous method in the evaluation result of MES-Pseudo as shown in Table 3. These results indicate that *Direct* [15] for MES-Normal is significantly affected by the leakage of the other instrument features and its disentanglement performance is actually low. On the other hand, the proposed method *Direct-Reconst* not only improves the evaluation scores of MES-Pseudo but also maintains the evaluation scores of MES-Normal at the same level as the previous method. Therefore, the proposed method can achieve better InMSRL performance than the previous method. Table 3 also shows that DA significantly improves the MES-Pseudo score, demonstrating the effectiveness of DA. Additionally, a comparison of *Direct* w/ DA and *Direct-Reconst* in Tables 2 and 3 shows that the multi-task learning of the disentangled music similarity feature extraction and the reconstruction is effective for improving the InMSRL performance.

4.4.3 Evaluation of Perceptual InMSRL Performance Without PAFT

Table 5 shows the evaluation results of perceptual InMSRL performance. *Cascade* in Table 5 is corresponding to *Cascade* w/o pre-trained MSS in Table 2 and 3, and *Direct* in Table 5 is corresponding to *Direct* w/ DA in Table 2 and 3. Moreover, *Cascade-PAFT* refers to the models that replaces the E2E-FT of *Cascade-FT* with PAFT and conducts the E2E-FT and PAFT simultaneously.

First, we discuss the perceptual InMSRL performance of models without PAFT as shown in "w/o PAFT" columns of Table 5. Under the One-Shared condition, *Direct-Reconst* achieves a higher score than *Direct*. This is expected since the One-Shared condition can be regarded as based on S4, which is same to the MES-Normal and MES-Pseudo, indicating that *Direct-Reconst* achieve better perceptual InMSRL performance between the same musical pieces than *Direct*. Additionally, although the performance gain of *Cascade-FT* over *Cascade* is marginal under the One-Shared setting, the consistent improvements observed on MES-Normal and MES-Pseudo indicate an enhanced capability of InMSRL between the same musical piece.

Furthermore, in "w/o PAFT" columns of Table 5, the scores of models under the All-Diff condition are significantly lower than the scores of models under the One-Shared condition. However, this does not mean that similarity is not represented at all in the All-Diff condition, i.e., all models achieved an average perceptual similarity agreement performance of over 60%. This results are similar to the previous study [18]. These results indicate that S4-based training contributes to some extent not only to capturing similarity within same musical pieces but also to representing similarity between different musical pieces although this contribution is not sufficient.

Moreover, under the All-Diff condition, *Cascade* outperforms *Cascade-FT* and *Direct* outperforms *Direct-Reconst*, which is the opposite to that observed under the One-Shared condition and in the quantitative MES-Normal and MES-Pseudo evaluations. This means that effective learning under the training strategy based on S4 leads to higher InMSRL performance between the same musical pieces but does not contribute to improving InMSRL performance between different musical pieces.

4.4.4 Evaluation of Perceptual InMSRL Performance with PAFT

Next, we discuss the perceptual InMSRL performance of models with PAFT as shown in "w/ PAFT" columns of Table 5. We can observe that scores of models with PAFT are higher than those of models without PAFT under the All-Diff condition. In contrast, the scores under the One-Shared condition remain at the same performance level. These results demonstrate that PAFT contributes not only to enhancing the perceptual InMSRL performance between different musical pieces, which cannot be sufficiently trained based on S4 similarity, but also to maintaining to represent the similarity between same musical pieces.

Additionally, when comparing to the perceptual InMSRL performance between different musical pieces of models with PAFT, *Cascade* (row 2) is superior to *Cascade-FT* (row 5), and *Direct* (row 7) is superior to *Direct-Reconst* (row 8). This indicates that an effective training strategy based on S4 similarity does not necessarily lead to performance improvement through PAFT. It can also be considered that there is a significant discrepancy between the features emphasized across different musical pieces and those emphasized within the same musical piece.

Furthermore, *Cascade-PAFT* achieves the highest perceptual InMSRL performance under All-Diff condition. This can be considered as a result not only of minimizing the impact of separation errors from MSS model on the subsequent feature extractor but also of optimizing the entire network for music similarity based on human preference. This indicates that E2E-FT is also effective for improving perceptual InMSRL performance.

4.5 Discussion

The results of the visualization of music similarity feature vectors are shown in Figure 7, and the corresponding MES-Pseudo values for the target and non-target music IDs in the visualization are presented in Table 6.

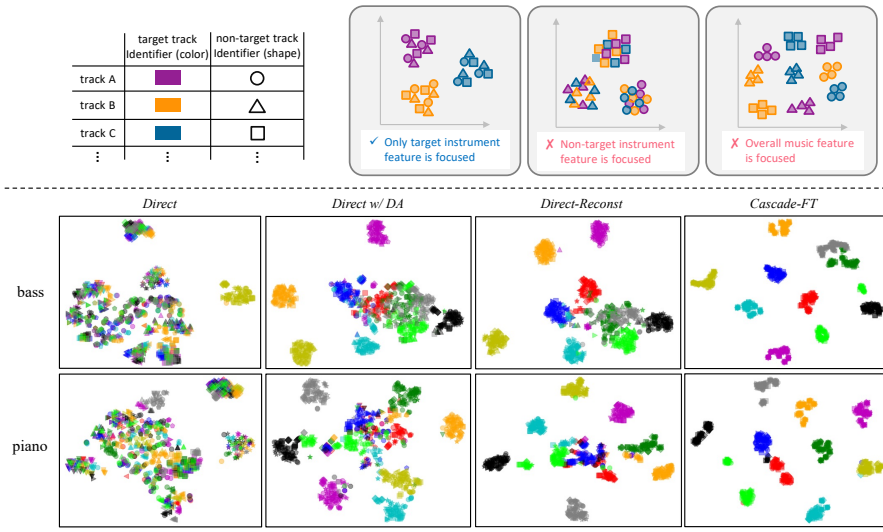

Figure 7: Visualization results of the music similarity features for pseudo-musical-pieces. In visualization, the music identification for the target instrument is represented by colors, while that for non-target instruments is represented by shapes. In this setting, the aggregation of music similarity features with the same color but the different shapes indicates that the model focuses only on the feature of target instrument. In contrast, the aggregation of music similarity features with the same shape but different colors indicates that the model focuses on the features of non-target instrument, while the aggregation of music similarity features with the same shape and color indicates that the model focuses on the features of overall musical pieces. The music similarity feature vectors were compressed to 2 dimension vectors by t-SNE [34].

Table 6: Results of MES-Pseudo computed on the pseudo-musical-pieces used in the visualization (Figure 7). The "Target" column shows MES-Pseudo scores for the music IDs corresponding to the target instruments, while the "Non-Target" column presents the scores for the music IDs corresponding to non-target instruments. **Bold** indicates the best performance, while underline denotes the second-best.

Method	**Target ↑**					**Non-Target ↓**				
	drums	bass	piano	guitar	residuals	drums	bass	piano	guitar	residuals
Direct	83.30	13.00	15.40	5.70	59.30	11.80	85.40	83.00	95.00	46.80
Direct w/ DA	98.90	76.00	78.60	57.50	95.90	0.70	17.50	17.10	40.70	4.80
Direct-Reconst	99.30	89.90	88.90	68.70	**98.40**	0.60	7.1	11.30	28.80	**2.60**
Cascade-FT	**100.00**	**99.90**	**99.50**	**92.20**	–	**0.00**	**0.00**	**0.20**	**6.80**	–

4.5.1 The Results of Visualizing the Music Similarity Feature Vectors

In Figure 7, *Direct-Reconst* exhibits the most desirable aggregation pattern among *Direct* approaches. Consistently, Table 6 shows that it achieves the highest MES-Pseudo for target music IDs and the lowest for non-target IDs, indicating the best separation of instrument-specific features. *Direct* [15] produces many clusters with identical shapes. Coupled with its low MES-Pseudo for target instruments and high scores for non-target instruments, this suggests that *Direct* captures features of unintended instruments rather than the target ones. *Direct* w/ DA forms many more same-color clusters than *Direct*, and its MES-Pseudo for target music IDs increases substantially, demonstrating the effectiveness of data augmentation. Moreover, *Direct-Reconst* further suppresses feature dispersion relative to *Direct* w/ DA and yields higher MES-Pseudo values, confirming the benefit of multi-task learning within *Direct* framework. Finally, *Cascade-FT* reduces feature dispersion even further, forming tight clusters that share the same color while differing in shape, thereby validating its superior instrument-wise feature disentanglement.

4.5.2 Clean, Cascade and Direct Approaches Comparison

First, we compare *Cascade* approach with *Direct* approach. Tables 2 and 3 show that *Direct* approach tends to have higher MES-Normal scores than MES-Pseudo scores for some instruments except for drums and residuals. Normally, MES-Normal score would be lower than or equal to the MES-Pseudo score because the MES-Normal uses 136 target labels compared to 10 for the MES-Pseudo at 5NN. This is considered to be due to the leakage of the other instrument features as discussed in Section 4.4.2. In contrast, the *Cascade* approach can more precisely focus only on target instruments, as demonstrated by the higher MES-Pseudo scores than the MES-Normal scores. Table 4 further supports this finding: *Cascade* approach achieves greater separation accuracy than *Direct-Reconst*. Additionally, in the visualization results shown in Figure 7 and the MES-Pseudo results for the same data presented in Table 6, *Cascade-FT* suppresses feature dispersion and achieves better MES-Pseudo values compared to *Direct* approach. These results demonstrate that *Cascade* approach achieves more reasonable S4-based InMSRL and higher disentanglement performance than *Direct* approach. Furthermore, from All-Diff scores of "w/ PAFT" columns of Table 5, *Cascade* approach also tends to have higher perceptual InMSRL performance between different musical pieces than *Direct* approach. This indicates that *Cascade* approach also outperforms *Direct* approach in terms of the effectiveness of PAFT. These several results demonstrates that *Cascade* approach achieves higher InMSRL performance not only in objective evaluation but also in perceptual similarity representation than *Direct* approach. On the other hand, *Direct* approach needs to use only the

disentangled music similarity feature extractor in the inference step, and therefore, its computational cost is lower than *Cascade* approach.

Next, we compare *Clean* approach with *Cascade* and *Direct* approach. MES-Normal scores of *Clean* are the highest in all of InMSRL models as shown in Table 2 and it is predicted that MES-Pseudo scores would be much higher scores than MES-Normal scores because of its less target labels. Additionally, *Clean* records the highest scores on the perceptual similarity agreement under the One-Shared condition as shown in "w/o PAFT" columns of Table 5. This result is to be expected because *Clean* utilizes clean individual instrument stems as input, which are generally not publicly available, therefore explicitly providing distinct individual instrument features to the music similarity feature extractors. In contrast, in terms of perceptual InMSRL performance between different musical pieces, *Clean* is not the top-performing model. Specifically, the scores of *Clean* without PAFT under the All-Diff condition are comparable to those of the *Cascade* and *Direct* approaches without PAFT as shown in "w/o PAFT" columns of Table 5, and the *Cascade* approach with PAFT rather outperforms *Clean* with PAFT as shown in "w/ PAFT" columns of Table 5. It is possible that this is caused by the overlearning of the music similarity between the same musical pieces. Since learning the similarity between the same musical pieces possibly requires only a specific part of the instrument features, such as hi-hat features in drums, models that use clean instrument stems have potential to focus primarily on those specific features. In such cases, the model is expected to be struggle to perform well when it needs to capture more diverse features, such as when learning similarity between different musical pieces. Conversely, in *Cascade*, while the feature extractors receive separated instrument stems that contain separation errors, these errors possibly mask some parts of the instrument features and potentially lead to more robust similarity representation learning. As a result, in perceptual InMSRL between different musical pieces with PAFT, the model was able to capture various features, which likely contributed to performance improvement.

From evaluation results, *Clean* is superior to *Cascade* and *Direct* for the InMSRL performance between same musical pieces, while *Cascade* tends to outperform *Clean* for the InMSRL performance between different musical pieces.

4.5.3 The Effectiveness of Pseudo-musical-pieces

In *Cascade* approach, the evaluation result of w/o pseudo-musical-pieces showed in Table 2 and Table 3 indicates that by using pseudo-musical-pieces, we can minimize the adverse effects of separation errors caused by the MSS model. Note that although the performance w/o pseudo-musical-pieces looks higher than that w/ it in Table 2, this result is caused by the leakage of the

other instrument features, and therefore, the actual InMSRL performance is limited. The use of pseudo-musical-pieces is also essential in *Direct* approach as reported in [15]. These results demonstrate that the use of pseudo-musical-pieces is an important technique to improve InMSRL performance. Furthermore, in Table 5, the All-Diff scores of *Cascade* approach which utilizes pseudo-musical-pieces during PAFT are higher than those of *Cascade* approach which utilizes normal musical pieces during PAFT. This means that pseudo-musical-pieces are more effective than normal musical pieces during PAFT in terms of perceptual InMSRL performance between different musical pieces since they minimize distraction from the features of other instrument features.

5 Conclusion

In this paper, we have proposed three methods to improve InMSRL performance. First, for *Cascade*, we have proposed end-to-end fine-tuning (E2E-FT) of the MSS model and the music similarity feature extractors using an auxiliary separation loss. Second, for *Direct*, we have proposed joint training of the disentangled feature extraction and MSS based on the reconstruction with the disentangled music similarity features. Third, we employ perception-aware fine-tuning (PAFT) utilizing human preference. We have conducted experimental evaluations and have demonstrated that the E2E-FT for *Cascade* improves InMSRL performance, the multi-task learning for *Direct* is also helpful to improve disentanglement performance in the feature extraction, PAFT enhances the perceptual InMSRL performance, and *Cascade* with the E2E-FT and PAFT outperforms *Direct* with the multi-task learning and PAFT.

In this study, we have relied on the Slakh dataset, which provides MIDI-generated stems. However, the most of real-world music is recorded as live, non-MIDI performances, and the corresponding stems are seldom released publicly. Moreover, since live recordings exhibit far greater expressive variability and stochasticity than MIDI-based data, performance on raw-audio songs can be expected to degrade compared to that on MIDI-generated tracks. As future work, we will extend our approach to operate on raw-audio songs. This will require the development of training strategies and model architectures capable of extracting instrument-specific features from fully mixed recordings, even when only limited stem tracks are available during training stage, and of maintaining high accuracy under realistic deployment conditions, such as real-world music recommendation and retrieval scene.

References

[1] J.-J. Aucouturier and F. Pachet, "Music Similarity Measures: Whats the Use?", *In Proc. ISMIR*, 2002, 157–63.

[2] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language", *In Proc. ICML*, 2022, 1298–312.

[3] M. Balabanovi and Y. Shoham, "Content-Based, Collaborative Recommendation", *Association for Computing Machinery*, 40(3), 1997, 66–72.

[4] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval", *In Proc. ISMIR*, 2021, 88–96.

[5] J. Choi, J. Lee, J. Park, and J. Nam, "Learning to Rank Music Tracks Using Triplet Loss", *In Proc. ISMIR*, 2019, 67–74.

[6] J. Choi, J. Lee, J. Park, and J. Nam, "Zero-shot Learning for Audio-based Music Classification and Tagging", *In Proc. ISMIR*, 2019, 67–74.

[7] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks", *In Proc. ISMIR*, 2016, 805–11.

[8] C. J. Clopper and E. S. Pearson, "The use of confidence or fiducial limits illustrated in the case of the binomial", *Biometrika*, 1934, 404–13.

[9] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression", *arXiv preprint arXiv:2210.13438*, 2022, 19 pages.

[10] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A Generative Model for Music", *arXiv preprint arXiv:2005.00341,* 2020, 20 pages.

[11] A. Elbir and N. Aydin, "Music Genre Classification and Music Recommendation by Using Deep Learning", *Electronics Letters*, 2020, 627–9.

[12] S. Florian, K. Dmitry, and P. James, "FaceNet: A Unified Embedding for Face Recognition and Clustering", *In Proc. CVPR*, 2015, 815–23.

[13] J. T. Foote, "Content-based retrieval of music and audio", *In Proc. SPIE Multimedia Storage and Archiving Systems II*, 1997, 138–47.

[14] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "DeViSE: A Deep Visual-Semantic Embedding Model", *In Advances NeurIPS*, 2013, 2121–9.

[15] Y. Hashizume, L. Li, A. Miyashita, and T. Toda, "Learning Multidimensional Disentangled Representations of Instrumental Sounds for Musical Similarity Assessment", in *arXiv e-prints: 2404.06682*, 8 pages, 2024.

[16] Y. Hashizume, L. Li, A. Miyashita, and T. Toda, "Learning Separated Representations for Instrument-based Music Similarity", *APSIPA Transactions on Signal and Information Processing*, 2025, 32 pages.

[17] Y. Hashizume, L. Li, and T. Toda, "Music Similarity Calculation of Individual Instrumental Sounds Using Metric Learning", *APSIPA ASC*, 2022, 33–8.

[18] Y. Hashizume and T. Toda, "Investigation of Perceptual Music Similarity Focusing on Each Instrumental Part", *In Proc. ICASSP*, 2025, 5 pages.

[19] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: A Fast and State-of-the Art Music Source Separation Tool with Pre-Trained Models", *The Journal of Open Source Software*, 5(50), 2019, 2154.

[20] A. Holzapfel and Y. Stylianou, "Musical Genre Classification Using Nonnegative Matrix Factorization-Based Features", *IEEE Trans. Audio, Speech, & Language Processing*, 16(2), 2008, 424–34.

[21] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units", *IEEE/ACM TASLP*, 2021, 3451–60.

[22] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis, "MuLan: A Joint Embedding of Music Audio and Natural Language", *In Proc. ISMIR*, 2022, 559–66.

[23] W.-C. Huang, E. Cooper, and T. Toda, "MOS-Bench: Benchmarking generalization abilities of subjective speech quality assessment models", in *arXiv preprint arXiv:2411.03715*, 15 pages, 2024.

[24] A. Jansson, E. J. Humphrey, N. Montecchio, R. M. Bittner, A. Kumar, and T. Weyde, "Singing Voice Separation with Deep U-Net Convolutional Networks", *In Proc. ISMIR*, 2017, 23–7.

[25] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization", *In ICLR*, 2014, 13 pages.

[26] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Disentangled Multidimensional Metric Learning for Music Similarity", in *IEEE ICASSP*, 2020, 6–10.

[27] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Xiao, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi, W. Huang, Z. Wang, Y. Guo, and J. Fu, "MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training", *In ICLR*, 2024, 1–23.

[28] Y. Li, R. Yuan, G. Zhang, Y. Ma, C. Lin, X. Chen, A. Ragni, H. Yin, Z. Hu, H. He, E. Benetos, N. Gyenge, R. Liu, and J. Fu, "MAP-Music2Vec: A Simple and Effective Baseline for Self-Supervised Music Audio Representation Learning", *In Late-Breaking Demo Session of ISMIR*, 2022, 3 pages.

[29] T. Lidy and A. Rauber, "Evaluation of feature extractors and psycho-acoustic transformations for music genre classification", *In Proc. ISMIR*, 2005, 34–41.

[30] Z. Liu and Q. Huang, "Content-based indexing and retrieval-by-example in audio", *In Proc. ICME*, 2000, 877–80.

[31] B. Logan and A. Salommon, "A music similarity function based on signal analysis", *In Proc. ICME*, 2001, 22–5.

[32] V. Lostanlen and C.-E. Cella, "Deep convolutional networks on the pitch spiral for music instrument recognition", *In Proc. ISMIR*, 2016, 612–8.

[33] Y. Ma, R. Yuan, Y. Li, G. Zhang, X. Chen, H. Yin, C. Lin, E. Benetos, A. Ragni, N. Gyenge, R. Liu, G. Xia, R. Dannenberg, Y. Guo, and J. Fu, "On the Effectiveness of Speech Self-supervised Learning for Music", *In Proc. ISMIR*, 2023, 457–65.

[34] L. van der Maaten and G. Hinton, "Visualizing Data Using t-SNE", *Journal of Machine Learning Research*, 2008, 2579–605.

[35] E. Manilow, G. Wichern, P. Seetharaman, and J. L. Roux, "Cutting Music Source Separation Some Slakh: A Dataset to Study the Impact of Training Data Quality and Quantity", in *IEEE WASPAA*, 2019, 45–9.

[36] M. C. McCallum, "Unsupervised Learning of Deep Features for Music Segmentation", *In ICASSP*, 2019, 346–50.

[37] M. C. McCallum, M. E. P. Davies, F. Henkel, J. Kim, and S. E. Sandberg, "On the Effect of Data-Augmentation on Local Embedding Properties in the Contrastive Learning of Music Audio Representations", *In ICASSP*, 2024, 671–5.

[38] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. F. Ehmann, "Supervised and Unsupervised Learning of Audio Representations for Music Understanding", *In Proc. ISMIR*, 2022, 256–63.

[39] B. McFee and G. Lanckriet, "Heterogeneous Embedding for Subjective Artist Similarity", *In Proc. ISMIR*, 2009, 513–8.

[40] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation", *In Advances NeurIPS*, 2013, 2643–51.

[41] E. Pampalk, "Computational Models of Music Similarity and their Application in Music Information Retrieval", *PhD thesis, Vienna University of Tech*, 2006, 165 pages.

[42] J. Park, J. Lee, J. Park, J.-W. Ha, and J. Nam, "Representation Learning of Music Using Artist Labels", *In Proc. ISMIR*, 2018, 717–24.

[43] T. Pohle, D. Schnitzer, M. Schedl, P. Knees, and G. Widmer, "On Rhythm and General Music Similarity", *In Proc. ISMIR*, 2006, 525–30.

[44] J. Pons and X. Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging", *arXiv preprint arXiv:1909.06654*, 2019, 2 pages.

[45] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", in *MICCAI*, 2015, 234–41.

[46] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "Utmos: Utokyo-sarulab system for voicemos challenge 2022", *In Proc. Interspeech*, 2022, 4521–5.

[47] Y. Saito, S. Takamichi, and H. Saruwatari, "DNN-based speaker embedding using subjective inter-speaker similarity for multi-speaker modeling in speech synthesis", *In Proc. SSW10*, 2019, 51–6.

[48] J. Spijkervet and J. A. Burgoyne, "Contrastive Learning of Musical Representations", *In Proc. ISMIR*, 2021, 673–81.

[49] F. R. Stöter, A. Liutkus, and N. Ito, "The 2018 Signal Separation Evaluation Campaign", in *Latent Variable Analysis and Signal Separation*, 2018, 293–305.

[50] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques", *Adv. in Artif. Intell.*, 2009, 19 pages.

[51] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals", *IEEE Trans. Speech & Audio Processing*, 10(5), 2002, 293–302.

[52] A. Veit, S. Belongie, and T. Karaletsos, "Conditional Similarity Networks", in *IEEE CVPR*, 2017, 1781–9.

[53] R. Yuan, Y. Ma, Y. Li, G. Zhang, X. Chen, H. Yin, L. Zhuo, Y. Liu, J. Huang, Z. Tian, B. Deng, N. Wang, C. Lin, E. Benetos, A. Ragni, N. Gyenge, R. Dannenberg, W. Chen, G. Xia, W. Xue, S. Liu, S. Wang, R. Liu, Y. Guo, and J. Fu, "MARBLE: Music Audio Representation Benchmark for Universal Evaluation", *In Advances NeurIPS*, 2023, 39626–47.