Multi-Task Learning Approaches for Music Similarity Representation Learning Based on Individual Instrument Sounds

Takehiro Imamura*, Yuka Hashizume* and Tomoki Toda*
* Nagoya University, Japan
imamura.takehiro@g.sp.m.is.nagoya-u.ac.jp, hashizume.yuuka@g.sp.m.is.nagoya-u.ac.jp, tomoki@icts.nagoya-u.ac.jp

***Abstract*—Music similarity representation learning (MSRL) based on individual instrument sounds (InMSRL) is a potential technique to develop a new function in music recommendation and retrieval systems, allowing users to focus on multiple partial elements of music pieces. There have been proposed two main approaches, *Cascade* that sequentially performs music source separation (MSS) and music similarity feature extraction for each instrument sound and *Direct* that directly extracts disentangled music similarity features. Each approach has a specific problem; in *Cascade*, a separation error often causes adverse effects on the feature extraction; and in *Direct*, it is usually hard to learn accurately disentangled features. In this paper, we propose multi-task learning approaches that leverage MSS to address these issues. For *Cascade*, we propose end-to-end fine-tuning of the MSS model and the music similarity feature extractors using an auxiliary separation loss. For *Direct*, we propose multi-task learning based on the disentangled music similarity feature extraction and MSS based on reconstruction with the disentangled music similarity features. We conduct experimental evaluations and demonstrate that 1) the end-to-end fine-tuning for *Cascade* significantly improves InMSRL performance, 2) the multi-task learning for *Direct* is also helpful to improve disentanglement performance in the feature extraction, and 3) *Cascade* with the end-to-end fine-tuning outperforms *Direct* with the multi-task learning.**

I. Introduction

Music recommendation and retrieval systems are useful technologies under the current situation that the number of music pieces has already exceeded 1 billion[1] and further market expansion is expected[2]. Methods utilizing users' listening histories [1], [2] have been widely used in these systems although these methods cause several limitations, e.g., hard to handle music pieces with less listening records. To address this issue, content-based methods to extract content features from a music piece to capture its characteristics have been studied. Recently methods based on deep learning have attracted attention since they can extract more precise content features [3]–[6] than classical methods [7], [8]. In particular, music similarity representation learning (MSRL) capable of extracting music similarity features in an unsupervised manner without using any hand-crafted labels is a promising way to develop the music recommendation and retrieval systems to widely handle existing music pieces.

MSRL based on individual instrument sounds (InMSRL) [9], [10] is a potential technique to develop a new function allowing users to focus on multiple partial elements of music pieces, e.g., searching for music pieces with similar drum sounds. Hashizume et al. [9] proposed InMSRL model, which inputs clean individual instrument sounds into the music similarity feature extractors (*Clean*), and demonstrated its high performance in the music similarity feature extraction. However, in general, these clean individual instrument sounds are not publicly available, making it practically impossible to utilize them in general-purpose music recommendation and retrieval systems. Therefore, research on the InMSRL model, which inputs the music pieces themselves, has progressed and two main approaches have been proposed. The first approach sequentially performs music source separation (MSS) and music similarity feature extractions (*Cascade*) [9]. However, since an MSS model and music similarity feature extractors are independently trained, separation errors are likely to cause adverse effects on the music similarity feature extraction. On the other hand, the second approach directly extracts disentangled music similarity features (*Direct*) [10]. This method learns a disentangled feature space consisting of different subspaces for individual instrument sounds. However, such a learning is not straightforward, and InMSRL performance tends to degrade for some instruments.

In this paper, we propose multi-task learning approaches that leverage MSS to address issues of *Cascade* and *Direct* and aim to construct a universally applicable InMSRL model. For *Cascade*, we propose *Cascade-FT* that performs end-to-end fine-tuning (FT) of the pre-trained 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. We conduct experimental evaluations and demonstrate that 1) *Cascade-FT* can improve InMSRL performance compared to *Cascade*, 2) *Direct-Reconst* can improve disentanglement performance in the music similarity feature extraction, and 3) *Cascade-FT* have better performance than *Direct-Reconst*.

[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

II. Conventional InMSRL Methods

A. Clean and Cascade

Hashizume et al. [9] proposed two InMSRL methods: one inputting clean individual instrument sounds into music similarity feature extractors (*Clean*) and the other inputting individual instrument sounds separated by the pre-trained MSS model of Spleeter [11] into them (*Cascade*).

The music similarity feature extractor of *Clean* and *Cascade* is both trained using a triplet loss. In the i-th triplet, the three types of sample segments, which is 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:

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

where δ is a margin that defines the minimum distance between the anchor-positive and anchor-negative pairs. To perform label-free learning, assuming that sample segments extracted from the same music piece are similar to each other, a triplet is constructed as follows:

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

In *Cascade*, it is inevitable to cause separation errors in MSS. The previous studies [9] 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.

B. Direct

Hashizume et al. [10] 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 the disentangled features as reference targets, which is formed by concatenating 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 (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-music-pieces as inputs.

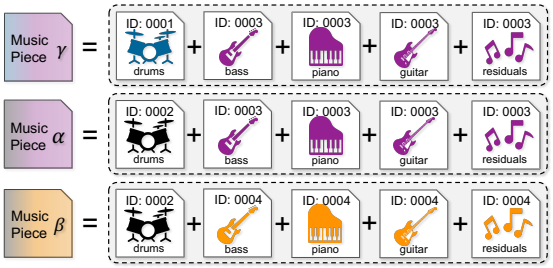

Fig. 1. Overview of pseudo-music-pieces. Instruments of the same color and the same ID indicate sample segments extracted from the same music-piece. This figure illustrates an example of the pseudo-music-pieces created for learning focusing on drums.

Conditioning process conduct a masking operation inspired by other disentangled representation learning [12], [13]. 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 sound.

The Use of pseudo-music-pieces aims to improve the disentangled performance of *Direct*. Fig. 1 shows an overview of the pseudo-music-pieces. In Fig. 1, a music piece α and a music piece β are similar to each other in drums but dissimilar in the other instruments sounds. In contrast, the music piece α and a music piece γ are dissimilar in drums but similar in the other instruments. In the triplet loss-based learning, by using the music piece α as the anchor, the music piece β as the positive, and the music piece γ as the negative, the model can focus only on the drum features.

However, it is still challenging to accurately disentangle a music piece into the instrument-dependent subspace features. Consequently, the performance of InMSRL based on *Direct* tends to be insufficient.

III. Proposed InMSRL Methods Leveraging Multi-task Learning

A. Cascade-FT

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

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 Fig. 2. The MSS model is based on the U-Net [14], [15] structure, similar to the Spleeter [11] used in *Cascade*. The network outputs a separation mask and the separated instrument sound 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 sounds. The instrument-dependent music similarity feature extractor is based on the U-Net encoder structure

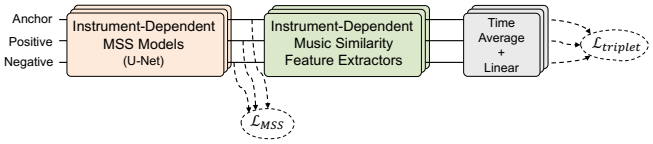

Fig. 2. Overview of *Cascade-FT* model.

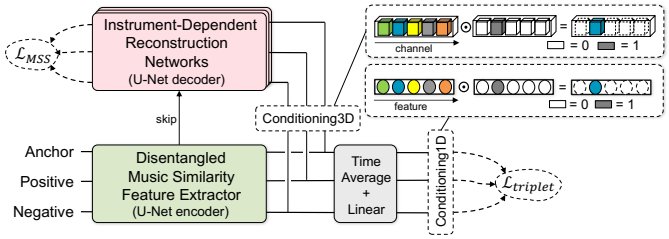

Fig. 3. Overview of *Direct-Reconst* model.

additionally using time-averaging and flattening operations and a fully-connected layer to output a 128-dimensional feature vector for each instrument sound.

2) Training: The training procedure consists of three stages: training of the MSS models, training of the instrument-dependent music similarity feature extractors and end-to-end fine-tuning. First, the MSS models are trained in the same manner as proposed by Jansson et al. [15]. The separation loss for each instrument sound (denoted as $\mathcal{L}_{MSS}$ in Fig. 2) is calculated as the L1 loss between the output separated instrument sound and a clean target instrument sound. Next, the music similarity feature extractors are trained using the triplet loss given by (1) (denoted as $\mathcal{L}_{triplet}$ in Fig. 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 end-to-end fine-tuning 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-music-pieces segments (shown in Fig. 1) are also used as in *Direct*. Besides, we implement the data augmentation as described in Section III-B3.

B. 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.

1) Network Architecture: Fig. 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 sound from output sequences of the disentangled music similarity feature extractor, and a 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 [15], and the reconstruction network has a similar structure to the decoder of U-Net [15]. The 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 instrument sounds. As in the MSS models, the reconstructed instrument sound is generated by Hadamard product of the input music source spectrogram and the output separation mask.

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* [10]. 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 (1) (denoted as $\mathcal{L}_{triplet}$ in Fig. 3) for the disentangled music similarity features and the reconstruction loss (denoted as $\mathcal{L}_{MSS}$ in Fig. 3) for the output reconstructed instrument sounds. 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 defined as the L1 loss between the output instrument sound from the reconstruction network and the clean instrument sound in the same manner proposed by Jansson et al. [15]. As in *Direct*, we use the conditioning operation and the pseudo-music-pieces.

3) Disentanglement Enhancement: To enhance the disentangled music similarity feature extractor, we modify the conditioning process and utilize pseudo-music-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 Fig. 3) but also the input of the reconstruction network (Conditioning3D in Fig. 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 sound. For the pseudo-music-pieces, we further introduce data augmentation (DA). While *Direct* generates a fixed set of triplet data of the pseudo-music-pieces beforehand and use it in the training, *Direct-Reconst* introduces a process of randomly generating triplet data of the pseudo-music-pieces each time to construct a mini-batch during training.

IV. Experimental Evaluations

A. Dataset

The dataset used for evaluation was the Slakh [16], which was also used in the previous study [9], [10]. The dataset consisted of MIDI-generated music pieces and MIDI instrument tracks in the music pieces. Each music piece contained drums, bass, piano, and guitar sounds. Following previous studies [9], [10], the other sounds in the music pieces were treated as residuals.

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

B. Evaluation Metrics

The proposed methods were based on the same learning paradigm as the previous studies [9][10], assuming that segments extracted from the same music piece were similar to each other. Therefore, as an evaluation metric, we used music ID estimation accuracy as used in the previous studies. In this experiment, we used the following two metrics, a music ID estimation score on normal-test-music-pieces (MES-Normal) and a music ID estimation score on pseudo-test-music-pieces (MES-Pseudo).

1) Music Estimation Score on Normal-Test-Music-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. 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 ID of the test segment. In the evaluation for each instrument sound, 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 music pieces) was used to calculate the music ID estimation accuracy.

2) Music Estimation Score on Pseudo-Test-Music-Pieces (MES-Pseudo): The proposed method aimed to learn the music similarity feature representations focusing on individual instrument sounds. However, in MES-Normal, the ground truth label for the 5NN method was the same over all instrument sounds as shown in the top part of Fig. 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 pseudo-test-music-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 sound (i.e., drums label) was different from the others as shown in the bottom part of Fig. 4. Furthermore, we excluded all segments extracted from the same pseudo-music-piece as that of each test segment to prevent the music ID estimation focusing on the no-targeted instruments. The pseudo-test-music-pieces used for the test consisted of 50 music pieces in total, with 10 music pieces for each target instrument sound label and 5 music pieces for each non-target instrument sound label.

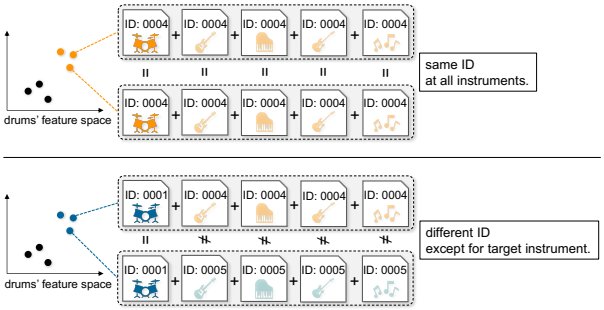

Fig. 4. Difference between MES-Normal and MES-Pseudo. The top part of the figure shows MES-Normal, and the bottom part shows MES-Pseudo. This is the example of evaluation for the drums. Instruments of the same color and the same ID indicate segments extracted from the same music piece.

C. Experimental conditions

Music segments used in the experiments were cut into 3-second segments for training and 10-second segments for validating and testing. The music segments where the target instrument was silent were excluded. The sampling rate was set to 44100 Hz, and a window size of 2048 and an offset of 512 were used for the short-time Fourier transform (STFT). The number of mel frequencies for the log Mel-spectrogram used as input to the *Cascade-FT* music similarity feature extractors was set to 259. The learning rate for the training of the MSS models and the music similarity feature extractors in *Cascade-FT* was set to 0.00005, and the learning rate for the fine-tuning was set to 0.00001. 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 0.0001. Adam [17] was used to train both models. The maximum number of epochs was set to 400, and training was terminated if the minimum value of the loss function on the validation data was not updated over 100 epochs.

D. Experimental Results

Evaluation results of MES-Normal and MES-Pseudo are shown in Table I and Table II, respectively. We also show an evaluation result of MSS accuracy for the output separated sounds in *Cascade* methods in Table III.

1) Evaluation of Cascade-FT: It can be observed from Table I that *Cascade-FT* achieves higher evaluation scores than the previous method [9] for all instruments. This suggests that the proposed methods achieve higher InMSRL performance compared to the previous method. From a comparison between *Cascade* w/ from-scratch w/o pseudo-music-pieces and *Cascade* w/ Spleeter [9], the performance improvements can be seen in former. This is caused by the insufficient separation accuracy of the MSS model of *Cascade* [9], as shown in Table

TABLE I

EVALUATION RESULTS OF MES-NORMAL (%). THE EVALUATION SCORES OF *Clean* [9], *Cascade* W/ SPLEETER [9] AND *Direct* [10] ARE RESPECTIVELY QUOTED FROM [9] AND [10]. IN W/O PSEUDO-MUSIC-PIECES, THE MUSIC SIMILARITY FEATURE EXTRACTORS ARE SIMPLY TRAINED WITH NORMAL MUSIC PIECES. EXCLUDING ABLATION, THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Method	drums	bass	piano	guitar	residuals
Clean [9]	98.04	94.60	98.14	96.35	-
Cascade w/ Spleeter [9]	88.91	63.87	50.34	-	-
Cascade w/ from-scratch	90.98	73.39	80.77	79.53	-
w/o pseudo-music-pieces (ablation study)	92.71	90.20	93.62	90.90	-
Cascade w/ from-scratch, FT (*Cascade-FT*)	**93.03**	**74.96**	**81.96**	**82.78**	-
w/o pseudo-music-pieces (ablation study)	94.89	95.63	96.21	94.40	-
Direct [10]	89.69	**84.45**	**85.70**	**86.27**	84.86
Direct w/ DA	89.33	71.09	79.74	81.75	85.67
Direct w/ DA, Reconst (*Direct-Reconst*)	**91.14**	81.30	84.76	85.17	**88.84**

TABLE II

EVALUATION RESULTS OF MES-PSEUDO (%). THE EVALUATION SCORES OF *Direct* [10] ARE QUOTED FROM THE PREVIOUS STUDY [9]. IN W/O PSEUDO-MUSIC-PIECES, THE MUSIC SIMILARITY FEATURE EXTRACTORS ARE SIMPLY TRAINED WITH NORMAL MUSIC PIECES.

Method	drums	bass	piano	guitar	residuals
Cascade w/ from-scratch	98.68	93.02	91.73	92.19	-
w/o pseudo-music-pieces (ablation study)	95.09	77.30	81.02	77.60	-
Cascade w/ from-scratch, FT (*Cascade-FT*)	**98.91**	**94.80**	**93.55**	**93.89**	-
w/o pseudo-music-pieces (ablation study)	95.96	71.54	69.59	77.40	-
Direct [10]	85.5	37.1	31.3	44.7	74.7
Direct w/ DA	97.93	68.22	69.22	63.24	89.99
Direct w/ DA, Reconst (*Direct-Reconst*)	**98.25**	**77.74**	**79.20**	**82.47**	**94.62**

TABLE III

EVALUATION RESULTS OF THE MSS ACCURACY FOR THE OUTPUT SEPARATED SOUND IN *Cascade*. SDR (SIGNAL-TO-DISTORTION RATIO) WAS USED FOR THE EVALUATION. THE RESULTS OF *Cascade* W/ SPLEETER [9] ARE QUOTED FROM THE PREVIOUS STUDY [9]. MUSEVAL [18] WAS USED FOR CALCULATION OF SDR.

	SDR			
Method	drums	bass	piano	guitar
Cascade w/ Spleeter [9]	-13.7	-15.5	-14.7	-
Cascade w/ from-scratch	**15.50**	10.54	**7.81**	6.94

III. 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 [9] use music pieces generated from MIDI. Moreover, we can also observe that the fine-tuning in the proposed method is effective for further performance improvements from a comparison between *Cascade* w/ from-scratch and *Cascade-FT*. This results demonstrates that the performance of the MSS model in *Cascade* methods strongly affects the accuracy of InMSRL.

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

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

2) Evaluation of Direct-Reconst: Table I shows that *Direct-Reconst* does not outperform the previous method [10] 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 II. These results indicate that *Direct* [10] for MES-Normal is significantly affected by the leakage of the other instrument sounds 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 II 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 I and II shows that the multi-task learning of the disentangled music similarity feature extraction and the reconstruction is effective for improving the InMSRL performance.

3) Clean, Cascade and Direct Approaches Comparison: Tables I and II show that the *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 sounds as discussed in Section IV-D2. In contrast, the *Cascade* approach can more precisely focus only on target instrument sound, as demonstrated by the higher MES-Pseudo scores than the MES-Normal scores. Therefore, it is demonstrated that the *Cascade* approach achieves higher InMSRL performance than the *Direct* approach. On the other hand, the *Direct* approach needs to use only the disentangled music similarity feature extractor in the inference step, and therefore, its computational cost is lower than the *Cascade* approach.

MES-Normal scores of *Clean* are the most highest in all

of InMSRL models and it is predicted that MES-Pseudo scores would be much higher scores than MES-Normal scores because of its less target labels. This result is to be expected because *Clean* utilizes clean individual instrument sounds as input, which are generally not publicly available, therefore explicitly providing distinct individual instrument features to the music similarity feature extractors. By utilizing multi-task learning, we were able to improve the performance of *Cascade* and *Direct* models, which input the music pieces themselves. However, there is still room for further improvement in *Cascade* and *Direct* approach considering *Clean* model.

4) The effectiveness of pseudo-music-pieces: In *Cascade* approach, the evaluation result of w/o pseudo-music-pieces showed in Table I and Table II indicates that by using pseudo-music-pieces, we can minimize the adverse effects of separation errors caused by the MSS model. Note that although the performance w/o pseudo-music-pieces looks higher than that w/ it in Table I, this result is caused by the leakage of the other instrument sounds, and therefore, the actual InMSRL performance is limited. The use of pseudo-music-pieces is also essential in *Direct* approach as reported in [10]. These results demonstrate that the use of pseudo-music-pieces is an important technique to improve InMSRL performance.

V. Conclusion

In this paper, we have proposed two methods to improve InMSRL performance by modifying two existing approaches, *Cascade* and *Direct*. For *Cascade*, we have proposed end-to-end fine-tuning of the MSS model and the music similarity feature extractors using an auxiliary separation loss, and for *Direct*, we have proposed joint training of the disentangled feature extraction and MSS based on the reconstruction with the disentangled music similarity features. We have conducted experimental evaluations and have demonstrated that the end-to-end fine-tuning for *Cascade* improves InMSRL performance, the multi-task learning for *Direct* is also helpful to improve disentanglement performance in the feature extraction and *Cascade* with the end-to-end fine-tuning outperforms *Direct* with the multi-task learning. Future work includes using raw-audio-songs and vocal sound.

Acknowledgment

This work was partly supported by JST CREST under Grant Number JPMJCR19A3.

References

[1] M. Balabanović and Y. Shoham, "Content-based, collaborative recommendation," *Association for Computing Machinery*, vol. 40, no. 3, pp. 66–72, 1997.

[2] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. in Artif. Intell.*, 2009, 19 pages.

[3] A. Elbir and N. Aydin, "Music genre classification and music recommendation by using deep learning," *Electronics Letters*, vol. 56, no. 12, pp. 627–629, 2020.

[4] M. Won, S. Oramas, O. Nieto, F. Gouyon, and X. Serra, "Multimodal metric learning for tag-based music retrieval," in *IEEE ICASSP*, 2021, pp. 591–595.

[5] J. Singh and V. K. Bohat, "Neural network model for recommending music based on music genres," in *ICCCI*, 2021, pp. 1–6.

[6] M. S. Fathollahi and F. Razzazi, "Music similarity measurement and recommendation system using convolutional neural networks," *International Journal of Multimedia Information Retrieval*, vol. 10, pp. 43–53, 2021.

[7] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech & Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[8] A. Holzapfel and Y. Stylianou, "Musical genre classification using nonnegative matrix factorization-based features," *IEEE Trans. Audio, Speech, & Language Processing*, vol. 16, no. 2, pp. 424–434, 2008.

[9] Y. Hashizume, L. Li, and T. Toda, "Music similarity calculation of individual instrumental sounds using metric learning," *APSIPA ASC*, pp. 33–38, 2022.

[10] 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.

[11] 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*, vol. 5, no. 50, p. 2154, 2019.

[12] A. Veit, S. Belongie, and T. Karaletsos, "Conditional similarity networks," in *IEEE CVPR*, 2017, pp. 1781–1789.

[13] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Disentangled multidimensional metric learning for music similarity," in *IEEE ICASSP*, 2020, pp. 6–10.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.

[15] 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 *ISMIR*, 2017, pp. 23–27.

[16] E. Manilow, G. Wichern, P. Seetharaman, and J. L. Roux, "Cutting music source separation some slakh: A dataset to study the impact of iraining data quality and quantity," in *IEEE WASPAA*, 2019, pp. 45–49.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 13 pages, 2014.

[18] F. R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *Latent Variable Analysis and Signal Separation*, 2018, pp. 293–305.