Learning disentangled representations for instrument-based music similarity

Yuka Hashizume

Graduate School of Informatics, Nagoya University, Nagoya, Japan

Li Li

Information Technology Center, Nagoya University, Aichi, Japan,

Atsushi Miyashita

Graduate School of Informatics, Nagoya University, Nagoya, Japan, Tomoki Toda
Information Technology Center, Nagoya University, Nagoya, Japan

March 24, 2025

Abstract

A flexible recommendation and retrieval system requires music similarity in terms of multiple partial elements of musical pieces to allow users to select the element they want to focus on. A method for music similarity learning using multiple networks with individual instrumental signals is effective but faces the problem that using each clean instrumental signal as a query is impractical for retrieval systems and using separated instrumental sounds reduces accuracy owing to artifacts. In this paper, we present instrumental-part-based music similarity learning with a single network that takes mixed sounds as input instead of individual instrumental sounds. Specifically, we designed a single similarity embedding space with disentangled dimensions for each instrument, extracted by Conditional Similarity Networks, which are trained using the triplet loss with masks. Experimental results showed that (1) the proposed method can obtain more accurate feature representation than using individual networks using separated sounds as input in the evaluation of an instrument that had low accuracy, (2) each sub-embedding space can hold the characteristics of the corresponding instrument, and (3) the selection of similar musical pieces focusing on each instrumental sound by the proposed method can obtain human acceptance, especially when focusing on timbre.

1 Introduction

Today's music market is proliferating, with digital products and online services at its core. In 2023, sales from music streaming services increased by 10.4% from the previous year to 19.3 billion dollars, and streaming services account for 67.3% of the music industry's sales share [1]. The number of musical pieces available on music streaming services is about 100 million today [2]. Subsequently, it is impossible for users to listen to all of them to find their favorite music. Therefore, Music Information Retrieval (MIR) technologies, such as music recommendation systems, are needed to help users find their favorite music efficiently.

Research in the field of MIR involves several tasks, mainly the retrieval, recommendation, and estimation tasks to obtain information useful for the retrieval or recommendation, such as tagging, genre classification, melody extraction, and cover song identification. Among these tasks, our focus in this study is to help users find new favorite musical pieces given the background described above. Within the retrieval task, research aimed at such a purpose is ill-defined because it does not have an objective answer unlike cover song retrieval or version retrieval. Hence, various approaches are possible depending on the criteria considered to retrieve or recommend music.

The most common way to access music online today is through text-based metadata retrieval. However, metadata retrieval has limitations in expressiveness when using objective information such as the artist name and publication year [3]. On the other hand, music recommendation is a typical technique for users to efficiently discover new favorite music. There are several main approaches in music recommendation [4]: an approach using user information [5], [6], a content-based approach [7]–[9], and a combination of the two [10]–[12]. Many studies have been conducted using users' listening history, and one of the most representative approaches is using collaborative filtering [13]. In collaborative filtering, it is assumed that users who have similar ratings or the same behavior toward a certain content will have similar ratings toward other contents. This enables the prediction of the ratings of unknown tracks on the basis of the ratings of other users who have engaged in similar behavior. However, a limitation is that newly released music may receive few recommendations until a certain amount of listening history has been recorded. Another problem is that less well-known music may not be recommended as often since popular music generally receives more ratings.

The content-based approach has the potential to avoid the problems of metadata-based retrieval and collaborative filtering-based recommendation, which uses the features of the content itself for recommendation and retrieval. Content-based methods for recommendation, retrieval, and related MIR tasks have been investigated for a long time [3] and have traditionally included signal processing, handcrafted features, and classical machine learning methods [14]–[18]. Fujishima [14] applied pattern matching to acoustic features, Logan and Salomon [16] applied clustering to acoustic features, and Tzanetakis and Cook [15] introduced several handcrafted features for genre classification. Whitman and Rifkin [17] proposed the query-by-description method using a classical machine learning method, and Gómez [18] presented a method to extract a tonal description from audio signals. With the advent of deep learning, data-driven feature extraction has been shown to be effective in improving the performance of MIR systems [19]–[22]. Hamel and Eck [19] showed that using the features extracted from Deep Belief Networks is better than using the Mel-Frequency Cepstrum Coefficient (MFCC) in the classification tasks, and Elbir and Aydin [20] also showed that deep learning methods outperform classical machine learning methods in accuracy for genre classification tasks. Furthermore, the effectiveness of using Convolutional Neural Networks (CNNs) was shown [8], [23]–[33]. These studies include a method for a recommendation system [8], an automatic musical instrument identification method [23], [27], [33], an autotagging method [26], [31], a music classification method [24], [28], [29], [31], a representation learning method [30], [32], and analysis of mechanisms [25].

One effective method for content-based music recommendation or retrieval is to define similarities between musical pieces and use the user's favorite piece as a query for retrieval or recommendation. This method requires designing suitable similarity criteria for calculating music similarity, and many MIR researchers discussed this issue [9], [12], [16], [30], [32], [34]–[45]. Aucouturier and Pachet [34] introduced similarity measures based on a Gaussian model of the cepstrum coefficient. Cheng et al. [44] showed some acoustic features related to human per-

ceptions. Moreover, several methods based on machine learning and algorithmic approaches were proposed: a method based on classification [35], string matching [36], learning binary codes for music representations [39], and a path-based music similarity measure [45]. Urbano et al. [37] analyzed the reliability of the results in the evaluation of music similarity. Some data-driven methods for calculating similarity were proposed: using a classification model [9], [38], the metric learning [12], [40], [43], and transfer learning [38], [41]. McFee et al. proposed a training method with sampling using collaborative filter data [12]. Furthermore, some methods were proposed to learn feature representation by deep metric learning using labels or tags [30], [32], [42]. In these methods, music similarity is calculated by evaluating a mixture of various sounds using a single criterion. However, music has a complex structure with various significant elements, and what users focus on when listening to music varies from user to user. The MIR system with music similarity regarding multiple partial elements in musical pieces enables users to select the element they want to focus on and flexibly search music.

We previously proposed a music similarity learning method based on each instrumental part, where networks are trained for each instrument using single instrumental signals with deep metric learning [46]. One limitation of this method is the need for individual instrumental sounds not only in training but also in inference, where clean instrumental sounds in the query piece the users want to input are usually not publicly available. We also investigated the use of instrumental sounds separated from mixed sounds, but using the separated sounds resulted in lower accuracy than using the original instrumental sounds owing to artifacts.

Another potential approach is to extract feature representation for each instrumental part directly from the mixed musical piece. Disentanglement feature representation learning is one such method that can extract several different conceptual representations from a single input [47], which has been adopted to disentangle the speaker identity and noise in the speech domain [48], [49], timbre and pitch information in the music domain [50]–[52], and so on. Veit et al. [53] proposed Conditional Similarity Networks (CSNs) in the image domain, which learn embeddings differentiated into semantically distinct subspaces that capture the different notions of similarities. Lee et al. [54] applied CSNs to the music domain and designed an embedding space such that each subspace represents the four similarity metrics: genre, mood, instrumentation, and tempo.

In this paper, we propose a method for calculating an instrumental-part-based music similarity in one network using mixed sounds as input by employing CSNs. The proposed network is trained with deep metric learning to embed mixed musical pieces into differentiated feature space, where each subspace selected by a binary mask represents a musical feature when focusing on a particular instrumental part. To successfully train the network, we implement new ideas for the training, such as the use of pseudo musical pieces, a norm loss, and pre-training. In the experiments, we investigate whether more accurate feature representations can be obtained using our proposed method than using conventional methods, whether each subspace holds the characteristics of the assigned instrument sounds, and whether the learned similarity criterion matches human perception.

2 Related Work

2.1 Instrumental-part-based similarity with individual networks

In deep metric learning with a triplet loss [55], a distance metric is trained with a triplet of samples, where one is considered as an anchor, and the other two are considered as positive and negative samples. Here, the positive sample should have a higher similarity to the anchor

than the negative one does. Lee et al. [54] proposed track-based music similarity learning; segments divided from the same track as the anchor are defined as positive samples, and those from different tracks from the anchor are defined as negative samples.

To achieve a highly flexible MIR system, we proposed a music similarity learning method that focuses on each instrumental part [46]. In this method, metric learning with triplet loss is applied to individual instrumental sources such as drums, bass, piano, and guitar, unlike in previous methods [54]. Positive and negative samples are defined by track-based similarity. Different networks are separately trained for individual instrumental sounds. This method requires individual instrumental sounds not only in training but also in inference, although clean instrumental sounds in the query piece the users want to input are usually not publicly available. Thus, we applied instrument signals separated from the mixed music signals through a music source separation method [56] to this method for evaluation in practical use.

Letting $x_i^{(a)}$, $x_i^{(p)}$, and $x_i^{(n)}$ denote the i-th anchor, positive sample, and negative sample, respectively, we constructed the i-th triplet as a set of $\{x_i^{(a)}, x_i^{(p)}, x_i^{(n)}\}$, where $i = 1, \ldots, I$ denotes the index of training samples. The triplet loss is defined as

$$
\begin{aligned}
&\mathcal{L}_{\mathrm{triplet}}(x_i^{(a)}, x_i^{(p)}, x_i^{(n)}) \\
&= \max\{d(x_i^{(a)}, x_i^{(p)}) - d(x_i^{(a)}, x_i^{(n)}) + \delta, 0\},
\end{aligned}
\tag{1}
$$

where d is a distance function for measuring the distance between two audio samples, such as the Euclidean distance, and δ is a margin value, which defines the minimum distance between the positive and negative samples.

2.2 CSNs

To measure the similarity between images considering multiple notions of similarity, Veit et al. [53] proposed CSNs that learn embeddings differentiated into semantically distinct subspaces that capture the different notions of similarities. In the example where the input is an image of a shoe, the notions of similarity are, for example, the height of the shoes' heels and the suggested gender of the shoes.

In this method, a network extracting an embedding representation is trained by the triplet loss using masks. For the triplet loss, samples $x^{(a)}$, $x^{(p)}$, and $x^{(n)}$ are selected according to condition c that is defined as a certain notion of similarity. Namely, in the notion corresponding to condition c, $x^{(p)}$ is more like $x^{(a)}$ than $x^{(n)}$. To disentangle the embedding space, a mask is applied to all dimensions except the dimension corresponding to the notion to be considered in the triplet loss calculation. The network is given by function $f(\cdot)$, and $\mathbf{m}_c$ is a mask that activates only the dimension corresponding to condition c. The masked distance function between two images x_i and x_j is given by

$$
d(x_i, x_j; \mathbf{m}_c) = \| f(x_i)\mathbf{m}_c - f(x_j)\mathbf{m}_c \|_2 .
\tag{2}
$$

Thus, the triplet loss can be written as

$$
\begin{aligned}
&\mathcal{L}_{\mathrm{triplet}}(x^{(a)}, x^{(p)}, x^{(n)}, c) \\
&= \max\{d(x^{(a)}, x^{(p)}; \mathbf{m}_c) - d(x^{(a)}, x^{(n)}; \mathbf{m}_c) + \delta, 0\}.
\end{aligned}
\tag{3}
$$

Lee et al. [54] proposed the disentangled multidimensional metric learning for music similarity using CSNs. They used musical genre, mood, instrument, and tempo for the notions of similarity.
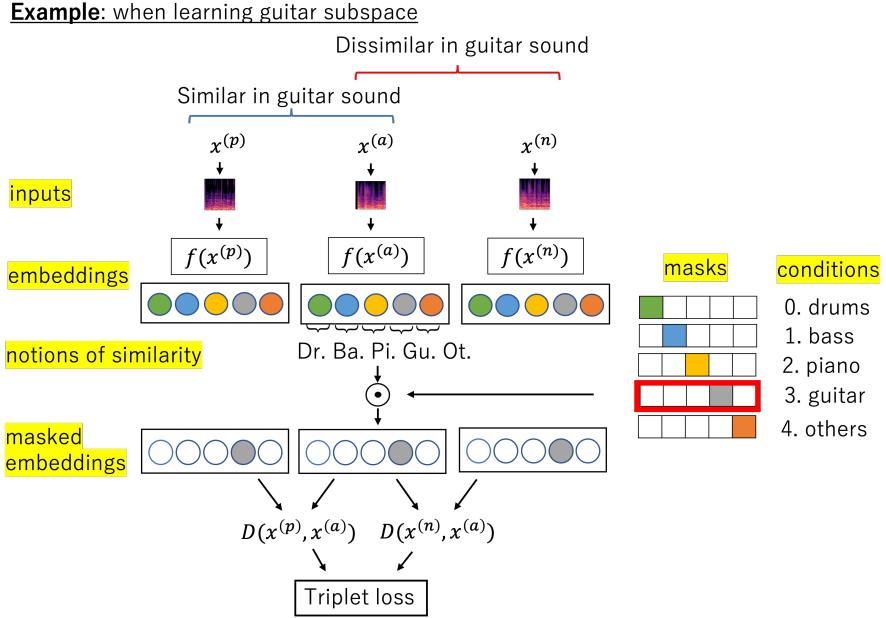
Figure 1: Overview of the proposed method. $x^{(a)}$, $x^{(p)}$, and $x^{(n)}$ denote the anchor, positive, and negative samples, respectively. "Dr.," "Ba.," "Pi.," "Gu.," and "Ot." are drums, bass, piano, guitar, and others, respectively. This figure shows an example of setting the condition to $c = 3$, i.e., similarity focusing on the guitar part, where an anchor sample $x^{(a)}$ and a positive sample $x^{(p)}$ are similar, and an anchor sample $x^{(a)}$ and a negative sample $x^{(n)}$ are dissimilar when focusing on the guitar part. From each sample, the embedded representation is extracted by the network and is masked so that the subspace to which the guitar is assigned only validates in the triplet loss calculation.

3 Proposed method

3.1 Triplet loss with mask

In this study, the CSNs described in Section 2.2 are used, with each notion of similarity defined as each instrumental-part-based similarity. We define c as the condition where $c = 0, 1, 2, 3, 4$ represent the similarity based on drums, bass, piano, guitar, and others, respectively, to design an embedding representation space whose subspace represents the semantic distance that focuses on each instrumental part. Letting D be the number of dimensions of a subspace assigned for one instrumental part, the subspace of the embedding assigned to condition c is $f(x)[cD : (c + 1)D - 1]$, where $f(x)$ is an output of the network. The following formula defines each element of a $(5 \times D)$-dimensional vector $\mathbf{m}_c$ as a mask that keeps the subspace corresponding to c and sets the other dimensions to 0, with k being the dimension index.

$$m_{ck} = \begin{cases} 1, & (cD \leqq k < (c + 1)D) \\ 0, & (\text{otherwise}). \end{cases} \tag{4}$$

The triplet loss in CSNs shown in Equation 2 is used for training with the mask described above. An overview of the proposed method is shown in Figure 1.

3.2 Norm loss

In our method, it is required to prevent any leakage of features of the instrumental parts to the unassigned subspaces. When the input music does not contain some instrumental parts, we add the constraint to output a value close to a zero vector in the subspace corresponding to those instrumental parts. This constraint at least ensures that if the input does not contain some instrumental parts, the corresponding subspaces do not contain any values calculated from other instrumental parts' signals.

We use the Binary Cross Entropy Loss (BCELoss) to satisfy this constraint. The input of the BCELoss is a five-dimensional vector $\mathbf{p}$, whose values are calculated from the norm of each subspace. Each value of $p_j, (j = 0, 1, 2, 3, 4)$ is calculated by taking the logarithm to the norm of the masked embedding $f(x)\mathbf{m}_c, (c = j)$ and then adding a learnable parameter b:

$$p_j = \sigma(\log(||f(x)\mathbf{m}_j||_2) + b_j), \tag{5}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$. The target is a five-dimensional multi-hot vector $\mathbf{q}$ that is set to 1 if each instrumental sound is included in the input and 0 if not:

$$q_j = \begin{cases} 1, & (P_{\mathrm{avg}}(x^{(j)}) > threshold) \\ 0, & (\text{otherwise}), \end{cases} \tag{6}$$

where P_{avg} means a time average power of the signal, and $x^{(j)}$ is a clean individual instrumental signal, where the subscript represents each instrument. When $\mathbf{p}$ computed from the i-th anchor $x_i^{(a)}$ is denoted as $\mathbf{p}_i^{(a)}$, and in the same way for a positive sample and a negative sample, the formulation of the norm loss is as follows.

$$\begin{aligned} &\mathcal{L}_{\mathrm{norm}}(x_i^{(a)}, x_i^{(p)}, x_i^{(n)}) \\ &= \frac{1}{3}\{BCE(\mathbf{p}_i^{(a)}, \mathbf{q}_i^{(a)}) \\ &\quad + BCE(\mathbf{p}_i^{(p)}, \mathbf{q}_i^{(p)}) \\ &\quad + BCE(\mathbf{p}_i^{(n)}, \mathbf{q}_i^{(n)})\}, \\ &BCE(\mathbf{p}, \mathbf{q}) = \frac{1}{5}\sum_{j=0}^{4}\{q_j \log p_j + (1-q_j)\log(1-p_j)\}. \end{aligned} \tag{7}$$

This procedure is shown in Figure 2. Note that we use not only the observed silent sections for each instrumental part but also the created silent sections, i.e., we arbitrarily mix only some instruments from the instruments contained in a musical piece to equalize the total time of each instrumental sound included in the train data.

3.3 Training network

The final loss function $\mathcal{L}$ is as follows, where λ is the hyperparameter that weights the loss function $\mathcal{L}_{\mathrm{norm}}$. Note that this loss function is averaged within the mini-batch.

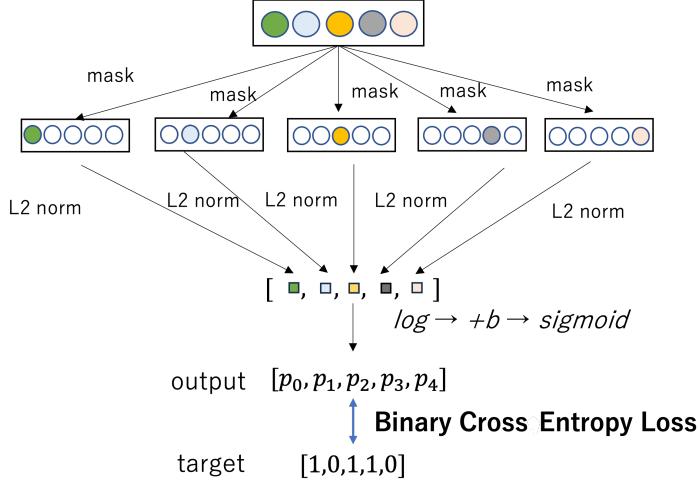$$\mathcal{L} = \mathcal{L}_{\mathrm{triplet}} + \lambda\mathcal{L}_{\mathrm{norm}} \tag{8}$$

Figure 2: Procedures for calculating norm loss. This is an example of when a sound containing only drums, piano, and guitar is input, where the bass's subspace and other's subspace are trained to be close to the zero vector.

3.4 Pseudo musical piece

As mentioned in Section 3.1, we need to sample triplets for each notion of similarity, i.e., a similar/dissimilar pair focusing on each instrumental part. However, there is no label that evaluates whether musical pieces are similar to each other focusing on each instrumental part. We can use the track information as in the previous study [46] as illustrated in Figure 3, but we found that using it in our proposed framework leads to all subspaces being trained on the same criteria.

To successfully train the disentangled subspaces, we propose a method to create a pseudo musical piece for input by mixing instrumental sounds in different musical pieces. For example, when the drum sound contained in piece A is called drum sound A, a pseudo musical piece can be created by mixing the drum sound A with other instrumental sounds from another piece B. We denote this piece's label as $(A, B)^{(dr,else)}$ and also call this piece with the drums label A. By this method, we can create a pair such as one that has the same drum label but different guitar labels.

3.4.1 Basic triplet

Considering that musical pieces with the same label for a particular instrument are similar to each other on that instrument, a triplet sample can be created. We can say that segment 1, randomly extracted from the piece with label $(A, B)^{(dr,else)}$, and segment 2, randomly extracted from the piece with label $(A, C)^{(dr,else)}$, are similar in drum sounds but dissimilar in sounds other than drums. On the other hand, segment 1 (with label $(A, B)^{(dr,else)}$) and segment 3 with label $(D, B)^{(dr,else)}$ are dissimilar in drum sounds but similar in other sounds. Therefore, samples with label $\{(A, B)^{(dr,else)}, (A, C)^{(dr,else)}, (D, B)^{(dr,else)}\}$ can be used as an anchor, a positive sample, and a negative sample in learning with condition $c = 0$. We can also sample the triplets for other conditions $c = 1, 2, 3, 4$ in the same way. The triplets extracted in this way are called the basic triplets.
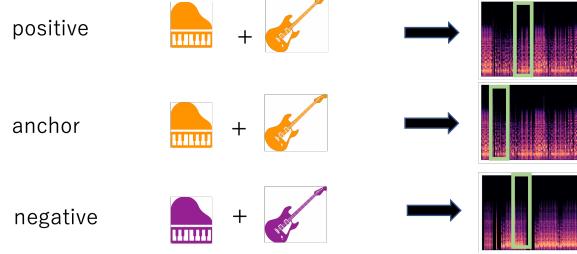
Figure 3: How to create a triplet using track information with the dataset pieces instead of pseudo musical pieces. We show the example of combining only some instruments for the calculation of the norm loss.
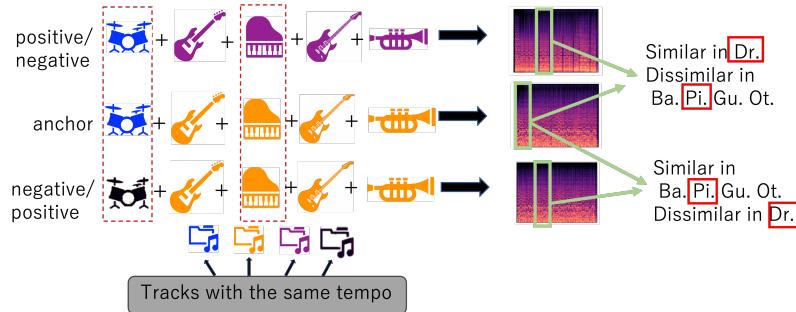

Figure 4: How to create the basic triplet and the additional triplet with pseudo musical pieces. When these triplet samples are input, two losses are calculated: a loss where the upper sample is calculated to be close to the anchor in the drum space, and a loss where the lower sample is calculated to be close to the anchor in the piano space.

3.4.2 Additional triplet

We further add triplets of interchanged positive and negative samples under a condition other than that of the basic triplet to allow each subspace to explicitly learn a different similarity criterion. The anchor and the negative sample in the above example, the segments from pieces with labels $(A, B)^{(dr,else)}$ and $(D, B)^{(dr,else)}$, are dissimilar in drum sounds but similar in instrumental sounds other than drum sounds. Thus, samples with label $\{(A, B)^{(dr,else)}, (D, B)^{(dr,else)}, (A, C)^{(dr,else)}\}$ can be used as an anchor, a positive sample, and a negative sample in learning with condition $c \neq 0$. By randomly selecting c from those that are different from the basic triplet, an additional triplet is created for each basic triplet.

These triplet extraction methods are shown in Figure 4. Note that the basic triplet's negative sample is selected so that there is no conflict between the basic triplet and the additional triplet.

3.5 Pre-training

We introduce pre-training to start training from a better initial value than a random value, enabling the above learning methods to be effective. We train the network with the Mean Squared Error Loss (MSELoss) using the output of the individual instrumental-part-based similarity network [46] as the ground truth for each subspace. Namely, the concatenation of $g_j(x^{(j)})$ is the target of pre-training, where $g_j(\cdot), (j = 0, 1, 2, 3, 4)$ are denoted as the individual networks corresponding to drums, bass, piano, guitar, and others. For the same reasons explained in Section 3.2, the ground truth sub-embedding is set to the zero vector if the input sample does not contain the corresponding instrumental sound. Figure 5 shows a way of creating the tar-
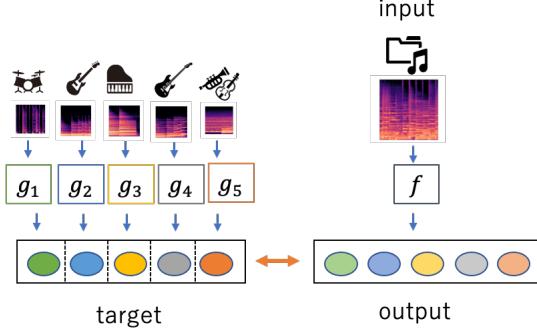
Figure 5: Procedures for pre-training. The network f is trained so that its output is close to the target by the MSE loss.
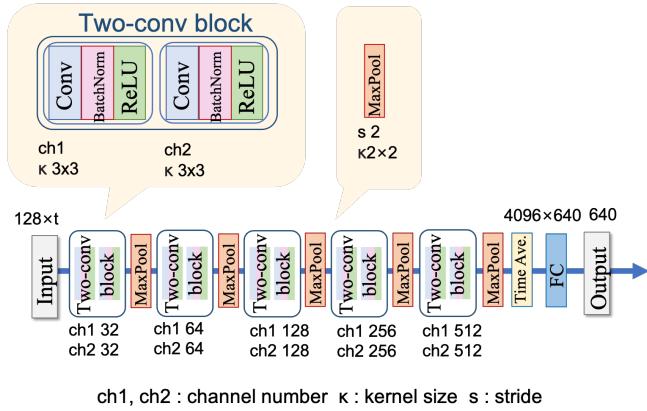

ch1, ch2 : channel number κ : kernel size s : stride

Figure 6: Network architecture. "ch1" and "ch2" denote the channel number, and "κ" and "s" denote kernel size and stride, respectively. "Conv" and "FC" denote the convolutional and fully connected layers, respectively. "Time Ave." means to take an average in the time direction. The numbers above input, output, and "FC" are their sizes. "t" is calculated by multiplying the number of seconds by the sampling rate and dividing it by the hop length.

get embeddings. $x_i^{(j)}$ is the individual instrumental sound segment of instrument j contained in the i-th mixed sound segment x_i. $\frac{\mathbf{y}_i}{\|\mathbf{y}_i\|_2}$ is the target embedding for the network training, which is created by concatenating embedding features extracted from $x_i^{(j)}, (j = 0, 1, 2, 3, 4)$ using the individual networks and divided by the norm. The formulations of the loss function of pre-training $\mathcal{L}_{\mathrm{pre}}$ and the target embedding are as follows:

$$\mathcal{L}_{\mathrm{pre}}(x_i) = \left(f(x_i) - \frac{\mathbf{y}_i}{\| \mathbf{y}_i \|_2} \right)^2,$$
$$y_{ik} = g_j(x_i^{(j)})_k, (jD \leqq k < (j+1)D). \tag{9}$$

This loss function is averaged within the mini-batch.

4 Experimental evaluation

4.1 Experimental conditions

4.1.1 Dataset and input features

The dataset we used is Slakh2100 [57], which contains non-vocal musical pieces and their stems. Following Slakh's recipe, individual instrumental sounds, namely, drums, bass, piano, and guitar sounds, were created from their stems, and the stems that did not fit into any of the four instruments were mixed as "others." We used the *redux* subset of Slakh2100, which is created by omitting some tracks so that each MIDI file only occurs once. The redux subset has a total of 1710 tracks: 1289 in train, 270 in validation, and 151 in test.

We used the 200 musical pieces and their instrumental sounds in the training set in pre-training, both for training the proposed network with $\mathcal{L}_{\text{pre}}$ and training individual networks [46]. Moreover, the pseudo musical pieces mentioned in Section 3.4 were created with the instrumental sounds contained in the 1200 pieces in the training set for training with $\mathcal{L}_{\text{triplet}}$. The 270 musical pieces in the validation set were used for validation. The Slakh test set was used for testing, but musical pieces with the shortest non-silent sections in individual instrument tracks were excluded one by one until 10% of the musical pieces were removed, after which 136 pieces were used.

When creating the pseudo musical pieces, the data set was classified into 36 classes according to tempo, and the instrumental sounds contained in musical pieces belonging to the same tempo group were allowed to be mixed together to preserve the music-like nature of the music. Under this rule, multiple different pseudo musical pieces containing the same instrumental sound were generated. The 5000 triplet pseudo musical pieces were randomly created every epoch using 1200 musical pieces for metric learning with $\mathcal{L}_{\text{triplet}}$.

Both the dataset pieces and pseudo musical pieces were divided into three-second segments for pre-training and training with 50% overlap, three-second segments without overlap for validation, and 3-, 5-, and 10-second segments without overlap for testing. All segments were converted to dB-scaled mel-spectrograms with 2048 window length and 512 hop length, normalized, and used as input for the training, validation, and testing.

4.1.2 Network

We used the network shown in Figure 6, which had 10 convolutional layers with batch normalization and ReLU, and Max pooling applied every two convolutional layers. The encoder portion of U-Net [58], [59] was referenced. This network was trained to extract a 640-dimensional embedding vector from a mel-spectrogram as embedding representations. The 640-dimensional embedding representation was aimed to have 128-dimensional subspaces assigned to each of the five instruments. The subspaces were assigned to drums, bass, piano, guitar, and others in order of increasing dimensions.

4.1.3 Pretraining conditions

For each instrument, a Convolutional Network was trained as an individual network as in the previous study [46]. Then, we pre-trained the network shown in Figure 6 using the concatenations of outputs of trained individual networks as targets and mixtures of instrumental segments as inputs. We used the clean instrumental sound segments as input for the individual networks both in training and inference to create the target features.

4.1.4 Training conditions

The weighting parameter λ between two losses $\mathcal{L}_{\text{triplet}}$ and $\mathcal{L}_{\text{norm}}$ was set to 0.1. The margin of the triplet loss function, the number of epochs, and the batch size were set to 0.2, 1000, and 32, respectively.

4.1.5 Baseline model

We used our conventional method [46] as the baseline model. In our previous study, we tried using both clean instrument sounds and separated instrument sounds as input, but in order to make the conditions the same as in this study, where mixed sounds are used as input, we used the method of using separated sounds obtained by separating mixed sounds as the baseline. Following the previous study, we used the Spleeter [56] for the sound source separation method and used the separated sounds as input for both training and inference.

4.2 Evaluation method

We conducted experimental evaluations to investigate whether the following purposes of this study were achieved: (P1) to learn an embedding representation in which similar pieces are close and dissimilar pieces are far from each other more accurately than the conventional method, (P2) to output the similarity focusing on each instrumental part in the subspace assigned to each instrument, (P3) to ensure that the constraints imposed to be satisfied during training are also satisfied during inference, and (P4) to learn similarity criteria corresponding to human perception.

4.2.1 Accuracy of feature representation

In the evaluation on P1, we used the accuracy of music ID prediction with the dataset pieces in the same manner as the evaluation in the previous study [46]. This evaluation was based on the assumption that instrumental sounds that consist of different time segments of the same musical piece should be more similar than those of different musical pieces. Note that subjective evaluation experiments later confirmed whether this assumption fits the human senses. Specifically, we used the K-nearest neighbor (kNN) method to predict the music IDs of the test segments' representations. Let a segment to be predicted be called the target, and the music IDs of all test segments' representations except the target were assumed to be known. We predicted the music ID of each representation by a majority vote using the IDs of the top five nearest test segments' representations. To evaluate each instrument's representation, we extracted it by using only the corresponding subspace with masking, inputting the dataset pieces with our proposed method. For the evaluation of the conventional method, the same musical pieces were inputted into the source separation model [56] according to the previous study [46], and the separated signals were inputted into each individual network to extract each instrument's representations.

4.2.2 Capability to represent disentangled features

We evaluated the accuracy of the feature representation of each subspace in Section 4.2.1 but did not evaluate whether each subspace is disentangled by the instrument. The inputs in the evaluation in Section 4.2.1 had the same label for all instruments because they were the dataset pieces. For example, if a piano feature leaked into the drum space, we were still able to predict the correct drum label using that feature.

In the evaluation on P2, the pseudo musical pieces were created for the test pieces by the same method as described in Section 3.4. The kNN-based label prediction by the same method as described in Section 4.2.1 was conducted for the pseudo test pieces, where not only the target but also other segments divided from the same pseudo piece as the target were removed from the reference. For example, in the evaluation of the drums subspace, the test samples with the same drum label and different other instrument labels were created, as $\{(A, B)^{(dr,else)}, (A, C)^{(dr,else)}, (A, D)^{(dr,else)}, (A, E)^{(dr,else)}, (F, G)^{(dr,else)},$ $(F, H)^{(dr,else)}...\}$. With this test set, the correct drum label of a segment from the piece with the label $(A, B)^{(dr,else)}$ is "A" but other segments from the piece with $(A, B)^{(dr,else)}$ cannot be referred. Hence, only when segments from $(A, C)^{(dr,else)}$ or $(A, D)^{(dr,else)}$ or $(A, E)^{(dr,else)}$ were close to the target (even though they have different labels except for drums), the prediction works well. If the drum subspace contains the piano's feature, the prediction is affected by that and can be wrong. This is because segments that are similar on the piano to the target but have different drum labels come close to the target, and the drum label of that is output as a prediction. This can detect the leak and correctly evaluate the capability to represent disentangled features. We created 40 pseudo musical pieces with 10 labels for each instrument; in other words, four different pseudo musical pieces per label, and divided them into segments.

4.2.3 Instrumental sound identification accuracy

To confirm that the training with the norm loss described in Section 3.2 was successful (one of P3), we evaluated it with the instrumental sound identification task. We performed this to verify that the subspace corresponding to instrumental sounds not included in the input was close to the 0 vector, i.e., that the information for each instrumental sound did not leak into the subspace to which it did not correspond.

When an individual instrumental sound was input, a five-dimensional vector was calculated such that the j-th element had the norm of the masked embedding $f(x)\mathbf{m}_j$, and the index with the largest value of the vector was denoted as the prediction of the type of instrumental sound for that input. With each instrumental sound input, we made the above predictions and calculated the percentage of correct answers for identifying the type of input instrument.

4.2.4 Correlation between distance matrices

In our method, $f(x_i^{(j)})$, the output when inputting the instrumental sound $x_i^{(j)}$, and $f(x_i)\mathbf{m}_j$, the masked embedding when inputting a musical piece x_i containing that instrumental sound $x_i^{(j)}$ should represent the same feature. We confirm whether our model had a correlation between these features focusing on similarity relationships between musical pieces (one of P3). We calculated the two distance matrices between representations of all segments of the test set (the dataset pieces) for $f(x_i^{(j)})$ and $f(x_i)\mathbf{m}_j$. Then, we flattened them to single vectors and calculated the correlation between them.

In contrast to the above, a very high correlation between different subspaces may have resulted in a meaningless space that expresses the same feature in all of them when inputting the mixed sound. Correlations between two subspaces when inputting the mixed sound were also calculated to verify the difference between similarity criteria in subspaces (one of P3). Each distance matrix was calculated using $f(x_i)\mathbf{m}_j$, and the correlations between distance matrix pairs were calculated in the same manner as above.

4.2.5 Subjective evaluation

Subjective evaluation experiments through a listening test were conducted to confirm whether each subspace represented a similarity criterion such that the distance between sounds was small enough that humans would perceive them to be similar when listening to each assigned instrumental sound (P4). The procedure of the listening test is as follows.

Subjects were presented with three audio tracks of instrumental sounds, X, A, and B, and listened to all of them. They chose A+ if they perceived A to be more strongly similar to X than B, A− if slightly, B+ if they perceived B to be more strongly similar to X than A, and B− if slightly, on the basis of the following four perspectives: timbre, rhythm, melody, and overall similarity.

In each answer, they can select N/A from up to two perspectives except for *overall*. The following instruction was provided with the subjects; "You can select N/A if A and B are similar/dissimilar to X of equal degree, or the presented instrumental sound has no element corresponding to the perspective; e.g., drums have no melody."

We prepared two types of sample sets, test 1 and test 2. For test 1, we randomly selected three different musical tracks $\{\chi_i, \alpha_i, \beta_i\}$ and randomly captured five-second segments from each instrumental sound contained in the three tracks, respectively. Then we obtained one sample set for each instrument, {X, A, B}={$\chi_i^{(j)}$'s segment, $\alpha_i^{(j)}$'s segmant, $\beta_i^{(j)}$'s segment}. The subscript j represents each instrument ($j = 0, ..., 4$), with 0 representing drums, 1 bass, 2 piano, 3 guitar, and 4 others. For example, $\chi_i^{(0)}$ means the drum sound contained in the musical piece χ_i. This selection was repeated four times ($i = 0, ..., 3$), and 20 sample sets were created. For test 2, we used the same X as in test 1, and one of the other two samples was taken from a different time frame of the same song as X. Namely, we randomly selected γ_i from the test music tracks excluding χ_i and replaced α_i, β_i with χ_i and γ_i (in no particular order). The process was repeated in the same way, and 20 sample sets $\{\chi_i^{(j)}$'s segment 1, $\chi_i^{(j)}$'s segment 2, $\gamma_i^{(j)}$'s segment} were created. It was randomly determined whether this set was {X, A, B} or {X, B, A}. Thus, a total of 40 sample sets were created, and these were used as one listening test set.

This procedure was repeated with a random selection of sample sets, creating a total of 60 listening test sets. Experiments were conducted by recruiting participants through Crowd-Works [60]. The valid answers from a total of 586 subjects (281 unique subjects) were obtained, and for all 60 sets, valid answers from at least six different subjects were obtained.

We calculated whether A or B was closer to X using our proposed model for the same set as that used in the listening test and then calculated the matching rate between the model's results and the subjects' results. The model's results were obtained as follows: The music tracks originally containing the instrumental tracks (A, B, and X) used in the listening test were input into the model, and then the distance was measured by applying a mask that leaves only the subspace corresponding to the target instrument. Then, the result was the one with the smaller distance from A or B to X. On the other hand, the subjects' results were obtained as follows: A+ and A− were treated as the same answer, A. The same applied to B. Sample sets with less than 80% agreement among subjects were eliminated in the evaluation because the sample set with a low agreement rate among the subjects may be equally similar/dissimilar to X for both A and B. If N/A accounts for the largest percentage (even over 80% agreement), that sample set was also eliminated. All answers to the remaining sample set, A or B, were used as the subjects' results.

4.3 Results

4.3.1 Accuracy of embedding

The accuracy of the predicted music IDs is shown in Table 1. The table shows the results when 3, 5, and 10 s of data were used as input for inference. Each row represents the instrument to focus on. The column for the proposed method shows the results of inference using only the subspace to which the focused instrument sound is assigned. In contrast, the column for the conventional method shows the results of inputting the separated instrument sound to the individual networks.

It can be seen that the conventional method using instrumental sound separation is affected by sound quality degradation due to separation, and the accuracy of feature representation degrades, especially on piano with low separation accuracy [46]. In contrast, the proposed method shows stable accuracy regardless of which instrument is focused on. An example of the subspace is shown in Figure 7. It can be seen that segments from the same musical piece constitute a cluster and that the subspace can be learned with different distance relationships between the musical pieces.

	Input: 3 s of data		Input: 5 s of data		Input: 10 s of data	
instrument	proposed[%]	baseline[%]	proposed[%]	baseline[%]	proposed[%]	baseline[%]
drums	84.73	94.00	86.84	95.24	88.91	94.62
bass	51.01	51.78	59.20	61.54	64.87	70.32
piano	77.21	39.40	82.00	43.27	84.30	45.30
guitar	76.50	-	80.18	-	82.70	-
others	82.84	-	83.34	-	82.20	-

Table 1: kNN-based classification accuracy using each feature representation. The lines "proposed" and "separated" show the results using the proposed method and using the separated sounds as input to the individual networks of the conventional method [46].

(a) Drums subspace

(b) Bass subspace

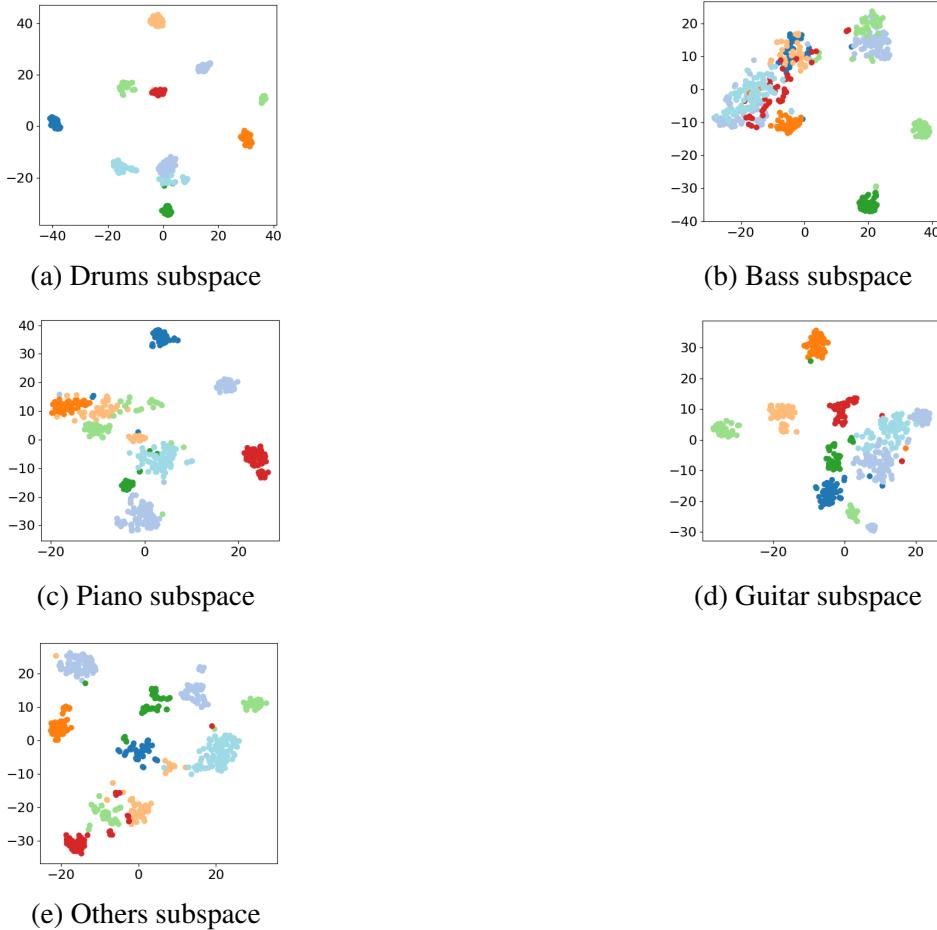(c) Piano subspace

(d) Guitar subspace

(e) Others subspace

Figure 7: Examples of visualized feature representations extracted from five-second segments in the test set of the dataset pieces. We used t-SNE [61] to compress the 640-dimensional feature representations to two-dimensional representations. Different time frames from the same musical piece are plotted with the same color.

	Method						Instrument[%]				
	baseline	norm	psd	basic	add	pre	drums	bass	piano	guitar	others
(a)	✓						90.19	40.00	41.28	-	-
(b)						✓	84.31	34.65	33.72	41.46	71.06
(c)				✓※			83.54	17.88	24.87	29.12	26.14
(d)		✓		✓※			88.99	19.54	26.99	32.18	31.63
(e)		✓		✓※		✓	90.19	14.66	26.73	31.26	42.55
(f)		✓	✓	✓			92.64	55.69	56.09	38.00	65.95
(g)		✓	✓	✓	✓		92.43	60.47	64.04	54.27	76.28
(h)		✓	✓	✓		✓	95.91	63.08	58.97	52.25	80.74
(i)		✓	✓	✓	✓	✓	95.80	61.91	67.37	57.32	80.69

Table 2: kNN-based classification accuracy using the pseudo piece. The baseline is our conventional method [46]. The "norm," "psd," "basic," "add," and "pre" mean using the norm loss, the pseudo musical pieces, the basic triplets, the additional triplets, and the pre-training, respectively. ※When learning without the pseudo musical pieces, the basic triplet sampling method is shown in Figure 3, and the additional triplet cannot be created. Only the results of the five-second segment are shown.

4.3.2 Capability to represent disentangled features

Table 2 shows the evaluation results for each subspace using pseudo musical pieces. We can see that our proposed method, including pre-training, creating the pseudo musical pieces, and training with the additional triplets, is effective, and using a combination of these methods can lead to higher scores than conventional methods.

The result (b) shows that the pre-trained model performed better than random prediction but worse than the conventional method. The results (c)–(e) show that the models trained without the pseudo musical pieces (Figure 3) did not work well even when used with the norm loss and the pre-training. Considering that only the drum score is high, these results suggest that all subspaces represent the drum's feature, which is consistent with the concern raised in Section 3.4 that all subspaces might be learned on the same criteria. We can see that the models trained with the pseudo musical pieces work well as shown in the results (f)–(i), and the additional triplet can improve the accuracy as shown by comparing (f) and (g); the pre-training also can improve the accuracy as shown by comparing (f) and (h), especially in low-accuracy instruments. These results mean that these methods can help disentanglement for each subspace.

All five-second segments divided from the 40 pseudo musical pieces used in the test are plotted and visualized in two dimensions in Figure 8 and Figure 9. It can be seen that the pieces with the same instrumental sound labels are close to each other.

4.3.3 Instrumental sound identification accuracy

Table 3 shows the results of instrumental sound identification accuracy. The pre-training is effective for making the unsounded instruments' subspaces zero vector. Using the norm loss can also improve the accuracy. We can see that only the corresponding subspace retains the values when inputting the individual instrumental sound as in Figure 10.

16

(a) Drums subspace

(b) Bass subspace

(c) Piano subspace

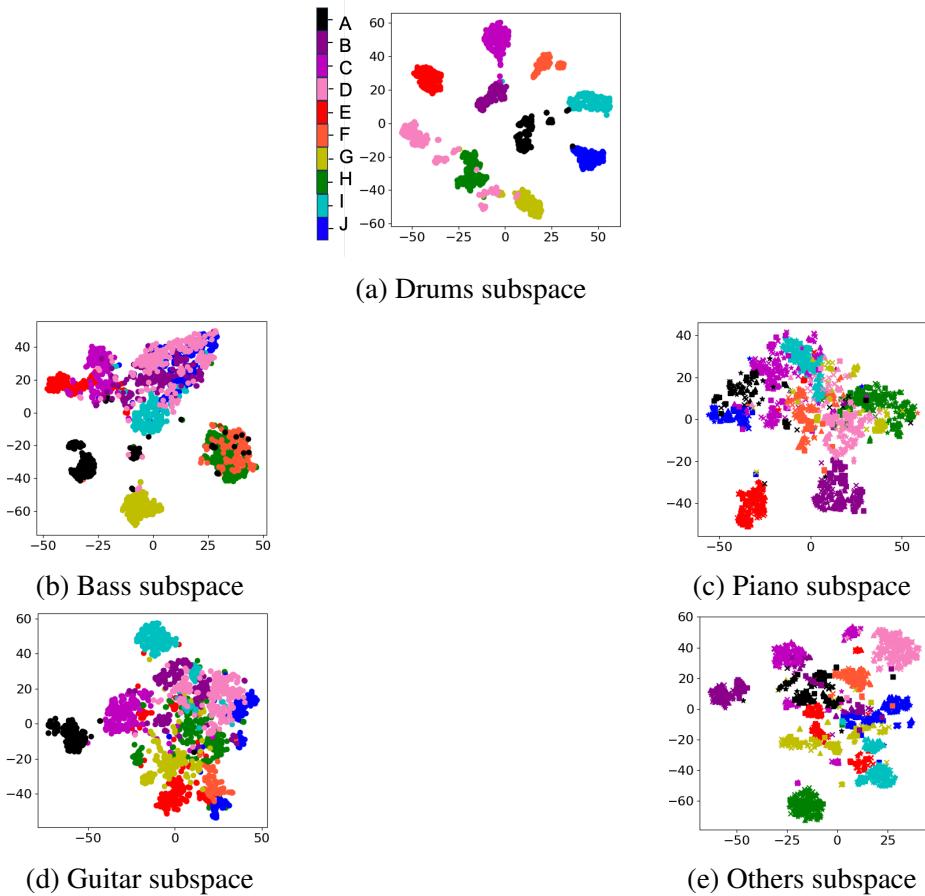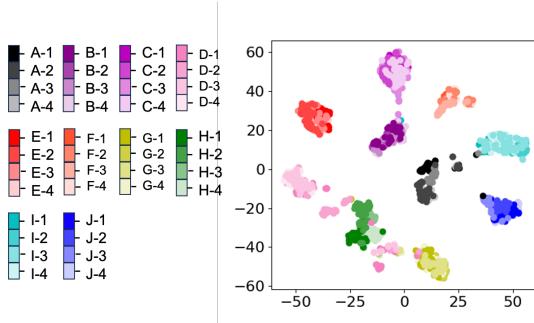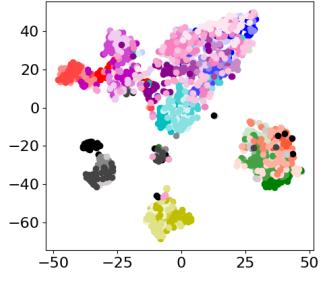(d) Guitar subspace

(e) Others subspace

Figure 8: Examples of visualized feature representations extracted from five-second segments in the test set of the pseudo musical pieces. In each instrument, the same color represents the same instrument label. A set of samples plotted in the same color for one instrument includes samples with different labels for other instruments. For example in (a), a sample with label $(A, B)^{(dr,else)}$ and a sample with label $(A, C)^{(dr,else)}$ are plotted with the same color as they have the same drum label. This also includes samples with exactly the same label, such as $(A, B)^{(dr,else)}$ and $(A, B)^{(dr,else)}$, which are different time frames from the same musical piece. We can see in Figure 9 the visualization when different colors are assigned to samples that have the same instrument label but different labels for other instruments.

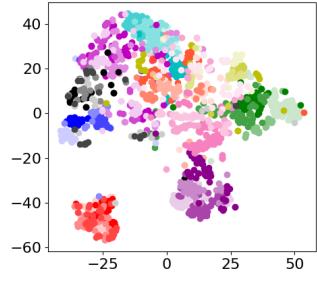|	drums (%)	bass (%)	piano (%)	guitar (%)	others (%)
wo/pre, wo/norm	40.27	0.39	67.00	10.10	10.75
w/pre, wo/ norm	67.94	76.89	78.65	67.01	95.66
w/pre, w/ norm	84.48	96.04	90.27	76.85	92.24

Table 3: Instrumental sound recognition rate when each individual instrument sound is input. "wo/pre" means using pre-training, and "wo/norm" means using norm loss. Each line name can be rephrased corresponding to the results in Table 2 as follows: "wo/pre, wo/norm," to psd+basic+add, "w/pre, wo/norm," to psd+basic+add+pre, and "w/pre, w/norm," to norm+psd+basic+add+pre.
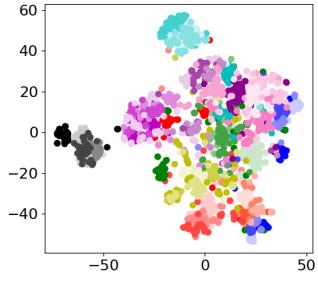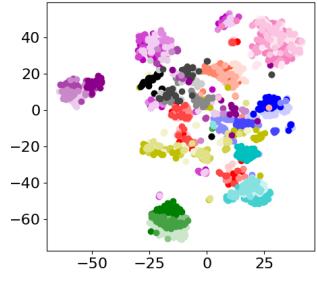
17

(a) Drums subspace

(b) Bass subspace

(c) Piano subspace

(d) Guitar subspace

(e) Others subspace

Figure 9: These are the same results as in Figure 8 but the color coding is different, where only samples with the same label for all instruments are plotted in the same color. For example in (a), a sample with label $(A, B)^{(dr,else)}$ and a sample with label $(A, C)^{(dr,else)}$ have the same drum label but are plotted with different colors. The color map in this figure denotes such labels with A-1, A-2, etc. When evaluating by kNN using the pseudo musical pieces in Section 4.2.2, the same color in Figure 8 was regarded as the same label, but samples with the same color in this figure were not used for reference.

(a) Drums subspace

(b) Bass subspace

(c) Piano subspace
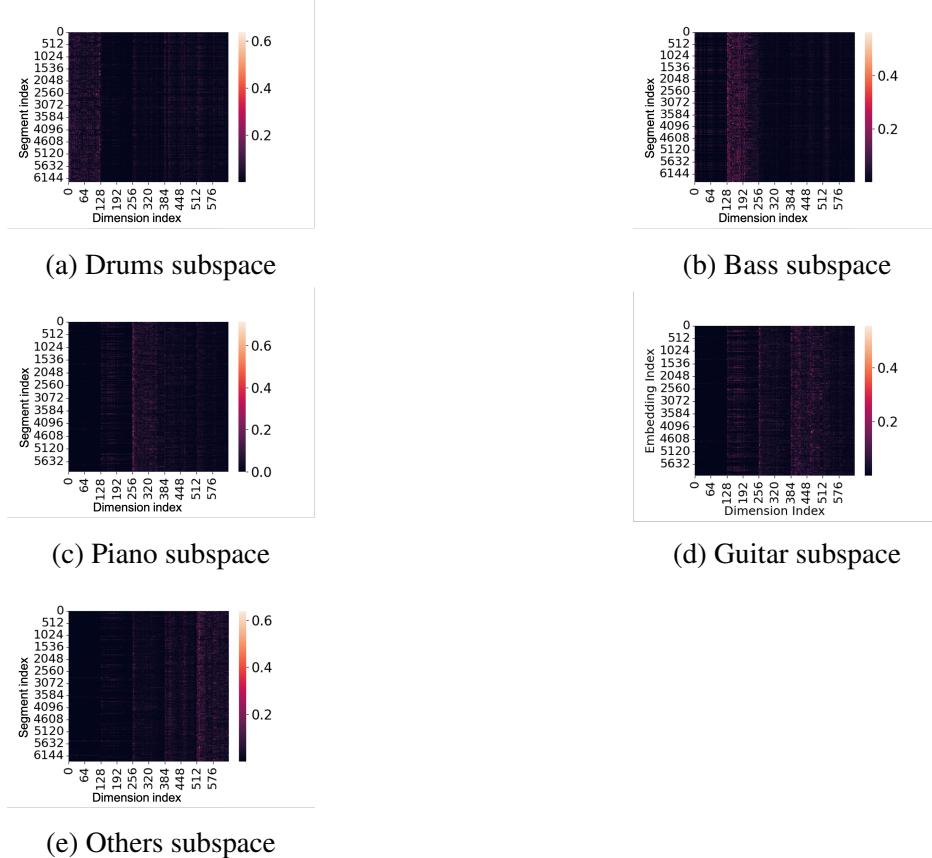
(d) Guitar subspace

(e) Others subspace

Figure 10: For each instrument, absolute values are taken for the output features (without masking) when inputting five-second segments of the individual instrument sounds in the test set, and stacked vertically. The vertical axis is the index of segments, and the horizontal axis is the index of dimensions.

4.3.4 Correlation between distance matrices

Table 4 shows the correlation between the distance matrix when an individual instrument sound is input and the distance matrix when a mixed sound containing that instrument sound is input and masked. It can be seen that the correlation is higher when pseudo musical pieces were used, which suggests that using pseudo musical pieces can help each subspace represent the target instrumental feature. A visualization of the two distance matrices is shown in Figure 11 taking the drums as an example. We can see a similar pattern in the similarity between the musical pieces both when instrumental sounds are input and when mixed sounds containing them are input and masked.

Table 5 shows the correlation between pairs of distance matrices across the subspaces. The results with the individual network with clean individual instrumental sound input, shown for reference, show low correlations, indicating that the correlation of embedding for each instrument should inherently be low. However, without the pseudo musical pieces, the correlation between subspaces is high, indicating that all spaces are learned with the same criteria. The correlation for the proposed method using the pseudo musical pieces is low, which suggests that the proposed method using the pseudo musical pieces can learn the instrument-dependent features in each subspace.

19

	drums	bass	piano	guitar	others
wo/ psd	0.6418	0.2357	0.1651	0.3777	0.2743
w/ psd	0.6159	0.4782	0.3608	0.4115	0.3243

Table 4: Correlation between distance matrices in the same subspace with individual instrumental sound input and with mixed sound input. Each line name can be rephrased corresponding to the results in Table 2 as follows: "wo/psd" to norm+basic+pre, and "w/ psd" to norm+psd+basic+add+pre.

w/ individual clean sound (reference)					
	drums	bass	piano	guitar	others
drums	-	0.000112	0.0133	0.0509	0.0123
bass	-	-	0.0320	0.0567	0.0531
piano	-	-	-	0.135	0.209
guitar	-	-	-	-	0.156
others	-	-	-	-	-
wo/ psd					
	drums	bass	piano	guitar	others
drums	-	0.610	0.359	0.224	0.226
bass	-	-	0.444	0.323	0.434
piano	-	-	-	0.304	0.460
guitar	-	-	-	-	0.246
others	-	-	-	-	-
w/ psd					
	drums	bass	piano	guitar	others
drums	-	-0.0298	-0.100	-0.00900	-0.0929
bass	-	-	-0.0488	0.0309	0.0315
piano	-	-	-	-0.170	0.0697
guitar	-	-	-	-	0.00354
others	-	-	-	-	-

Table 5: Correlation between pairs of distance matrices across the subspaces. The "wo/psd" and "w/psd" represent the same mean as Table 4.
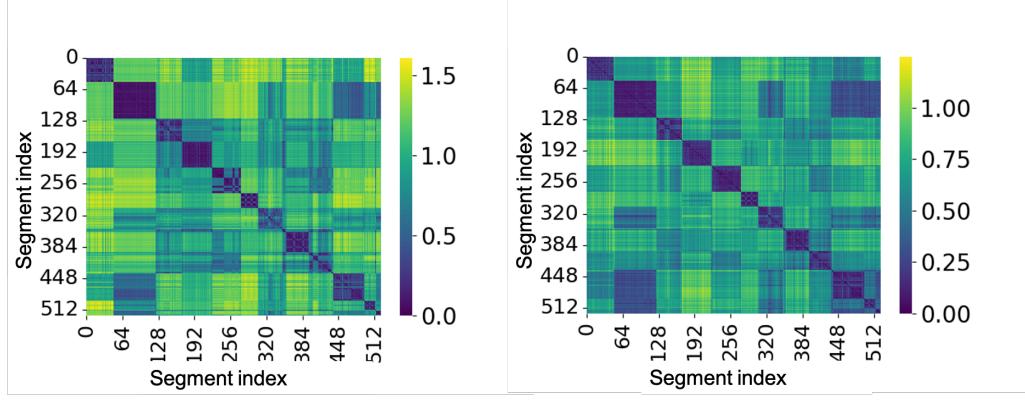
4.3.5 Subjective evaluation

The matching rate with the answers on *overall* was high for *rhythm* and *melody*, but low for *timbre*. In other words, different labels were assigned to the same sample set when focusing on *timbre* compared to focusing on *overall*. Therefore, we evaluated the model using these two types of answers.

The results are shown in Table 6, and the number of answers for each instrument on the two perspectives are shown in Table 7. We can see from Table 7 that there is more agreement among subjects in their answers in test 2 than in test 1. Moreover, there are fewer answers for *timbre* than for *overall* because the subjects cannot select N/A for *overall*, but they can for *timbre*, and here, N/A (after taking agreement) is omitted from the evaluation.

As shown in "set 2" in Table 6, different time segments within the same piece are perceived by humans as similar to each other, and their distances are also small in the distance metric learned by the proposed method.

Compared with the baseline [46], the matching rates of the proposed method for the drums and bass are comparable, and that for the piano is better.

Although test 1 is less accurate than test 2, accuracy is improved in drums, piano, and guitar in the evaluation using answers focusing on timbre compared with using answers focusing on

(a) Using the drums subspace representation with drums sound input

(b) Using the drums subspace representation with mixed sound input

Figure 11: Distance matrices between the representations of segments from musical pieces in the test set. Here, we show an example of 10 musical pieces. This is the result of the model trained using the pseudo musical pieces; "w/psd".

overall. This suggests that the model is trained to represent similarity mainly focusing on timbre. We consider that if we can design a model that captures the structure of the time direction so that melody and rhythm can be considered, it will be possible to obtain a music similarity that is also compatible with human perception when focusing on the overall similarity.

	Evaluation with answers on **overall** (%)				
	drums	bass	piano	guitar	others
test 1					
baseline	57.2±4.9	57.3±5.0	59.5±5.1	-	-
proposed	52.7±4.9	67.5±4.8	58.5±5.1	60.1±4.9	58.3±4.8
test 2					
baseline	94.9±1.6	83.1±2.8	75.5±2.9	-	-
proposed	94.8±1.6	88.6±2.4	87.1±2.3	92.0±1.9	91.6±2.1
	Evaluation with answers on **timbre** (%)				
	drums	bass	piano	guitar	others
test1					
baseline	63.7±7.2	58.5±7.4	55.7±8.6	-	-
proposed	65.3±7.2	67.0±7.2	74.3±8.1	69.5±6.8	59.5±7.7
test 2					
baseline	94.8±1.7	81.3±2.9	75.1±2.9	-	-
proposed	94.8±1.7	86.5±2.6	87.2±2.3	92.0±2.0	93.8±1.9

Table 6: Matching rate between the model's results and subjects' results focusing on overall and timbre, respectively. The baseline is our previous method [46].

Table 7: Number of answers using results with 80% agreement among subjects focusing on overall and timbre, respectively.

	Evaluation with answers on **overall**				
	drums	bass	piano	guitar	others
test 1	414	400	383	419	434
test 2	965	965	920	934	850
	Evaluation with answers on **timbre**				
	drums	bass	piano	guitar	others
test 1	193	188	140	203	173
test 2	879	879	913	892	834

5 Conclusions

In this paper, we proposed a method of computing similarities focusing on each instrumental sound using mixed sounds as input in one network, which extracts a single similarity embedding space with disentangled dimensions for each instrument using CSNs. To successfully train the network, we implemented new ideas for the training, such as the use of pseudo musical pieces, a norm loss, and pre-training. Experimental results showed the effectiveness of our strategies and that the selection of similar musical pieces focusing on each instrumental sound by the proposed method can obtain human acceptance, especially when focusing on timbre. Future work is to design a model that captures the structure of the time direction so that melody and rhythm can be considered.

Acknowledgment

References

[1] IFPI, *Global music report 2024*, `https://www.ifpi.org/wp-content/uploads/2024/04/GMR_2024_State_of_the_Industry.pdf`, 2024.

[2] Apple Inc, *Apple music*, `https://www.apple.com/jp/apple-music/`, 2024.

[3] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[4] Y. Song, S. Dixon, and M. Pearce, "A survey of music recommendation systems and future perspectives," in *International Symposium on Computer Music Modelling and Retrieval*, 2012, pp. 395–410.

[5] N. Koenigstein, G. Dror, and Y. Koren, "Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy," in *ACM Recommender Systems*, 2011, pp. 165–172.

[6] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer, "The yahoo! music dataset and kdd-cup'11," in *Proceedings of KDD Cup 2011*, ser. Proceedings of Machine Learning Research, vol. 18, 2012, pp. 3–18.

[7] B. Logan, "Music recommendation from song sets," in *International Society for Music Information Retrieval Conference*, 2004, 4 pages.

[8] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in neural information processing systems*, vol. 26, 2013, pp. 1–9.

[9] M. S. Fathollahi and F. Razzazi, "Music similarity measurement and recommendation system using convolutional neural networks," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 10, pp. 43–53, 2021.

[10] H.-C. Chen and A. Chen, "A music recommendation system based on music and user grouping," *Journal of Intelligent Information Systems*, vol. 24, pp. 113–132, 2005.

[11] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. Okuno, "Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences," in *International Conference on Music Information Retrieval*, 2006, pp. 296–301.

[12] B. McFee, L. Barrington, and G. Lanckriet, "Learning content similarity for music recommendation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2207–2218, 2012.

[13] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[14] T. Fujishima, "Realtime chord recognition of musical sound: A system using common lisp music," in *International Conference on Mathematics and Computing*, 1999.

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

[16] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *IEEE International Conference on Multimedia and Expo*, 2001, pp. 745–748.

[17] B. Whitman and R. Rifkin, "Musical query-by-description as a multiclass learning problem," in *IEEE Workshop on Multimedia Signal Processing*, 2002, pp. 153–156.

[18] E. Gomez, "Tonal description of polyphonic audio for music content processing," *INFORMS Journal on Computing*, vol. 18, pp. 294–304, 2006.

[19] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in *International Society for Music Information Retrieval Conference*, 2010, pp. 339–344.

[20] 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.

[21] L. A. Iliadis, S. P. Sotiroudis, K. Kokkinidis, P. Sarigiannidis, S. Nikolaidis, and S. K. Goudos, "Music deep learning: A survey on deep learning methods for music processing," in *International Conference on Modern Circuits and Systems Technologies*, 2022, pp. 1–4.

[22] K. Choi, G. Fazekas, K. Cho, and M. Sandler, *A tutorial on deep learning for music information retrieval*, arXiv:1709.04396, 2018.

[23] P. Li, J. Qian, and T. Wang, *Automatic instrument recognition in polyphonic music using convolutional neural networks*, arXiv:1511.05520, 2015.

[24] T. Hirvonen, "Classification of spatial audio location and content using convolutional neural networks," in *Audio Engineering Society Convention*, vol. 2, 2015, pp. 1–10.

[25] K. Choi, G. Fazekas, M. Sandler, and J. Kim, "Auralisation of deep convolutional neural networks: Listening to learned features," in *International Society for Music Information Retrieval Conference*, 2015, pp. 26–30.

[26] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," in *International Society for Music Information Retrieval Conference*, 2016, pp. 805–811.

[27] V. Lostanlen and C.-E. Cella, "Deep convolutional networks on the pitch spiral for music instrument recognition," in *International Society for Music Information Retrieval Conference*, 2016, pp. 612–618.

[28] Y. Costa, L. Soares de Oliveira, and C. Silla, "An evaluation of convolutional neural networks for music classification using spectrograms," *Applied Soft Computing*, vol. 52, pp. 1–37, 2017.

[29] C. Senac, T. Pellegrini, F. Mouret, and J. Pinquier, "Music feature maps with convolutional neural networks for music genre classification," in *International Workshop on Content-Based Multimedia Indexing*, 2017.

[30] J. Park, J. Lee, J. Park, J.-W. Ha, and J. Nam, "Representation learning of music using artist labels," in *International Society for Music Information Retrieval Conference*, 2018, pp. 717–724.

[31] J. Choi, J. Lee, J. Park, and J. Nam, "Zero-shot learning for audio-based music classification and tagging," in *International Society for Music Information Retrieval Conference*, 2019, pp. 67–74.

[32] L. Pretet, G. Richard, and G. Peeters, "Learning to rank music tracks using triplet loss," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 511–515.

[33] A. Wise, A. S. Maida, and A. Kumar, "Attention augmented cnns for musical instrument identification," in *European Signal Processing Conference*, 2021, pp. 376–380.

[34] J.-J. Aucouturier and F. Pachet, "Music similarity measures: What's the use?" In *International Society for Music Information Retrieval Conference*, vol. 7, 2002, pp. 339–340.

[35] E. Pampalk, "Computational models of music similarity and their application in music information retrieval," *PhD Thesis, Vienna University of Technology*, 40 pages, 2006.

[36] M. Casey and M. Slaney, "The importance of sequences in musical similarity," in *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5, 2006, 4 pages.

[37] J. Urbano, D. Martin, M. Marrero, and J. Morato, "Audio music similarity and retrieval: Evaluation power and stability," in *International Society for Music Information Retrieval Conference*, 2011, pp. 597–602.

[38] P. Hamel, M. E. P. Davies, K. Yoshii, and M. Goto, "Transfer learning in mir: Sharing learned latent representations for music audio classification and similarity," in *International Society for Music Information Retrieval Conference*, 2013.

[39] J. Schluter, "Learning binary codes for efficient large-scale music similarity search," in *International Society for Music Information Retrieval Conference*, 2013.

[40] W. T. Wolff Daniel, "Learning music similarity from relative user ratings," *Information Retrieval*, vol. 17, pp. 109–136, 2014.

[41] D. Wolff, A. Macfarlane, and T. Weyde, "Comparative music similarity modelling using transfer learning across user groups," in *International Society for Music Information Retrieval Conference*, 2015, pp. 24–30.

[42] R. Lu, K. Wu, Z. Duan, and C. Zhang, "Deep ranking: Triplet matchnet for music metric learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 121–125.

[43] J. Cleveland, D. Cheng, M. Zhou, T. Joachims, and D. Turnbull, *Content-based music similarity with triplet networks*, arXiv:2008.04938, 2020.

[44] D. Cheng, T. Joachims, and D. R. Turnbull, "Exploring acoustic similarity for novel music recommendation," in *International Society for Music Information Retrieval Conference*, 2020, pp. 583–589.

[45] G. Gabbolini and D. Bridge, "An interpretable music similarity measure based on path interestingness," in *International Society for Music Information Retrieval Conference*, 2021, pp. 213–219.

[46] Y. Hashizume, L. Li, and T. Toda, "Music similarity calculation of individual instrumental sounds using metric learning," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2022, pp. 33–38.

[47] Y. Bengio, "Deep learning of representations: Looking forward," in *International Conference on Statistical Language and Speech Processing*, 2013, pp. 1–37.

[48] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, *et al.*, "Hierarchical generative modeling for controllable speech synthesis," in *International Conference on Learning Representations*, 2019.

[49] W.-N. Hsu, Y. Zhang, R. J. Weiss, Y.-A. Chung, Y. Wang, Y. Wu, *et al.*, "Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 5901–5905.

[50] Y.-N. Hung, Y. Chen, and Y.-H. Yang, *Learning disentangled representations for timber and pitch in music audio*, arXiv:1811.03271, 2018.

[51] K. A. Y-J. Luo and D. Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders," in *International Society for Music Information Retrieval Conference*, 2019, pp. 746–753.

[52] K. Tanaka, R. Nishikimi, Y. Bando, K. Yoshii, and S. Morishima, "Pitch-timbre disentanglement of musical instrument sounds based on vae-based metric learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 111–115.

[53] A. Veit, S. Belongie, and T. Karaletsos, "Conditional similarity networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1781–1789.

[54] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Disentangled multidimensional metric learning for music similarity," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 6–10.

[55] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Similarity-Based Pattern Recognition*, 2015, pp. 84–92.

[56] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: A fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, p. 2154, 2020.

[57] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2019, pp. 45–49.

[58] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[59] 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 *International Society for Music Information Retrieval Conference*, 2017.

[60] *Crowdworks*, https://crowdworks.jp.

[61] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.