Research on Part-Level Music Similarity for Music Retrieval Focusing on Individual Instrumental Parts

Yuka Hashizume

Contents

Abstract

Music information retrieval (MIR) techniques, such as music retrieval and recommendation systems, enable listeners to access the vast collections of music available online. Music similarity is a fundamental cue in these systems. However, designing a music similarity measure remains challenging, since perceptual music similarity is complex and has no objective ground truth. One of the factors contributing to the complexity of music similarity perception is the multidimensionality of music. One potential direction is to decompose music similarity from the entire track level into the element level and flexibly reconstruct it. Although several representation learning and retrieval methods have been proposed to extract multiple features from music and combine them, all of them have considered entire music tracks, which are a mixture of multiple instrumental parts. However, popular music songs often contain a variety of instrumental sounds and singing voices. Considering this point, some instrumental parts within a music track, such as the drum part or the vocal part, may play a more significant role in listeners' perception, and the parts that are focused on can vary depending on the listener and the track. In this thesis, we focus on instrumental part-level similarity, which decomposes entire music track-level similarity into individual instrumental parts, enabling finer-grained modeling of music similarity that can flexibly adapt to differences across tracks and listeners.

We propose two methods based on deep metric learning: one trains separate networks

for each instrumental part with separated instrumental signals as input, and the other employs a unified network that extracts separated features from the original music signal. In the experiment of the first method, we show that using a source separation model to estimate the instrumental part signals leads to degraded accuracy due to errors in the source separation. The second method successfully extracts the feature representation corresponding to the timbre-related similarity perception.

Also, it is essential to understand the nature of part-level similarity perception and the relationship between perceptual and model-estimated similarity. To achieve this, we collect part-level similarity evaluations through crowdsourcing and analyze them. We show that the instrumental parts that are focused on when listening to the music track vary depending on the listener and the music track. Additionally, we find that rhythm plays a crucial role in similarity perception, although current estimation models do not adequately capture it.

Finally, we present the practical application of part-level similarity by implementing a music retrieval system interface that allows users to select and focus on specific instrumental parts. Through insights obtained from studies conducted from multiple perspectives, this thesis highlights the value of part-level similarity, the potential for effective estimation techniques, and their applicability in interactive music retrieval.

1 Introduction

1.1 General Background

Currently, streaming accounts for 69% of the global music market, with an estimated 752 million users subscribed to music streaming services worldwide [1]. Therefore, the convenience of streaming services has become an essential issue for many people. At the same time, subscription-based streaming services currently provide access to over 100 million tracks [2], representing an enormous volume of available music. It will take approximately 761 years to listen to all the available music, assuming the average music track duration is about four minutes. This highlights that it is practically impossible for listeners to find music they like by listening to every track. Given this situation, efficient methods that help listeners discover music they will like are essential.

Music information retrieval (MIR) techniques, particularly music retrieval and music recommendation systems, have been developed to address this challenge. Such techniques leverage computational power to efficiently help listeners access desired music tracks from vast music collections. Currently, the most common methods for accessing music are text-based retrieval and listening-history-based recommendation. However, text-based retrieval using metadata such as track title or artist name is too specific to discover unfamiliar music, while retrieval based on abstract attributes such as genre or mood requires costly manual annotation for each track [3]. Meanwhile, listening-history-based recommendation, such as collaborative filtering [4], suffers from the cold-

start problem, where recommendations for new listeners or new tracks are unreliable [5], and the long-tail problem, where less popular tracks are rarely recommended [6]. To address these limitations, content-based approaches [3] that predict and utilize features directly from the musical contents have been proposed as an effective alternative. Many content-based music retrieval systems rely on estimating the similarity between features extracted from musical content. In such similarity-based methods, the accuracy of similarity estimation is a key factor determining system performance. However, calculating music similarity in retrieval tasks is inherently challenging, as it must align with human perception and thus cannot be captured by a simple formula. Furthermore, unlike tasks such as cover song identification or version identification, there is no objective ground truth for music similarity when the goal is to enable systems to help users discover unknown music they will like.

One factor contributing to the complexity of human perception of music similarity is the multidimensionality of music [7,8]. Listeners are thought to rely on certain features as a cue when perceiving musical similarity [9], and many studies have attempted to identify these features. However, multiple features are known to be involved, and they interact with each other [10, 11]. In addition, there are individual differences in music perception [7, 8], and it has been pointed out that perception may also vary depending on changes in mood, situation, and circumstances [8]. To address these challenges, one potential direction is to decompose music similarity from the entire track level into the element level, allowing finer-grained modeling of music similarity that can flexibly adapt to differences across tracks and listeners. Previous studies have investigated perceptual similarity from multiple perspectives [12]. In addition, multidimensional retrieval systems [13, 14] and methods for decomposing musical signals into pitch- and timbre-related features [15, 16] or higher-level attributes [17] have been developed.

In popular music, a wide variety of genres exist, and genre serves as an important descriptor. Instrumentation, which is closely related to genre, has also been used as an important descriptor [17–19]. However, since certain instrumental parts within a music track may contribute more significantly to perceived similarity than others, it is important to account for differences in their relative importance. For example, drum sounds contribute to groove [20, 21] and genre characteristics [22], and guitar timbre defines stylistic identity in genres like metal [23]. In the context of perceptual similarity, it is known that the main instrument influences perceived similarity [10]. These observations suggest that music retrieval and similarity modeling that focus on individual instrumental parts have the potential to provide more flexible and fine-grained access to musical content. While some existing approaches focus on vocals [24, 25], retrieval frameworks that explicitly focus on non-vocal instrumental parts, or that jointly account for multiple instrumental parts within a track, remain limited.

1.2 Thesis Scope

In this thesis, we aim to estimate and analyse perceptual music similarity focusing on each instrumental part (*part-level* similarity) rather than the similarity of entire music tracks (*track-level* similarity). Throughout this thesis, the term *instrumental part* refers to sounds performed by individual instruments, such as drums or bass, as well as aggregated sounds of instruments belonging to the same category. Vocal sounds are also treated as an instrumental part. A *music track* is defined as a complete musical piece in which multiple instrumental parts are mixed together. Here, part-level similarity refers to similarity assessed by separately focusing on individual instrumental parts, whereas track-level similarity refers to similarity assessed by considering all sounds contained in a music track as a whole. Decomposing music similarity into

part-level similarities enables the flexible reconstruction of music similarity, allowing for adaptation to individual differences as well as variations in the salient elements across music tracks. For example, it allows focusing on drums in one music track while focusing on guitars in another, or enables one listener to attend to the vocals while another listener attends to the piano in the same track.

The scope of this thesis is organized into three main parts:

1. **Model development:** We develop a deep learning model for automatically estimating part-level similarity.
2. **Perceptual analysis:** We analyze part-level similarity from a perceptual perspective through a listening test. Additionally, we compare the model's estimations with perceptual similarity to evaluate its effectiveness and gain insights for future model development.
3. **Application:** We implement a music retrieval system interface based on part-level similarity, demonstrating the applicability of the part-level similarity in interactive music retrieval.

Through these three stages, this thesis aims to bridge computational modeling and human perception of musical similarity, contributing to the advancement of perceptually grounded and flexible music retrieval.

1.3 Thesis Overview

The overview of this thesis is illustrated in Figure 1.1.

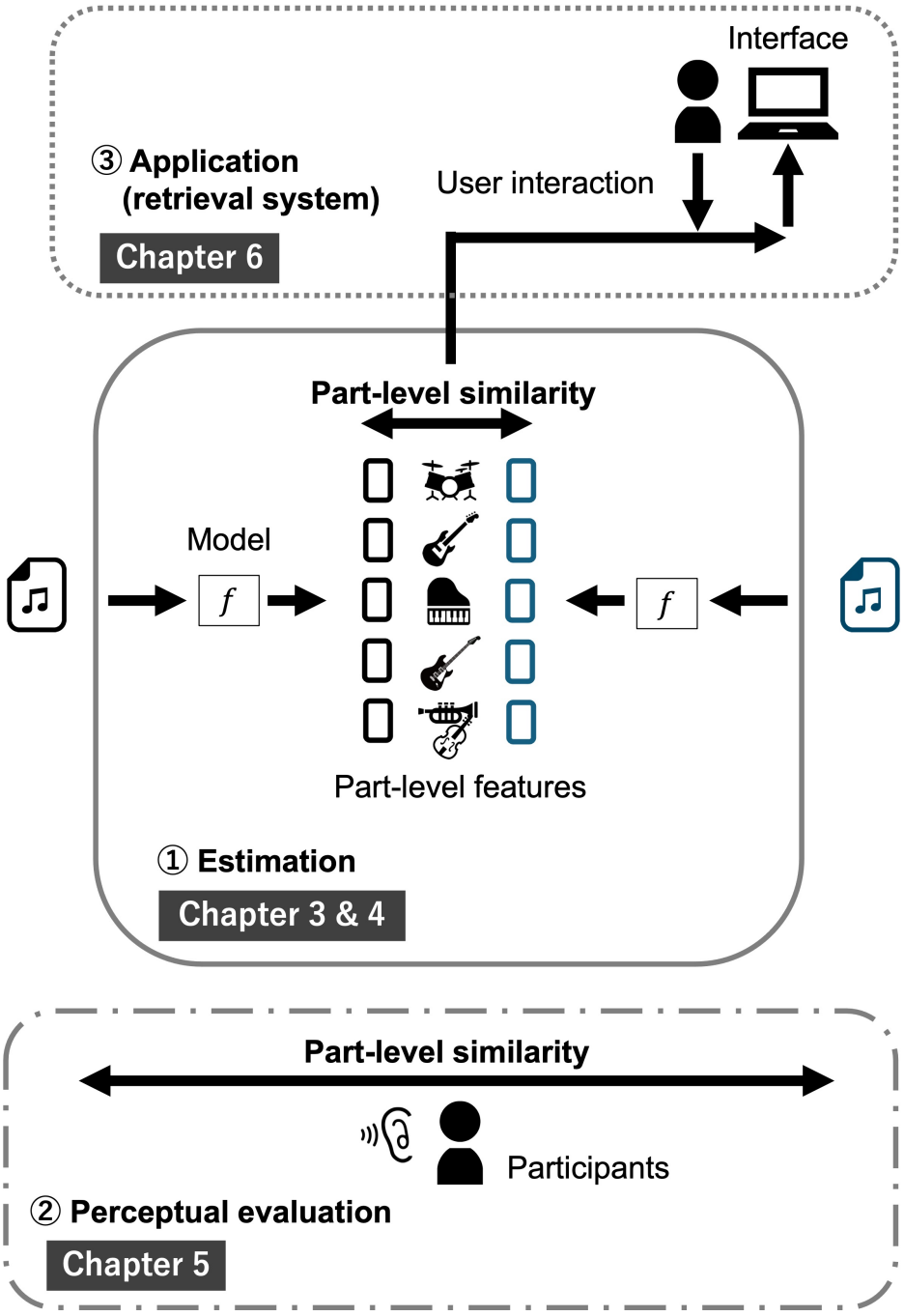


Figure 1.1: *The overview of this thesis*

1.3.1 Part-Level Similarity Estimation Model Development

While previous works have focused on estimating track-level music similarity, we propose methods for estimating part-level similarity. We propose two methods based on deep learning in Chapter 3 and 4. Estimating part-level similarity requires a mechanism that focuses on specific instrumental parts within a music track.

The first proposed method described in Chapter 3 takes an approach where the audio signal is first separated into individual instrumental parts before being converted into feature representations. In other words, audio signals representing individual instrumental parts are required as inputs to a deep learning model. In popular music production, a recording is typically created through a mixing process. During mixing, recorded sounds of instruments and vocals are handled either individually or in groups of similar categories, and are treated as the smallest units in downstream mixing operations. These units are commonly referred to as stems. The final music track is produced by adjusting and combining these stems, and the resulting mixture is publicly released in audio form. The corresponding stems are generally not made publicly available. As a result, when individual instrumental part signals are required, they must be extracted from the mixed music signal at the signal level. This problem has been extensively studied in the field of music source separation. In the music source separation task, signals corresponding to individual instrumental parts are estimated from a mixture signal. Under an ideal separation scenario, the estimated signals are equivalent to the original stems. By applying music source separation techniques, it is therefore possible to obtain estimated instrumental part signals, which can be used as inputs to the proposed model. However, in practice, current music source separation methods are not perfect, and the resulting separation errors can affect the accuracy of similarity estimation.

Second, to address this issue, we propose a method that directly extracts individual part-level feature representations from a music track in Chapter 4. We train a unified feature extraction network using the original music as input. This network aims to capture the distinct similarities of individual instrumental parts from a single input. This is achieved by feature-level separation, extending Conditional Similarity Networks [26] with our novel data augmentation approach. Specifically, the model is trained to learn a representation in which a feature extracted from a music track signal is organized into subspaces, each corresponding to the characteristics of a distinct instrumental part. An overview of these proposed methods is illustrated in Figure 1.2.

1.3.2 Part-Level Perceptual Similarity Analysis

In Chapter 5, we investigate part-level similarity from a perceptual perspective. Part-level similarity must align with human perception for applications such as music retrieval and recommendation. Developing models that estimate similarity based on perceptual criteria requires both an analysis of human perception and an evaluation of model performance using perceptual ratings. Accordingly, we collect perceptual evaluations of part-level similarity via crowdsourcing. We reveal the following points: (1) the value of calculating part-level similarity; (2) the relationship between the part-level and track-level similarity; (3) the perspectives that listeners give priority to in part-level similarity; (4) the correspondence between the perceptual and estimated part-level similarity; (5) validation of the learning method for part-level similarity estimation that we proposed.

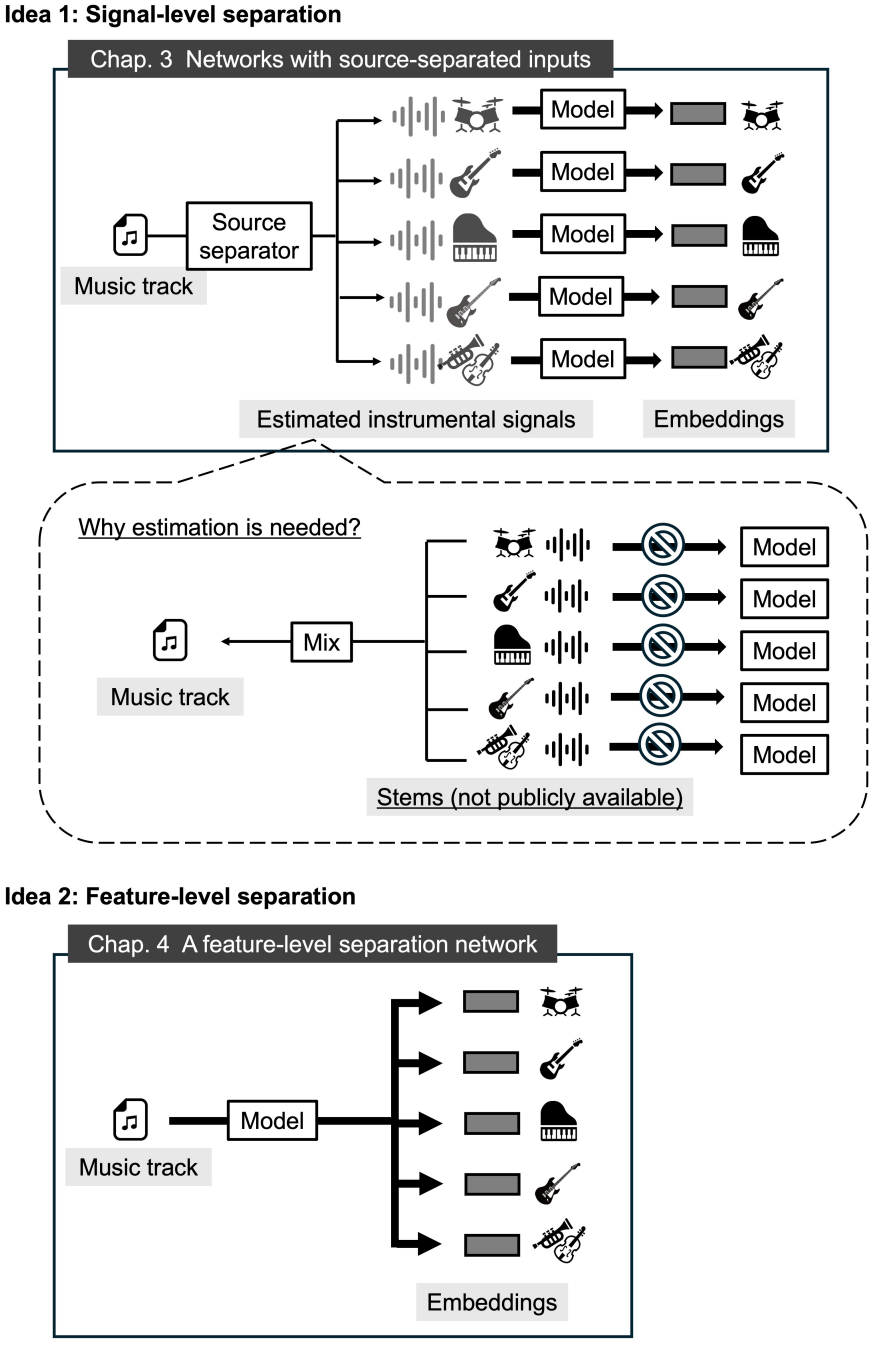


Figure 1.2: *Two ideas for the part-level similarity estimation. (Idea 1) First, we consider the idea of separating the input at signal levels to extract the distinctive features of each instrumental part. In Chapter 3, estimated instrumental part signals using the source separation model are used as input instead of stems, and individual feature extraction models are trained for each. (Idea 2) Second, we consider the idea of separating at the feature level without separating the inputs. In Chapter 4, we propose a unified network for extracting part-level feature representations directly from a music track.*

1.3.3 Application: Part-Level Music Retrieval Interface

As an application of part-level similarity, we discuss a music retrieval system in which users are free to select the instrumental parts they wish to focus on in Chapter 6. Accordingly, we implemented a retrieval interface that enables such user interaction. Users can select their favorite instrumental parts from any song and aggregate them as a query for the retrieval. In other words, users can retrieve tracks that simultaneously resemble, for example, the drums of track A and the vocals of track B.

2 Background and Related Work

In this chapter, we introduce previous works related to this thesis, covering three main aspects: feature extraction from audio signals (related to Chapter 3 and 4), perceptual analysis (related to Chapter 5), and music retrieval systems (related to Chapter 6). Section 2.1 presents conventional MIR techniques, particularly methods for extracting musical features. Section 2.2 reviews research on perceptual music similarity, highlighting its multidimensional nature and the construction of ground truth data. Section 2.3 discusses existing music retrieval systems. We summarize and illustrate the position of our study in relation to the related works discussed in Sections 2.1 and 2.3 in Figure 2.1.

2.1 Content-Based Music Information Retrieval (MIR) Techniques

In Chapters 3 and 4, we propose methods for estimating part-level similarity. In this section, we review works related to this topic. Our approach estimates music similarity based on the similarities between representations trained with data-driven deep learning, extracted from audio features converted from music audio signals. The review focuses on works that are most relevant to this framework. Note that audio signals refer to raw acoustic waveforms in the time domain, whereas audio features denote

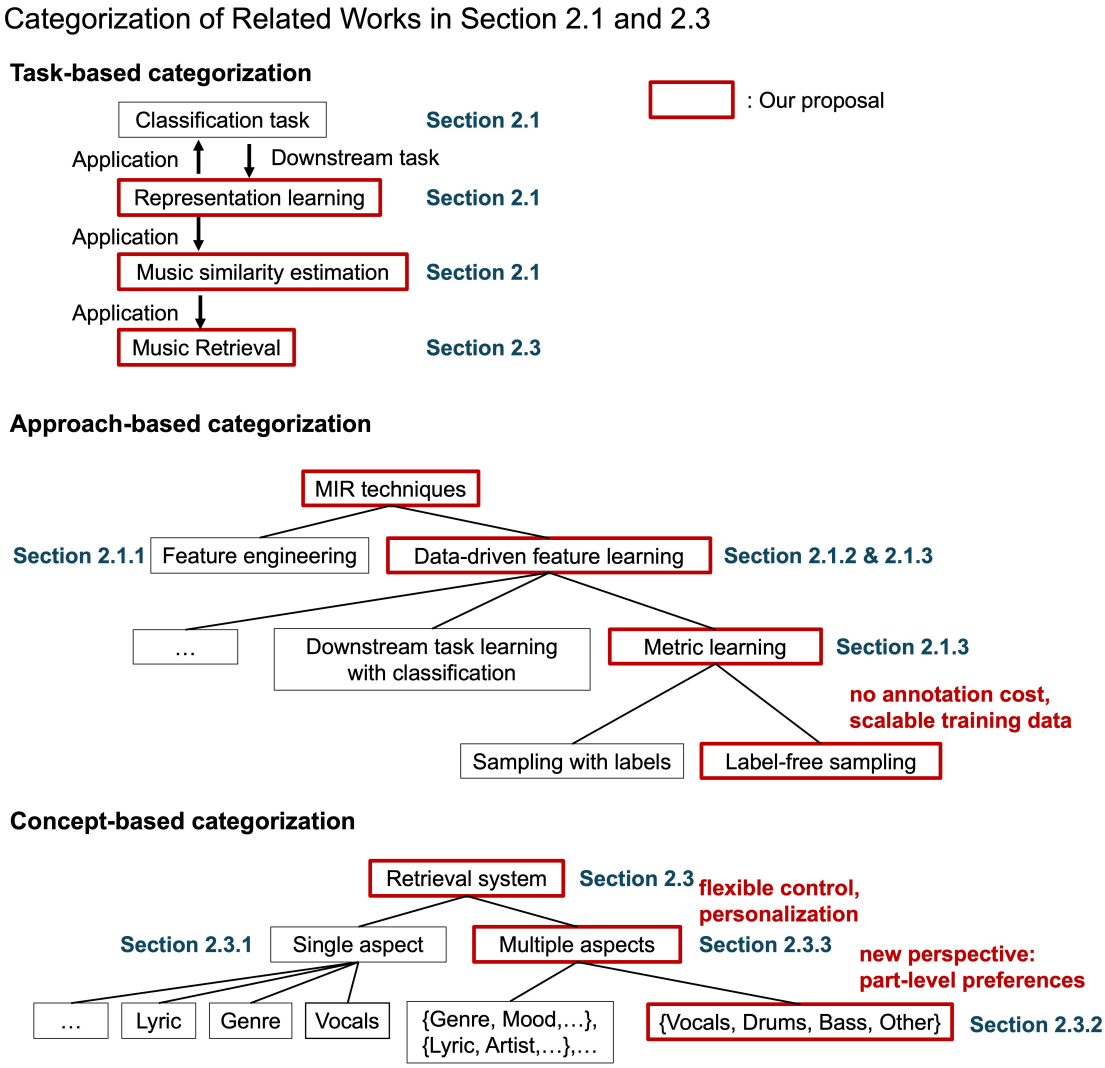


Figure 2.1: *The position of our study (red squares) in relation to the related works discussed in Sections 2.1 and 2.3 is illustrated from three aspects: task-based categorization, approach-based categorization, and concept-based categorization.*

time-frequency features or descriptors obtained by simply applying signal processing transformations to these waveforms. Section 2.1.1 provides an overview of content-based approaches, particularly those that use audio signals. Section 2.1.2 reviews data-driven deep learning methods among them, and Section 2.1.3 and 2.1.4 further explore this line of works in detail.

2.1.1 Overview of Content-Based MIR

MIR refers to a research field that focuses on developing technologies to assist listeners in discovering and accessing information about music, as well as information contained within music [3, 8, 27, 28]. In addition to the typical tasks such as music recommendation and music retrieval, the field encompasses a wide range of specific tasks, including similarity estimation, beat tracking, auto-tagging, music captioning, genre classification, and mood or emotion recognition, among others. Although these tasks differ in their specific objectives, they serve as fundamental techniques that support retrieval, recommendation, and other navigation systems for music content.

As discussed in Chapter 1, content-based MIR techniques have the potential to overcome the limitations inherent in text-based retrieval and listening-history-based recommendation. Content-based MIR has been studied since the 1990s, mainly using signal processing methods and handcrafted features capturing musical attributes. These approaches aimed to extract task-relevant and perceptually meaningful information from raw audio signals, which are typically high-dimensional and contain substantial redundant or irrelevant information, such as silence or repeated patterns. By designing features that emphasize important musical characteristics, researchers aimed to enhance task performance while minimizing computational complexity, thereby making the data more tractable for clustering, pattern matching, and traditional machine learning al-

gorithms [3,29]. During this period, researchers mainly focused on theoretically or empirically designing processing methods to extract musically meaningful features, such as the choice of audio features, their combinations, and subsequent processing steps. For example, Fujishima [30] proposed musical chords recognition method by applying the discrete fourier transform followed by pattern matching; Logan and Salomon [31] extracted mel frequency cepstral coefficients (MFCCs), clustered them to construct feature representations, and computed the earth mover's distance between features to estimate music similarity; Tzanetakis and Cook [32] used handcrafted features based on the short-time fourier transform (STFT) and wavelet transform to represent timbre, instrumentation, and rhythm, and trained classifiers for genre classification; Whitman and Rifkin [33] reduced the dimensionality of power spectral density features via principal component analysis and performed multi-class classification using regularized least squares to achieve query-by-description retrieval; and Gómez [34] computed harmonic pitch class profile vectors from spectral peaks to represent tonal features and calculate tonal similarity. Methods have also been proposed that statistically represent MFCCs using Gaussian mixture models and predict music similarity by computing likelihoods or Kullback–Leibler divergences between the models [35,36].

Around 2010, the field began transitioning from handcrafted feature engineering to learning feature representations directly from data. As described above, the former relies on researchers manually designing transformation functions and selecting meaningful features, for example by proposing timbral or rhythmic descriptors and further combining them into feature vectors that are fed into classifiers. In contrast, the latter directly takes relatively high-dimensional audio features, such as spectrograms, as model inputs and learns from data to extract semantically meaningful representations while progressively reducing dimensionality. Dieleman and Schrauwen [37] proposed

a feature representation learning method using the spherical K-means algorithm with mel spectrograms converted from audio segments as input. Hamel *et al.* [38] proposed a linear embedding method to obtain latent representations trained on genre classification and tagging tasks, and subsequently evaluated these learned representations on music similarity tasks, thereby demonstrating the potential of transfer learning in MIR. Schlüter [39] learned music similarity representations using auto-encoders and metric learning frameworks. McFee *et al.* [40] proposed a metric learning approach that incorporates collaborative filter data for sampling, enabling music recommendation based on learned distance metrics. Furthermore, Wolff and Weyde [41] learned perceptual music similarity spaces using metric learning from relative similarity annotations.

Since the mid-2010s, the emergence of deep learning as a powerful approach in computer vision and natural language processing has profoundly influenced the audio, speech, and MIR community [42]. Data-driven deep neural network approaches have increasingly replaced traditional machine learning pipelines, achieving state-of-the-art performance across MIR tasks. For instance, Elbir and Aydin [43] demonstrated that deep learning–based methods significantly outperformed conventional approaches that rely on handcrafted features and classical classifiers in genre classification and music recommendation tasks. In line with this technological progression, this thesis focuses on data-driven approaches based on deep learning approaches, which are discussed in detail in the following section.

2.1.2 Overview of Deep Learning Frameworks

A wide range of deep learning approaches have been proposed for various MIR tasks after the mid-2010s [42, 44]. Although the final outputs of models may take different forms depending on the task, latent feature representations play a central role in most

deep learning–based MIR frameworks, as they are used to compute both the loss functions and the final outputs. Therefore, the design and learning of these representations are crucial to model performance, regardless of the task type. Along with this trend, the representation learning task, which aims to acquire general-purpose feature representations, has also become an important research direction within MIR. In practice, such latent representations are typically obtained by training neural networks to map high-dimensional audio features, such as spectrograms, into lower-dimensional embedding spaces optimized for a given objective. This can be achieved, for example, by training models to solve specific downstream tasks or by learning representations in an unsupervised or self-supervised manner.

Although the specific criteria for optimal feature representations differ across tasks, all approaches share the common goal of extracting musically meaningful features. Consequently, even though MIR tasks vary widely in objective and form, many data-driven methods are conceptually related to similarity estimation. In fact, previous works have shown that feature representations learned from genre classification tasks can be effectively transferred to music similarity prediction [38]. This observation explains why training models on specific tasks is commonly adopted as an effective approach for representation learning.

We can review previous works on deep learning frameworks for MIR from two perspectives. The first perspective concerns learning strategies, which include the design of loss functions and data utilization methods. These strategies are typically highly dependent on the desired output, such as whether the task involves classification or regression. The second perspective concerns the neural network architectures. Although the usefulness of a particular architecture may vary depending on the task, in data-driven learning, the dependence on task characteristics tends to be stronger for

learning strategies than for network structures. Therefore, knowledge about effective network architectures can often be generalized and shared across a wide range of MIR tasks.

2.1.3 Learning Strategies

Overview

This section discusses the design of loss functions and data utilization strategies in deep learning frameworks for content-based MIR. While these approaches depend on the specific objectives of each task, some of them also provide general-purpose methods, such as transfer learning and representation learning, that can be applied across different tasks.

In classification tasks, datasets annotated with class labels are typically used, and models are trained using loss functions such as cross-entropy loss. Representative tasks include genre classification [43, 45–48], mood and emotion classification [47, 49], instrumentation recognition [46,47,50], and era classification [46,47]. Music tagging [51] can also be regarded as a multi-label classification task. Also, classification serves as a downstream task to learn feature representations [52]. The feature representations are intended to be used later for other MIR applications, such as recommendation, retrieval, or similarity estimation. For example, feature representations learned from genre classification have been successfully applied to music recommendation [43] and music similarity estimation [53].

In recent years, metric learning has become a popular alternative to classification-based training for representation learning. Metric learning enables models to learn meaningful similarity relationships directly, rather than being limited to discrete class

boundaries as in classification. Metric learning with triplet loss [54], a popular approach in representation learning for MIR, requires data sampling: anchor and positive samples, which are similar to the anchor, and negative samples, which are dissimilar to the anchor (details will be explained later). Data sampling often relies on labeled datasets to construct meaningful triplets. Some works have used similarity-annotated datasets [55], while others have employed traditional classification labels or tags, such as genre [17, 56], mood [17, 56], instrumentation [17], or artist [57], as surrogate similarity criteria. In addition, label-free approaches have been proposed, in which a track is divided into temporal segments, and segments from the same track are encouraged to be close in the latent space [17]. This strategy enables unsupervised representation learning based solely on the structure of the audio data itself. Another line of research focuses on crossmodal approaches, where audio signals are learned jointly with other modalities such as natural language. In these frameworks, models learn to map acoustic and textual representations into a shared latent space, allowing for semantically meaningful retrieval across modalities. Metric learning with contrastive loss is often used in learning cross-modal embeddings [58].

More recently, self-supervised learning has demonstrated strong performance in MIR tasks [59]. This approach aims to learn general-purpose latent representations without the need for explicit labels. Models are pretrained on large-scale unlabeled datasets through self-supervised objectives, and the resulting representations can be fine-tuned for downstream tasks such as classification, recommendation, or similarity estimation. These pretrained models provide a flexible and efficient foundation for a wide range of MIR applications.

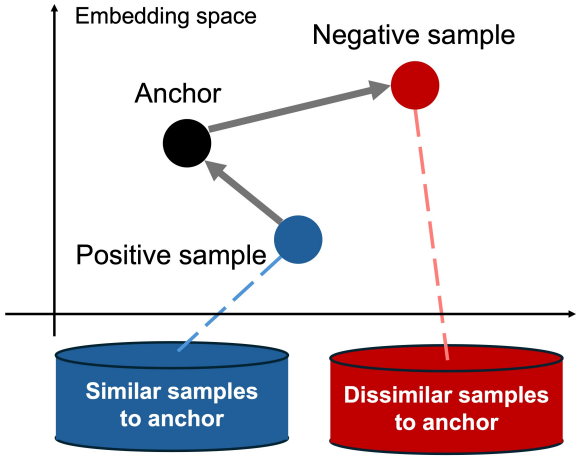


Figure 2.2: *Overview of metric learning with triplet loss*

Formulation of Metric Learning with Triplet Loss

Here, we describe the formulation of metric learning using triplet loss, which has been widely adopted in recent representation learning works and is also employed in our work. As mentioned above, metric learning with triplet loss uses triplets consisting of an anchor, a positive sample, and a negative sample as inputs to the loss function. The positive sample is sampled from data similar to the anchor, whereas the negative sample is sampled from data that are not similar to the anchor. Training encourages the distance from the anchor to the positive sample to be less than that to the negative sample as illustrated in Figure 2.2.

If we use $x_i^{(a)}$, $x_i^{(p)}$, and $x_i^{(n)}$ to denote the ith anchor, positive sample, and negative sample, respectively, the triplet t_i is constructed 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

$$\mathcal{L}(t_i) = \max\{d(x_i^{(a)}, x_i^{(p)}) - d(x_i^{(a)}, x_i^{(n)}) + \Delta, 0\}, \tag{2.1}$$

where d is a distance function for measuring the distance between two audio samples, such as the Euclidean distance or cosine distance, and Δ is a margin value, which

defines the minimum distance between the positive and negative samples.

2.1.4 Network Architectures

As mentioned earlier, network architectures used in MIR often share common structures across different tasks. While the specific design may vary depending on the goal, such as genre classification or beat tracking, they generally require a functionality that captures musically salient elements from both temporal and spectral information.

Before deep learning emerged as a dominant approach in computer vision and natural language processing, several works had already explored the use of deep neural networks for feature extraction in MIR. One early example is the Deep Belief Network (DBN), a neural network composed of multiple layers of restricted Boltzmann machines. Hamel and Eck [60] demonstrated that using a DBN trained on MFCCs features for genre classification, and then feeding the learned feature representations into a support-vector machine (SVM), outperformed using raw MFCCs directly as SVM input. Similarly, convolutional neural networks (CNNs) were also introduced around 2010 for MIR tasks [61]. In this work, MFCCs were used as input, and the learned representations with the CNNs were utilized for SVM-based classification.

From the mid-2010s, following the success of deep learning in vision and language, DNN-based methods became increasingly prevalent in MIR [42]. Early approaches employed multilayer perceptrons for representation learning in MIR [45, 52]. As CNNs proved highly effective in image-related domains, they were soon adopted in a wide range of MIR tasks such as music recommendation [62], instrument recognition [50, 63, 64], genre classification [46, 48, 65, 66], and music tagging [51, 66, 67]. CNNs have the advantage of treating time–frequency representations as images, allowing them to capture both temporal and spectral information without losing the sequential structure

in either dimension. Specifically, examples of time–frequency representations include STFT spectrograms [46, 64, 65], mel spectrograms [51, 62, 66, 67], and CQT spectrograms [63, 64]. When dealing with temporal dependencies, recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) are often preferred, as they can model sequential dynamics. These architectures are particularly effective for tasks sensitive to temporal variations, such as dynamic emotion prediction [68]. To leverage both strengths, hybrid architectures combining CNNs and RNNs (or LSTMs) have also been proposed [47,49]. More recently, Transformer-based architectures have gained popularity in MIR research. Their ability to capture long-range dependencies and contextual relationships has made them a powerful alternative to traditional recurrent and convolutional models, marking the latest trend in deep learning approaches for music understanding [59].

2.2 Perceptual Music Similarity

In Chapter 5, we conduct a large-scale listening test to collect evaluations on perceptual music similarity at the instrumental part level. The objectives of this work are twofold. First, to investigate the perceptual tendencies that are observed when listening to music with attention to individual instrumental parts. Second, to construct ground-truth data for performance evaluation of deep learning models that aim to estimate part-level perceptual similarity. In this section, we review previous works closely related to our study, which aims at these purposes. Section 2.2.1 reviews works investigating the multidimensionality of music similarity perception. Section 2.2.2 reviews works that constructed ground-truth data from subjective evaluations. Finally, Section 2.2.3 reviews works on perceptual similarity between instrumental sounds.

2.2.1 Multidimensionality of Similarity Perception

To understand perceptual similarity in music, it is essential to clarify what makes two music tracks sound similar to listeners. Studies applying Gestalt theory to music perception [9] emphasize the role of *cues*, defined as "a salient element that is prominent at the musical surface." Repetition and transformation of such cues lead listeners to perceive similarity. However, music perception is inherently multidimensional, involving multiple *cues* that interact in complex and intertwined ways [7, 8].

For example, McAdams *et al.* [11] demonstrated through perceptual experiments that both pitch and rhythm play crucial roles in melodic similarity judgments. Listeners heard a reference melody followed by a transformed version and rated the perceived similarity on a scale from 1 to 9. Altering either pitch or rhythm reduces perceived similarity, with changes in rhythm having a greater impact than changes in pitch. Eerola *et al.* [69] showed that perceived melodic similarity can be explained to some extent by multiple frequency-based features, and that the explanatory power improves when additional features such as rhythm are incorporated. Participants were presented with pairs of folk music melodies and asked to rate their perceived similarity on a 9-point scale. The obtained similarity ratings were analyzed using multidimensional scaling and regression analyses. The results showed that frequency-based statistical properties, such as the distribution of tones, can explain about 40% of similarity judgments for folk melodies, and that incorporating descriptive features such as the number of tones, rhythmic variability, and melodic predictability improves the prediction rate to approximately 55%. Lamont and Dibben [70] reported that similarity judgments rely mainly on surface features, such as dynamics, articulation, texture, and contour, based on pairwise comparisons of piano pieces. Cupchik *et al.* [10] demonstrated that listeners' judgments of music similarity are influenced by both genre and musical at-

tributes such as tempo, dominant instrument, and articulation. In their perceptual experiment, participants rated the similarity of pairs of musical pieces, and the resulting judgments were analyzed using a multidimensional scaling paradigm. In the jazz-only condition, that is, within a single genre, the primary perceptual dimensions were tempo, dominant instrument, and articulation. In contrast, in experiments including multiple genres, the most salient distinction was between classical and modern music, followed by differences between pop-rock and jazz, highlighting the importance of genre-based categorization in perceived musical similarity. Novello *et al.* [71] also demonstrated that genre plays a central role in music similarity perception. In their perceptual experiment, listeners evaluated short music clips using a triadic comparison paradigm, in which they selected the most similar and the most dissimilar pair within each triad. The resulting similarity judgments were analyzed using multidimensional scaling, revealing clear clustering of songs belonging to the same genre. Their further work has identified genre, tempo, and timbre as statistically significant determinants of perceived similarity of western popular music, with genre exerting the strongest influence, followed by tempo and timbre [72].

These studies collectively demonstrate the inherently multidimensional nature of music perception. At the same time, they suggest that the dominant perceptual dimensions in similarity judgments can vary depending on the type of stimuli used, such as isolated melodies, piano pieces, or popular music excerpts, as well as on experimental designs that focus on specific musical elements, for example, pitch or rhythm. In this context, examining music similarity through evaluations of individual instrumental parts is expected to yield additional insights into music similarity perception.

2.2.2 Constructing Ground Truth from Subjective Evaluation

While human listeners tend to show agreement in their similarity judgments [71], individual differences inevitably exist. Listeners' perceptions vary due to experiential or physical differences. Moreover, as discussed in Section 2.2.1, musical similarity is inherently multidimensional, leading to variability across evaluators. Such variability provides a significant challenge when constructing ground-truth data from human assessments [7]. To address this, several works have proposed large-scale collection methods and aggregation strategies for human similarity data. Ellis *et al.* [7] collected 138,000 triplet comparisons for musical artist similarity, each consisting of a source artist, a relatively similar target, and a relatively dissimilar sample. They evaluated computational models by ranking the target's position relative to the source, providing a benchmark for algorithmic prediction of perceptual similarity. Building upon this, Berenzweig *et al.* [73] integrated these triplet data with expert opinions, playlist co-occurrence data, and web text information, constructing a similarity matrix to evaluate with more data points. Other works have focused on expert-based evaluations. Typke *et al.* [74] created ground-truth data for musical incipits by asking 35 human experts to rank approximately 50 candidates per query across 11 queries. Müllensiefen and Frieler [75] conducted perceptual similarity experiments with 23, 12, and 5 musical experts across three experiments, demonstrating both inter-evaluator consistency and the feasibility of algorithmic prediction of expert judgments. Volk *et al.* [12] collected annotations for 360 folk-song melodies from three experts, evaluating four musical aspects: rhythm, contour, motifs, and lyrics, as multiple perceptual dimensions.

These works reveal a general trend: expert evaluations can be obtained from smaller groups with high consistency, while non-expert evaluations typically require large participant pools to achieve statistical robustness. Furthermore, in connection with Sec-

tion 2.2.1, to mitigate the challenges arising from perceptual multidimensionality, some works, such as Volk *et al.* [12], explicitly collected dimension-specific similarity ratings, thereby decomposing overall similarity into distinct perceptual factors.

2.2.3 Similarity of Instrumental Sounds

Previous listening experiments focusing on instrumental sound similarity have primarily aimed to analyze timbre perception. Typically, such works used recordings of different instruments playing the same melody as stimuli to examine how changes in timbre influence perceptual judgments of similarity. Eerola *et al.* [76] explored the role of timbre in the perception of affective dimensions. Their findings suggest that instrumental sounds contain acoustic cues that represent affective expression, which are consistently recognized by listeners. Similarly, McAdams *et al.* [77] investigated perceptual timbre spaces by collecting similarity ratings among 18 synthesized instrument sounds, showing that log-rise time, spectral centroid, and spectral flux were strongly related to perceived timbral similarity. Lakatos [78] further compared evaluations by expert musicians and nonmusicians for pitched and percussive instruments, finding that spectral centroid and rise time form the principal perceptual dimensions of timbre regardless of musical training.

Overall, these works focused primarily on inter-instrument timbral comparisons, examining differences between distinct instrument categories. They did not typically address within-category variations, such as different playing techniques on the same instrument, nor did they address inter-part relationships within the same music track. Our study aims to extend this line of research by exploring perceptual music similarity at the instrument-part level.

2.3 Music Retrieval System Concepts

In Chapter 6, we implement an interactive music retrieval system interface that allows users to flexibly aggregate part-level similarities as an application of part-level similarity. This section reviews related works. Whereas Section 2.1 provided a technical overview of content-based MIR methods, this section focuses on conceptual aspects, specifically narrowing the scope to music retrieval systems. Section 2.3.1 presents an overview of retrieval systems, and Sections 2.3.2 and 2.3.3 review two research areas closely related to the proposed interface in Chapter 6: retrieval systems based on individual instrumental parts and retrieval systems based on multiple aspects.

2.3.1 Overview

Research on music retrieval began to gain attention in the 1990s [28]. Early examples include query-by-humming systems [79] and retrieval based on acoustic features such as loudness, pitch, and timbre [80]. Since the 2000s, a wide variety of music exploration interfaces have been proposed. Among them, map-based retrieval interfaces have attracted significant interest. Various visualization strategies have been explored in this line of work, for example, representing music collections as islands [81], 3D landscapes [82, 83], spherical mappings [84], album cover displays [85], and galaxy metaphors [14, 86]. In addition to visualization strategies, some systems differ in terms of the perspective from which music is explored. These include interfaces focusing on genre [87], artist similarity [88], mood or emotion [89, 90], and instrumentation [19]. Other systems combine multiple perspectives, for instance, interfaces that allow switching between timbre-, rhythm-, and metadata-based similarities (e.g., artist or genre) [13], or systems that allow focusing on multiple aspects from timbre, rhythm,

dynamics, and lyrics [14]. Public web services have also been developed to enable interactive visualization and exploration of music on video sharing platforms [86]. While the aforementioned examples focus on track-level representation, Artist Map [91] proposes an artist-level mapping that incorporates additional perspectives such as genre, tempo, and era. Other exploration-oriented interfaces include systems for discovering unknown artists [36] and lyric-based retrieval systems [92, 93].

Although tasks such as cover song detection [94] have long been important topics in music retrieval, they differ in motivation from the systems introduced here, which aim to support the discovery of new, personally appealing music. More recently, in music retrieval tasks, cross-modal retrieval has become an active area of research. While some approaches targeting other modalities, such as video retrieval using music queries, are less relevant, text-to-music retrieval systems [95] represent innovative methods that are closely aligned with our research motivation.

2.3.2 Retrieval Systems based on Instrumental Parts

Music retrieval systems that focus on vocals[1] have been proposed. Fujihara *et al.* [24] proposed a retrieval system based on vocal timbre similarity. Mao *et al.* [96] proposed a music retrieval method designed for singers rather than listeners, based on the vocal range profile as a measure of singing skill. Nakano *et al.* [25] proposed a music retrieval interface focusing on vocal timbre and pitch.

Additionally, methods based on other instrumental parts have been proposed, such as those using volume adjustment for each instrumental part [97]. However, this work did not consider queries that combine multiple instrumental parts from different songs, nor implement an interface. Although an instrumentation-based retrieval system has been

[1] We treat vocals as one of the instrumental parts for convenience.

proposed [19], this is a retrieval system based on features that indicate how long each type of instrument is performed and does not take into account the acoustic part-level similarity. To the best of our knowledge, no previous work has implemented a retrieval system that flexibly takes into account part-level preferences of multiple instrumental parts.

2.3.3 Retrieval Systems based on Multiple Aspects

Some MIR technologies that can focus on multiple aspects have been proposed. For example, a playlist generation method using a hypergraph with edges representing different musical aspects was proposed [98], such as user listening patterns, era, and lyrics. Watanabe *et al.* [99] proposed a music exploration system that finds music by combining different aspects: lyric word, audio, and artist. Lee *et al.* [17] proposed a disentangled feature representation whose subspaces represent different aspects of music: genre, mood, instrumentation, and tempo. Bostandjiev *et al.* [100] proposed a music recommender system that adjusts the weights of preferred artists and the contextual weights based on Wikipedia and social media. Millecamp *et al.* [101] conducted a user study in which musical attributes (acousticness, energy, valence, danceability, and instrumentalness) were adjusted using a radar chart and sliders to manipulate Spotify recommendations. While these works consider multiple aspects of music, they do not focus on instrumental parts or provide an interactive mechanism for users to specify and combine instrumental parts when constructing queries.

2.4 Summary

In this chapter, we reviewed previous works related to the estimation, perceptual analysis, and application of part-level similarity, which are the main focuses of this thesis. Section 2.1 presented methods for extracting feature representations from audio signals, mainly using deep learning approaches, following previous works in MIR. The techniques introduced here extract features from musical signals containing mixtures of various instruments, i.e., track-level features, rather than part-level. Section 2.2 reviewed works on perceptual music similarity, summarizing findings on the multidimensional nature of perceptual music similarity, as well as efforts to construct ground truth data from evaluations by multiple listeners. These works did not address the analysis of music similarity with a focus on individual instrumental parts. Section 2.3 reviewed the concepts of existing music retrieval systems. To our knowledge, no previous work has realized a retrieval system that computes part-level similarities and interactively integrates them for music retrieval.

3 Part-Level Similarity Estimation with Source-Separated Inputs

In this chapter, we describe one of our methods for estimating part-level similarity. We propose a method that trains networks for individual instrumental parts, with the separated instrumental part signals as input. An overview of this approach is shown in the upper part of Figure 1.2. Section 3.1 provides an introduction, Section 3.2 presents the proposed method, and Section 3.3 details the experiments.

3.1 Introduction

As described in Section 2.1, previous studies have developed feature extraction models for estimating track-level music similarity, and recent work has demonstrated the effectiveness of data-driven deep learning approaches, particularly deep metric learning. Deep metric learning, however, requires sampling positive and negative examples for an anchor, typically using available labels or metadata such as artist name or genre for triplet construction. In contrast, labels suitable for part-level similarity learning do not exist, such as "music track A and music track B are similar in drums but dissimilar in vocals". To address this limitation, we consider an unsupervised learning approach, inspired by methods in track-level similarity tasks that divide a music track temporally and assume segments from the same track are similar while segments from different

music tracks are dissimilar [17]. To extract part-specific features, we obtain individual instrumental part signals and apply this assumption to them. In other words, segments from the same instrumental part within a music track are encouraged to be close in the embedding space. Instrumental part signals are divided temporally, and the feature extractor is trained to bring the same-part segments closer, ultimately computing distances between learned embeddings.

However, actual instrumental part signals, i.e., the stems before mixing, are generally unavailable for most music tracks. For practical inference, these signals must be estimated, so we use source separation techniques. We then train separate feature extraction networks for each estimated instrumental part signal. In our experiment, we evaluate the feature representation using the track ID prediction task.

3.2 Method

We explain the proposed method in terms of the overall framework and the training approach.

3.2.1 Framework

We construct individual feature extraction networks for instrumental parts. First, the music signal is separated into each instrumental part signal using a source separation model. These estimated signals are converted into the mel spectrograms. Then, these mel spectrograms for each instrumental part are input into the corresponding feature extraction networks. Each network is trained to learn an embedding space in which distances reflect the similarity of sounds for a given instrumental part: smaller distances indicate higher similarity, whereas larger distances indicate lower similarity.

For example, in the drum-part space, if the drum sounds of song A and song B are similar, they are embedded close to each other. Finally, the part-level similarity is estimated using the Euclidean distance between extracted embeddings as *dissimilarity*. These distances are intended for downstream tasks, such as music retrieval, where they can be used to rank songs in ascending order of their distance to a given query.

3.2.2 Unsupervised Data Sampling for Triplet Learning

To train the feature extraction networks, we employ a metric learning framework with triplet loss. As described in Section 2.1.3, triplet-based metric learning requires sampling triplets consisting of an anchor, a positive sample that is similar to the anchor, and a negative sample that is dissimilar. For each network, triplets must be constructed based on the similarity between corresponding instrumental part signals. Since explicit labels indicating part-level similarity are unavailable, we adopt an unsupervised learning approach. Specifically, each instrumental part signal is segmented along the time axis, where segments from the same signal are treated as similar, and those from different signals are treated as dissimilar. The distance function is Euclidean distance.

3.3 Experimental Evaluation

We experimentally evaluate the effectiveness of our proposed method in estimating part-level similarity. The experimental conditions, evaluation method, and results are described in the following sections.

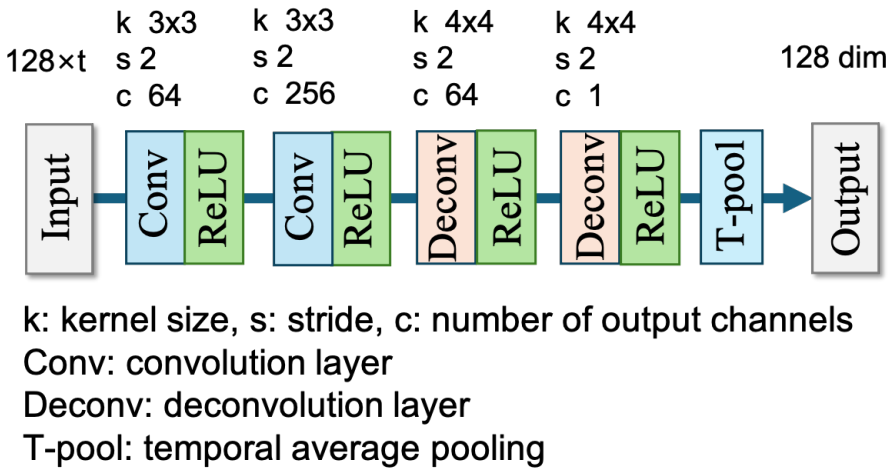


Figure 3.1: *Network architecture. The numbers above input and output are their data sizes.*

3.3.1 Experimental Conditions

Setting

In our framework, which aims for practical applicability, we performed source separation to obtain the individual instrumental part signals from music tracks, both in training and inference. We extracted drums, bass, and piano parts from the music tracks. Also, we conducted experiments using *ideal* instrumental part signals to examine the effect of source separation performance on similarity estimation. The ideal instrumental part signals mean the signals obtained through perfect source separation, that is, equivalent to the instrumental part signals before mixing. As a baseline, we also calculated track-level similarity. The method for estimating track-level similarity was fundamentally the same as the proposed method, except that the input was replaced with the music track during training and inference.

Dataset and Input Features

The dataset we used is Slakh2100 [102], which contains audio tracks synthesized from MIDI (non-vocal music tracks). Slakh2100 also contains the stems of the music tracks, i.e., individual instrumental parts that make up each music track. The individual stems in the dataset can be grouped into broad instrument categories following the dataset's published *recipe*. In this work, we classified the stems into four categories: drums, bass, piano, other, and mixed all stems within the same category to obtain a single instrumental part signal, which is defined as an ideal instrumental part signal. Note that we used the *redux* subset of Slakh2100, which was created by omitting some tracks so that each MIDI file only occurs once. The redux subset has a total of 1,710 tracks: 1,289 in train, 270 in validation, and 151 in test. The sampling rate for all of the data was 44.1 kHz. Each audio track was divided along the time axis. All segments were converted into dB-scaled mel spectrograms with 128 mel bins, using a window length of 2048 and a hop length of 512, and then normalized to the range [0, 1] as input.

Separation Model and Network Architecture of Feature Extractor

We used a pre-trained model "Spleeter" [103] for source separation. Its separation performance for the dataset used in this experiment is shown in Table 3.1. We used convolutional networks for feature extraction, which have been widely employed in previous studies due to their effectiveness when applied to mel spectrograms, as discussed in Section 2.1.4. The network architecture of the feature extractors is shown in Figure 3.1, which extracts a 128-dimensional embeddings from a mel spectrogram.

Table 3.1: *Average separation performance for the estimated instrumental part signals used as training data and test data in the experiment. These values were calculated using "fast_bss_eval" [104]. SDR, SIR, and SAR represent Signal-to-Distortion Ratio, Signal-to-Interference Ratio, and Signal-to-Artifacts Ratio, respectively, evaluating overall separation quality, residual interference, and artifacts introduced by separation (higher is better for all metrics).*

instrument	SDR	SIR	SAR
	train / test	train / test	train / test
drums	0.73, 2.73	12.36, 18.74	2.13, 1.51
bass	−3.42 , −4.79	2.07, 2.12	−1.76, −6.64
piano	−11.20 , −9.66	−4.82 , −3.29	−9.66 , −8.30

Training Conditions

For training, we used 1,200 music tracks from the train set in the Slakh-redux subset. These tracks were divided into three-second segments with 50% overlap, and the first 40 segments were used in each music track for the anchor of the triplet loss, excluding the silent segment. Silent segments were defined as segments where the average power was below the threshold: 0.0001. The margin of the triplet loss was set to 0.2. A batch size was set to 256. The number of epochs was set to 200.

Testing Conditions

For testing, we used the test set of the Slakh-redux subset, excluding tracks whose number of segments (after removing silent sections) was in the lowest 10%, leaving 136 tracks. The tracks were divided into three-second segments without overlaps, and all segments were used except for the silent segment, whose definition is the same as

training.

3.3.2 Evaluation Method using Track ID Prediction

We need to evaluate whether the part-level similarity estimated by our proposed method for unseen data is appropriate. However, as in the training phase, no similarity labels are available for instrumental part signals. Therefore, we conducted an evaluation based on the same assumption used during training: segments within an instrumental part signal in a music track are considered similar. Under this assumption, if the feature extractor of each instrumental part is successfully trained, the segments with the same track ID are expected to form clusters in each embedding space, even for the test data. To verify this assumption, we evaluated the accuracy of a track ID prediction task. Specifically, we used the K-nearest neighbor (kNN) method ($k = 5$) to predict the music IDs of the test segments. The predicted ID was determined by majority voting among the IDs of the five nearest neighbors in the embedding space. We embedded all test segments into the learned embedding space and predicted the music ID of each segment individually. Prediction was performed by a majority vote over the IDs of the five nearest neighbors, assuming the music IDs of all other test segments were known.

3.3.3 Results

The accuracy of the track ID prediction is shown in Table 3.2. In the proposed method using source separation, the accuracy is lower compared to the condition where ideal instrumental part signals are used as input. As shown in Table 3.1, drums achieve the best separation performance overall, both in terms of reduced interference from

Table 3.2: *Accuracy of track ID prediction task with kNN (%). The "baseline" means a track-level similarity learning method. The "proposed" means a part-level similarity learning method using estimated instrumental part signals by source separation, and "propose (ideal)" means that using ideal instrumental part signals as input.*

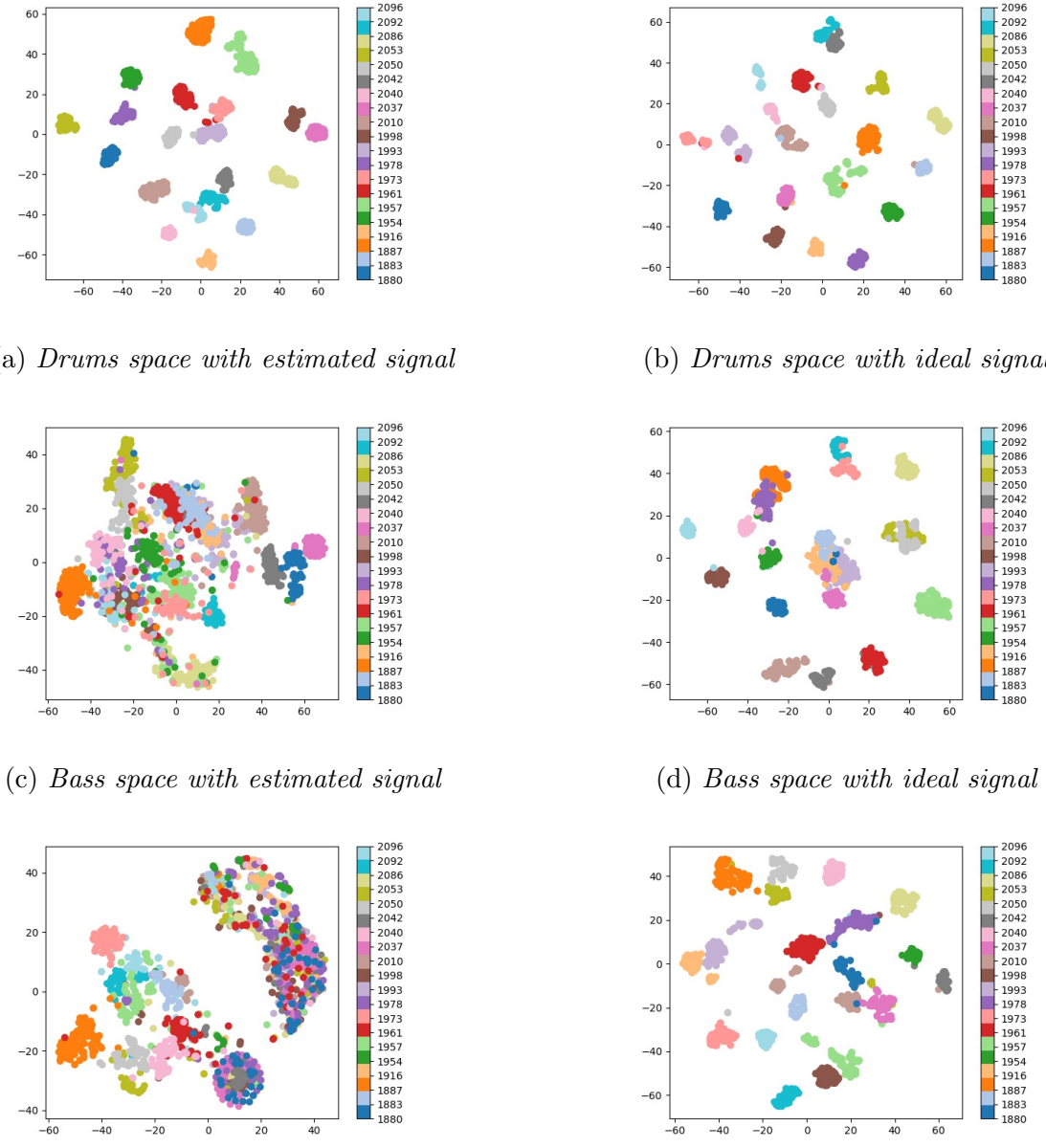
	music track	drums	bass	piano
baseline	95.29	–	–	–
proposed	–	94.00	51.78	39.40
proposed (ideal)	–	95.14	81.47	93.11

other instruments and fewer separation artifacts, whereas piano exhibits the poorest performance, with bass lying in between. When these separation results are considered together with the track ID prediction performance, it can be observed that lower separation quality leads to poorer track ID prediction accuracy. Notably, this tendency is not observed when ideal instrumental part signals are used as input, where piano does not show substantially lower accuracy than drums. This supports the interpretation that the observed degradation in track ID prediction performance is attributable to imperfect source separation rather than to inherent differences between instrumental parts. To summarize, the accuracy degradation of embeddings appears to be caused by the lower audio quality of the estimated signals compared to the ideal ones, due to artifacts and residual components from other instruments. Under the assumption of perfect source separation, drum-part-level and piano-part-level similarity achieves accuracy comparable to that obtained at the track-level similarity. For bass as well, the performance exceeds 80%. Moreover, as discussed earlier, unlike track-level similarity, these part-level similarities offer the advantage of calculating similarity focused solely on individual instrument parts. An example of the embedding spaces is shown

in Figure 3.2. We can see that embeddings from the same music track are concentrated and clusters are clear when the ideal signals are used as input, or when the separation performance is high. It is visually apparent that lower separation performance results in lower accuracy of the embedded representation. Furthermore, we observe that the similarity relationships among music tracks vary depending on the instrumental part, suggesting that each part space represents a distinct embedding space.

3.4 Conclusions and Discussions

We developed a method to extract feature representation using metric learning to estimate part-level similarity. Music tracks are separated into individual instrumental parts at the signal level using a source separation model, and these separated signals are used as input to the models. From the experimental result, we found that the accuracy of the feature representation is affected by the performance of the source separation model. There is room for improving the accuracy of feature representations through improved source separation performance. Alternatively, methods that do not perform signal-level source separation and are therefore unaffected by separation errors may also prove effective. We propose an improved method in the next chapter, Chapter 4. The evaluation of whether the similarity calculated from learned embeddings in this chapter's method aligns with human judgment will be discussed in the next chapter.



(a) *Drums space with estimated signal*

(b) *Drums space with ideal signal*

(c) *Bass space with estimated signal*

(d) *Bass space with ideal signal*

(e) *Piano space with estimated signal*

(f) *Piano space with ideal signal*

Figure 3.2: *The examples of embeddings visualized in two dimensions using dimensionally reduction with t-SNE [105]. Representations are extracted from three-second segments in the part of the Slakh2100 test set. Different time frames from the same track are plotted with the same color. The numbers beside the color bars denote the track IDs in the Slakh2100 dataset.*

4 Part-Level Similarity Estimation with Feature-Level Separation

In this chapter, we describe another method of ours for estimating part-level similarity. We propose a method that extracts part-level feature representations using music tracks as input, without separating into instrumental part signals, i.e., feature-level separation. An overview of this approach is shown in the lower part of Figure 1.2. Section 4.1 provides an introduction, Section 4.2 describes the Conditional Similarity Networks (CSNs) used in our proposed method, Section 4.3 presents the proposed method, and Section 4.4 details the experiments.

4.1 Introduction

We proposed a representation learning method for estimating part-level similarity using separated instrumental part signals as input in Chapter 3. Since the method required individual instrumental part signals, which are usually not publicly available, estimated signals by a source separation model were used as input. However, using the estimated signals resulted in lower accuracy than using the ideal instrumental part signals affected by separation errors.

Another potential approach to extracting part-level representations is feature-level separation, rather than using signal-level separated inputs. In other words, this ap-

proach directly extracts a representation for each instrumental part from the mixture signal. Several methods have been proposed to extract several different conceptual representations from a single input [106], for example, to disentangle the speaker identity and noise in the speech domain [107, 108], timbre and pitch information in the music domain [15, 16, 109], and so on. Veit *et al.* [26] proposed CSNs in the image domain, which learn embeddings differentiated into semantically distinct subspaces that capture the different notions of similarities. Lee *et al.* [17] 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 chapter, we propose a method for extracting instrumental part-level feature representations from a single network using a music track as input, employing CSNs. The proposed network is trained with deep metric learning to embed music tracks into a differentiated embedding 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 data augmentation, auxiliary loss, and pre-training. In the experiments, we investigate whether more accurate embeddings can be obtained using our proposed method than using the baseline method, whether each subspace holds the characteristics of the assigned instrumental part, and whether the learned similarity criterion matches human perception.

4.2 Conditional Similarity Networks (CSNs)

To measure the similarity between images considering multiple notions of similarity, Veit *et al.* [26] 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 is trained by the triplet loss using masks. For the triplet loss, samples $x^{(\mathrm{a})}$, $x^{(\mathrm{p})}$, and $x^{(\mathrm{n})}$ are selected according to condition c that is defined as a certain notion of similarity. Namely, in the notion corresponding to condition c, $x^{(\mathrm{p})}$ is more like $x^{(\mathrm{a})}$ than $x^{(\mathrm{n})}$. To differentiate 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

$$d(x_i, x_j; \mathbf{m}_c) = \| f(x_i)\mathbf{m}_c - f(x_j)\mathbf{m}_c \|_2 . \tag{4.1}$$

Thus, the triplet loss can be written as

$$\begin{aligned} &\mathcal{L}_{\mathrm{triplet}}(x^{(\mathrm{a})}, x^{(\mathrm{p})}, x^{(\mathrm{n})}, c) \\ &= \max\{d(x^{(\mathrm{a})}, x^{(\mathrm{p})}; \mathbf{m}_c) - d(x^{(\mathrm{a})}, x^{(\mathrm{n})}; \mathbf{m}_c) + \delta, 0\}. \end{aligned} \tag{4.2}$$

4.3 Method

4.3.1 Overview

Framework

Similar to the method proposed in Chapter 3, we convert the audio signals into mel spectrograms, input them into feature extractors, and calculate the Euclidean distances between the extracted features as *dissimilarity*. Unlike the previous method, the input is the music tracks, without source separation. A single feature extractor extracts features for each individual part from the music track.

Learning Strategies

To extract part-level features from music tracks by performing separation at the feature level rather than at the signal level, we explore the following learning strategies:

1. Triplet learning with dimension-wise masking inspired by the CSNs.

   As in CSNs, our method learns different notions of similarity for each subspace of the output feature space. In our approach, each subspace is assigned to a specific instrumental part. During training, we sample data based on the similarity of a particular instrumental part, apply masks to the embeddings to preserve only the subspace assigned to that instrument, and input masked embeddings to the loss function. At inference time, the same mask is used to extract the similarity corresponding to the target instrumental part. An overview of this method is shown in Figure 4.1.
2. Data augmentation using *pseudo musical pieces* for triplet sampling.

   As described above, performing separation at the feature level requires an appropriate data sampling strategy, i.e., one based on the similarity of individual instrumental parts. However, as discussed in Chapter 3, labels representing such similarities are not available. Instead, we propose a data augmentation method as an alternative.
3. An auxiliary loss to encourage feature separation.

   An auxiliary loss is introduced to minimize errors caused by feature leakage from other instruments into the subspace assigned to a given instrument.
4. Pre-training to facilitate learning.

   To facilitate learning, we consider a pre-training strategy in which the initial parameters of the network are trained to predict the output features corresponding to the ideal individual instrumental part signals.

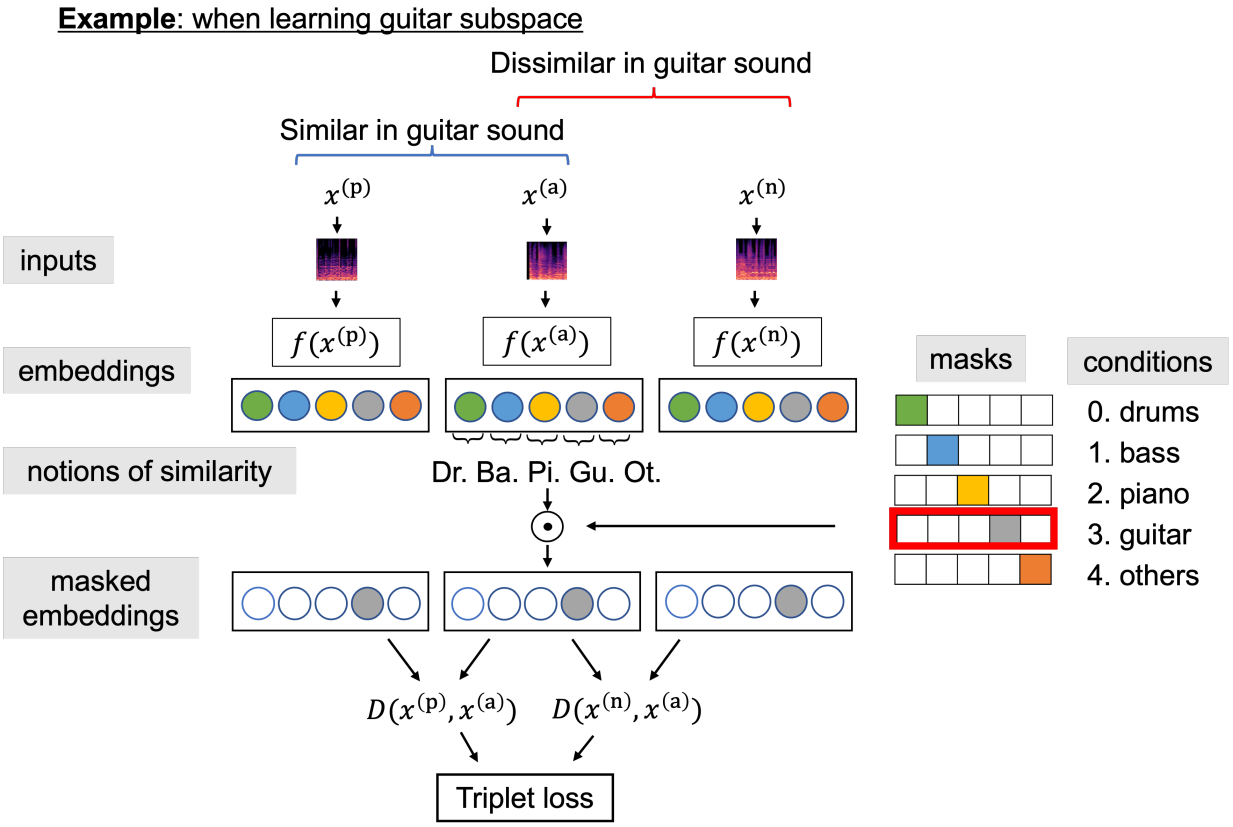


Figure 4.1: *Overview of the proposed method.* $x^{(a)}$, $x^{(p)}$, *and* $x^{(n)}$ *denote the anchor, positive, and negative samples, respectively. "Dr.," "Ba.," "Pi.," "Gu.," and "Ot." are drums, bass, piano, guitar, and others, respectively. This figure shows an example of setting the condition 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 embedding is extracted by the network and is masked so that only the subspace assigned to the guitar is considered in the triplet loss calculation.*

4.3.2 Triplet Learning via CSNs

In this method, the CSNs described in Section 4.2 are used, with each notion of similarity defined as each instrumental part-level 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 differentiate the embedding space into subspaces that each represent part-level similarity. 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.

$$m_{ck} = \begin{cases} 1, & (cD \leqq k < (c+1)D) \\ 0, & (\text{otherwise}). \end{cases} \tag{4.3}$$

The triplet loss in CSNs shown in Equation 4.1 is used for training with the mask described above. The distance function is Euclidean distance.

4.3.3 Pseudo Musical Pieces

As mentioned in Section 4.3.2, we need to sample triplets to learn each part-level similarity. However, there is no label that evaluates instrumental part-level similarity. Therefore, we use the unsupervised learning method similar to the previous methods. However, simply using the assumption that segments from the same music track are similar is not appropriate to obtain distinct representations for part-level similarity (Figure 4.2). This is because the sampling used for training is the same across all subspaces, which would lead to all subspaces learning the same similarity criteria, contrary to our goal of differentiating them into instrument-specific representations.

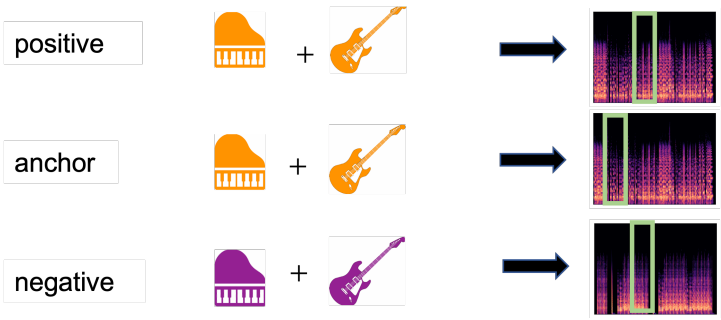


Figure 4.2: *Triplet of dataset musical pieces sampled simply using the assumption that segments from the same music track are similar. Each color represents a track ID. This example shows that the same triplet is used for both learning piano and guitar subspace when this assumption is applied to the dataset pieces. (In this example, music tracks only contain piano and guitar parts for the reasons explained in Section 4.3.4).*

To successfully train the separated subspaces, we propose a method to create a *pseudo musical piece* for input by mixing instrumental part signals in different musical pieces. For example, when the drum part contained in piece A is called drum part A, a pseudo musical piece can be created by mixing the drum part A with other instrumental parts from another piece B. We denote this piece's label as $(\mathrm{A}, \mathrm{B})^{(\mathrm{dr,else})}$ 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. To distinguish the pseudo musical pieces, we refer to the original musical pieces included in the dataset as *dataset musical pieces* throughout this chapter.

Basic Triplet

Considering that musical pieces with the same label for a particular instrument are similar to each other on that instrumental part, a triplet sample can be created. We can say that segment 1, randomly extracted from the musical piece with label $(\mathrm{A}, \mathrm{B})^{(\mathrm{dr,else})}$,

and segment 2, randomly extracted from the musical piece with label $(\mathrm{A}, \mathrm{C})^{(\mathrm{dr,else})}$, are similar in drum part but dissimilar in instrumental parts other than drums. On the other hand, segment 1 and segment 3 with label $(\mathrm{D}, \mathrm{B})^{(\mathrm{dr,else})}$ are dissimilar in the drum part but similar in other instrumental parts. Therefore, segments with label $\{(\mathrm{A}, \mathrm{B})^{(\mathrm{dr,else})}, (\mathrm{A}, \mathrm{C})^{(\mathrm{dr,else})}, (\mathrm{D}, \mathrm{B})^{(\mathrm{dr,else})}\}$ can be used as an anchor, a positive sample, and a negative sample in learning drums subspace. We can also sample the triplets for other instrumental parts in the same way. The triplets extracted in this way are called the basic triplets.

Additional Triplet

We further add triplets of interchanged positive and negative samples under a condition other than that of the basic triplet to encourage each subspace to explicitly learn a different similarity criterion. The anchor and the negative sample in the above example for the basic triplet, the segments from pieces with labels $(\mathrm{A}, \mathrm{B})^{(\mathrm{dr,else})}$ and $(\mathrm{D}, \mathrm{B})^{(\mathrm{dr,else})}$, are dissimilar in drum part but similar in instrumental parts other than drums. Thus, samples with label $\{(\mathrm{A}, \mathrm{B})^{(\mathrm{dr,else})}, (\mathrm{D}, \mathrm{B})^{(\mathrm{dr,else})}, (\mathrm{A}, \mathrm{C})^{(\mathrm{dr,else})}\}$ 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.

An example of these triplet extraction processes is shown in Figure 4.3. Note that negative samples in the basic triplet are selected in such a way that additional triplets can be constructed. Specifically, in this example, the instrumental parts other than drums share the same labels as those of the anchor.

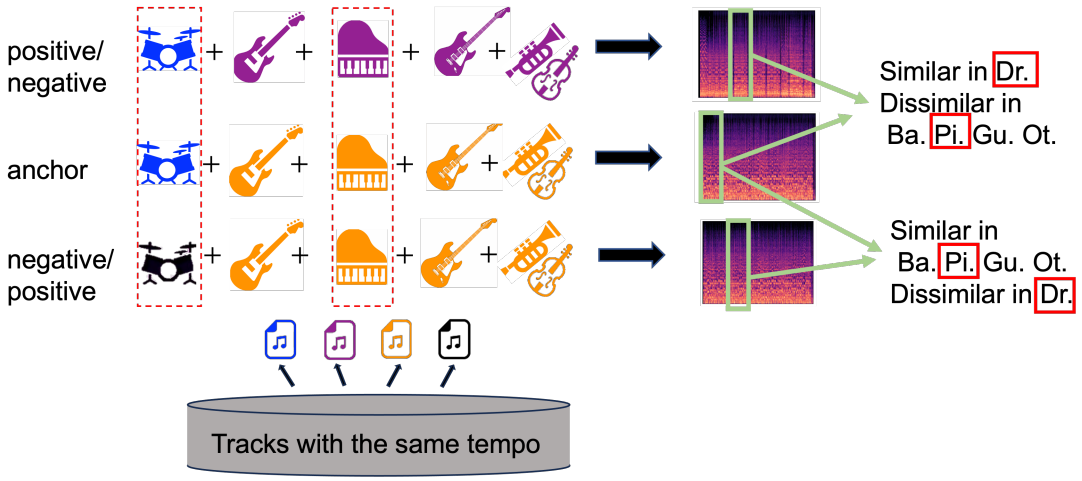


Figure 4.3: *How to create the basic triplet and the additional triplet with pseudo musical pieces. Each color represents an ID in the dataset musical piece, and four musical pieces are randomly selected from the same tempo class in the train set, and pseudo musical pieces are created using the individual instrumental part signals in them. 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.*

4.3.4 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 certain instrumental parts, the corresponding subspaces do not include any information derived from the signals that are present in the input. This is expected to prevent contamination by information computed from signals of other instrumental parts.

We use the Binary Cross Entropy Loss (BCELoss) to satisfy this constraint. The input to the BCELoss is a vector $\mathbf{p}_i$ whose dimensionality corresponds to the number of instrumental categories; in this work, it is five-dimensional. Each element is computed from the norm of the corresponding subspace for x_i. Each value of $p_{ij}, (j = 0, 1, 2, 3, 4)$ is calculated by taking the logarithm to the norm of the masked embedding $f(x_i)\mathbf{m}_c, (c = j)$ and then adding a learnable parameter b_i:

$$p_{ij} = \sigma(\log(||f(x_i)\mathbf{m}_j||_2) + b_{ij}), \tag{4.4}$$

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

$$q_{ij} = \begin{cases} 1, & (P_{\text{avg}}(x_{ij}) > threshold) \\ 0, & (\text{otherwise}), \end{cases} \tag{4.5}$$

where P_{avg} means a time average power of the signal, and x_{ij} is a clean individual instrumental part signal contained in x_i, where the subscript represents each instrument. When $\mathbf{p}_i$ computed from the i-th anchor $x_i^{(\text{a})}$ is denoted as $\mathbf{p}_i^{(\text{a})}$, and in the same way

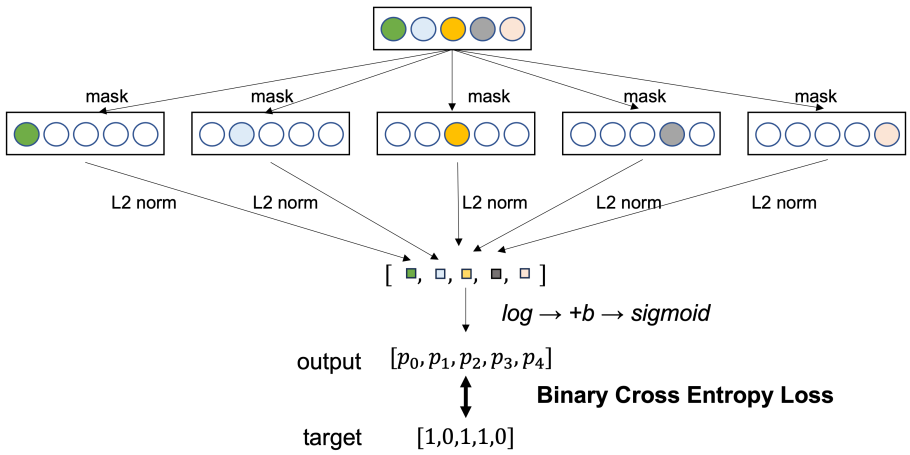


Figure 4.4: *Procedures for calculating norm loss. This is an example of when a track containing only drums, piano, and guitar is input, where the bass's subspace and other's subspace are trained to be close to the zero vector.*

for a positive sample and a negative sample, the formulation of the norm loss is as follows.

$$\begin{aligned}
&\mathcal{L}_{\text{norm}}(x_i^{(\text{a})}, x_i^{(\text{p})}, x_i^{(\text{n})}) \\
&= \frac{1}{3}\{BCE(\mathbf{p}_i^{(\text{a})}, \mathbf{q}_i^{(\text{a})}) + BCE(\mathbf{p}_i^{(\text{p})}, \mathbf{q}_i^{(\text{p})}) + BCE(\mathbf{p}_i^{(\text{n})}, \mathbf{q}_i^{(\text{n})})\}, \\
&BCE(\mathbf{p}, \mathbf{q}) = \frac{1}{5}\sum_{j=0}^{4}\{q_j \log p_j + (1-q_j)\log(1-p_j)\}\,.
\end{aligned} \tag{4.6}$$

This procedure is shown in Figure 4.4. 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 part included in the train data.

4.3.5 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) to estimate the concatenation of embeddings extracted from ideal instrumental part signals. Namely, the concatenation of $g_j(x_{ij})$ 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. The individual networks are trained in the same way as described in Chapter 3, except that the input signals are replaced with ideal instrumental part signals instead of the estimated ones. When extracting the target features for MSELoss, i.e., during inference of the individual networks, the inputs are also the ideal instrumental part signals. For the same reasons explained in Section 4.3.4, the ground truth sub-embedding is set to the zero vector if the input musical piece does not contain the corresponding instrumental part. Figure 4.5 shows a way of creating the target embeddings. x_{ij} is the ideal instrumental segment of instrument j contained in the i-th the dataset musical piece's segment x_i. $\frac{\mathbf{y}_i}{\|\mathbf{y}_i\|_2}$ is the target embedding for the network training, which is created by concatenating embeddings extracted from $x_{ij}, (j = 0, 1, 2, 3, 4)$ using the individual networks and divided by the norm. The formulations of the loss function of pre-training $\mathcal{L}_{\text{pre}}$ and the target embedding are as follows:

$$\begin{aligned} &\mathcal{L}_{\text{pre}}(x_i) = \left(f(x_i) - \frac{\mathbf{y}_i}{\| \mathbf{y}_i \|_2} \right)^2, \\ &y_{ik} = g_j(x_{ij})_k, (jD \leqq k < (j+1)D). \end{aligned} \tag{4.7}$$

This loss function is averaged within the mini-batch.

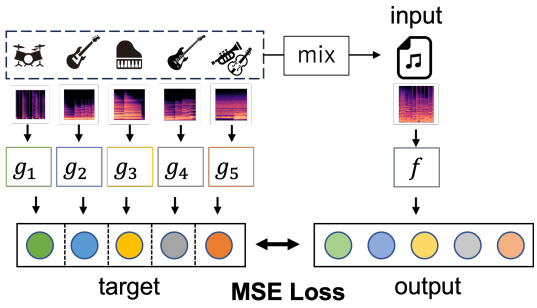


Figure 4.5: *Procedures for pre-training. The network f is trained so that its output is close to the target by the MSE loss. The input of the network f is the segments of the dataset musical pieces. The target is the concatenation of embeddings extracted from the ideal instrumental part signals.*

4.4 Experimental Evaluation

4.4.1 Experimental Conditions

Dataset and Input Features

The dataset we used is the *redux* subset of Slakh2100 [102], which is the same as the one used in the experiment in Chapter 3. Following Slakh's recipe, individual instrumental part signals, namely, drums, bass, piano, and guitar part signals, were created from their stems, and the stems that did not fit into any of the four instruments were mixed as "others."

In pre-training, we used the 200 dataset musical pieces and their ideal instrumental part signals in the training set, both for pre-training the proposed network with $\mathcal{L}_{\text{pre}}$ and training individual networks to extract the targets for pre-training. Moreover, the pseudo musical pieces for training were created with the instrumental parts contained in the 1,200 musical pieces in the training set for training with $\mathcal{L}_{\text{triplet}}$. The 270 musical

pieces in the validation set were used to crerate the pseudo musical pieces for validation. The redux test set was used for testing, but musical pieces with the shortest non-silent sections in individual instrumental parts 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 dataset was classified into 36 classes according to tempo, and the instrumental parts 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 part signal were generated. The 5,000 triplet pseudo musical pieces were randomly created every epoch using 1,200 musical pieces for metric learning with $\mathcal{L}_{\text{triplet}}$.

Both the dataset musical 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 into dB-scaled mel spectrograms with 128 mel bins, using a window length of 2048 and a hop length of 512, and then normalized to the range [0, 1] as input for the training, validation, and testing.

Network Architecture

We used the network shown in Figure 4.6, which had 10 convolutional layers with batch normalization and ReLU, and Max pooling applied every two convolutional layers. The encoder portion of U-Net [110, 111] was referenced. This network was trained to extract a 640-dimensional embedding vector from a mel spectrogram as embeddings. The 640-dimensional embedding was aimed to have 128-dimensional subspaces assigned to each of the five instruments. The subspaces were assigned to drums, bass, piano,

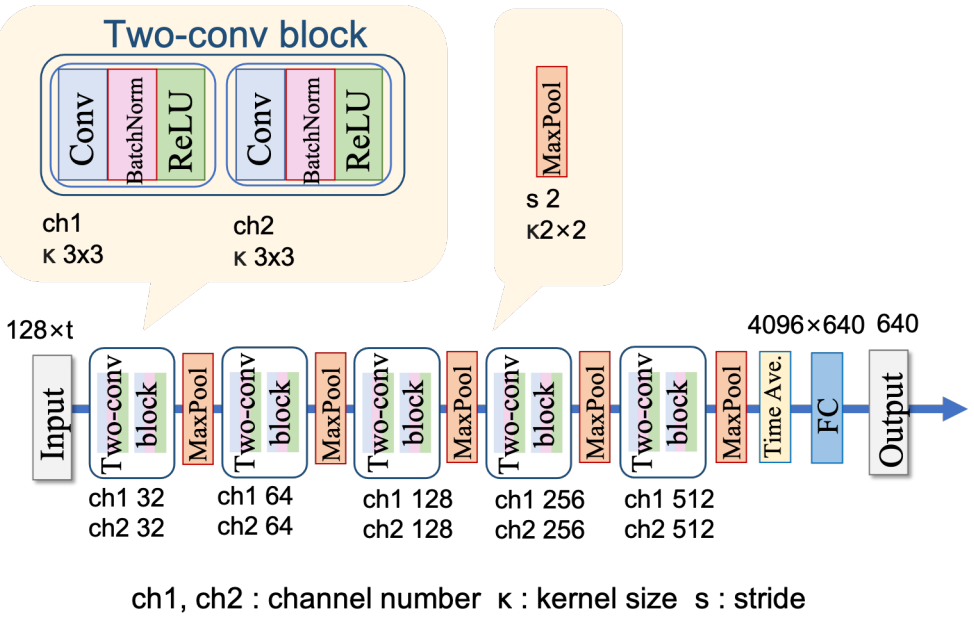


Figure 4.6: *Network architecture. "ch1" and "ch2" denote the channel number, and "κ" and "s" denote kernel size and stride, respectively. "Conv" and "FC" denote the convolutional and fully connected layers, respectively. "Time Ave." means to take an average in the time direction. The numbers above input, 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.*

guitar, and others in order of increasing dimensions.

Pretraining Conditions

For each instrument, a convolutional network (Figure 3.1) was trained as an individual network to extract the target of the pre-training. Then, we pre-trained the network shown in Figure 4.6 using the segments of the dataset musical pieces as inputs and the concatenations of outputs of individual networks as targets. We used the ideal

instrumental part segments as input for the individual networks both in training and inference to create the target embeddings.

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.

Baseline Model

We used our method using the source separation (described in Chapter 3) as the baseline model.

4.4.2 Evaluation Method

We conducted experimental evaluations to investigate whether the following purposes of this study were achieved: (P1) to learn embeddings in which similar tracks are close and dissimilar tracks are far from each other more accurately than the baseline method, (P2) to output each part-level similarity in the subspace assigned to each instrumental part, (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.

Track ID Prediction Accuracy

In the evaluation on P1, we used the accuracy of music ID prediction with the dataset musical pieces in the same manner as the evaluation in Chapter 3. This evaluation was

based on the assumption that instrumental parts that consist of different time segments of the same music track should be more similar than those of different musical pieces. Specifically, we used the kNN method ($k = 5$) to predict the music IDs of the test segments' embeddings. Let the segment to be predicted be called the target, and the music IDs of all test segments' embeddings except the target were assumed to be known. We predicted the music ID of the target by a majority vote using the IDs of the top five nearest test segments' embeddings, and this was done for all musical pieces in the test set.

To evaluate each instrument's embedding, we extracted it by using only the corresponding subspace with masking, inputting the dataset musical pieces with our proposed method. For the evaluation of the baseline method, the same musical tracks were input into the source separation model [103], and the separated signals were input into each individual network to extract each instrument's embeddings.

Feature-Level Separation Capability

We evaluated the accuracy of the embedding of each subspace in Section 4.4.2 but did not evaluate whether each subspace is separated by the instrumental part. The inputs in the evaluation in Section 4.4.2 had the same label for all instruments because they were the dataset musical 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.

For the evaluation on P2, the part-level label prediction by a similar method as described in Section 4.4.2 was conducted for the test pseudo musical pieces. The pseudo musical pieces were created using the musical pieces in the test set by the same method as described in Section 4.3.3. We evaluated whether different pseudo musical pieces with the same part-level label, e.g., drum label, form a cluster in the

corresponding subspace. Unlike Section 4.4.2, not only the target but also other segments divided from the same pseudo musical piece as the target were removed from the reference. For example, in the evaluation of the drums subspace, the test musical pieces with the same drum label and different other instrument labels were created, as $\{(\mathrm{A},\mathrm{B})^{(\mathrm{dr,else})}, (\mathrm{A},\mathrm{C})^{(\mathrm{dr,else})}, (\mathrm{A},\mathrm{D})^{(\mathrm{dr,else})}, (\mathrm{A},\mathrm{E})^{(\mathrm{dr,else})}, (\mathrm{F},\mathrm{G})^{(\mathrm{dr,else})}, (\mathrm{F},\mathrm{H})^{(\mathrm{dr,else})}...\}$. The correct drum label of a target, a segment from the musical piece with the label $(\mathrm{A},\mathrm{B})^{(\mathrm{dr,else})}$, is "A" but other segments from the musical piece with $(\mathrm{A},\mathrm{B})^{(\mathrm{dr,else})}$ cannot be referred. Hence, only when segments from $(\mathrm{A},\mathrm{C})^{(\mathrm{dr,else})}$ or $(\mathrm{A},\mathrm{D})^{(\mathrm{dr,else})}$ or $(\mathrm{A},\mathrm{E})^{(\mathrm{dr,else})}$ 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. This can detect the leak and correctly evaluate the capability to represent separated embeddings. 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.

Instrumental Sound Identification Accuracy

To confirm that the training with the norm loss described in Section 4.3.4 was successful (P3), we evaluated it with the instrumental sound identification task. We performed this to verify that the subspace corresponding to instrumental parts not included in the input was close to the 0 vector, i.e., that the information for each instrumental part did not leak into the subspace to which it did not correspond.

When an ideal instrumental part signal 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 part for that input. We made the above predictions using instrumental part signals of the musical pieces in the test set and calculated the percentage of correct responses for all instruments.

Correlation Analysis of Learned Similarity Measures

In our method, $f(x_{ij})$, the output when inputting the instrumental part signal x_{ij}, and $f(x_i)\mathbf{m}_j$, the masked embedding when inputting a music track x_i containing that instrumental part signal x_{ij} should represent the same feature. We confirm whether similarity measures between them have a correlation (P3). We calculated the two distance (i.e., dissimilarity) matrices between embeddings of all segments of the dataset musical pieces in the test set for $f(x_{ij})$ 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 high correlation between different subspaces may have resulted in a meaningless space that expresses the same feature in all of them. Correlations between two subspaces obtained from a given music track were also calculated to verify the distinctiveness between the similarity measures (related to P2). 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. The distance function in distance matrices is Euclidean distance.

Subjective Evaluation

We evaluated model performance from perceptual perspectives (P4) using evaluation labels collected from the listening tests introduced in Chapter 5. Since details are explained in Chapter 5, we provide a simplified explanation here. Note that here, we

present only the evaluation using a sample set ($\chi\alpha\beta$ in Chapter 5) composed of three distinct music tracks.

Participants 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 considering the three.

In each response, they can select N/A from up to two perspectives except for *overall*. The following instruction was provided with the participants: "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 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 participants' 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 participants' results were obtained as follows: A+ and A− were treated as the same response, A. The same applied to B. Sample sets with less than 80% agreement among participants were eliminated in the evaluation because the sample set with a low agreement rate among the participants 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 responses to the remaining sample set, A or B, were used as the participants' results.

4.4.3 Results

Track ID Prediction Accuracy

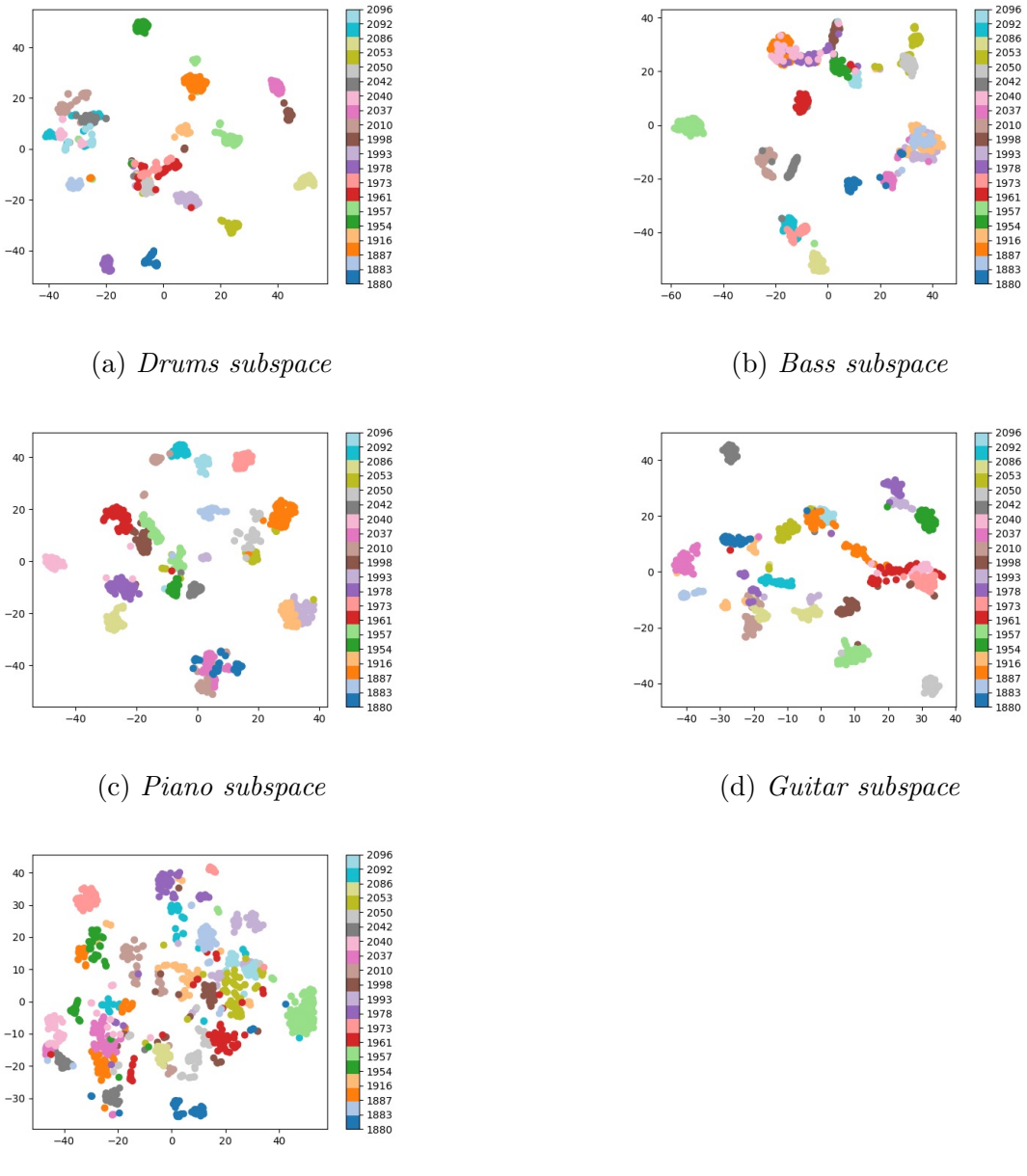
The accuracy of the predicted track IDs is shown in Table 4.1. The table shows the results when 3, 5, and 10 s of data were used as input for inference. Each row represents the instrument to focus on. The column for the proposed method shows the results of inference using only the subspace to which the focused instrumental part is assigned. In contrast, the column for the baseline shows the results of inputting the separated instrument signals to the individual networks.

It can be seen that the baseline using separated instrumental part signals is affected by sound quality degradation due to separation, and the accuracy of embedding degrades, especially on piano with low separation accuracy (as shown in Chapter 3). In contrast, the proposed method shows stable accuracy regardless of which instrument is focused on. A visualization of the subspace is shown in Figure 4.7. It can be seen that segments from the same music track constitute a cluster. Also, we can observe that the similarity measures, i.e., the distance relationship between music tracks, are differentiated across different subspaces.

Feature-Level Separation Capability

Table 4.2 shows the part-level label prediction accuracy 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 the baseline method.

The result (b) shows that the pre-trained model performed better than random



(a) *Drums subspace*

(b) *Bass subspace*

(c) *Piano subspace*

(d) *Guitar subspace*

(e) *Others subspace*

Figure 4.7: *The examples of embeddings with dataset pieces visualized in two dimensions. The 640-dimensional embeddings were extracted from five-second segments in the part of the Slakh2100 test set, masked to extract each subspace, and dimensionally reduced with t-SNE [105]. Different time frames from the same track are plotted with the same color. The numbers beside the color bars denote the track IDs in the Slakh2100 dataset, which is the same as Figure 3.2. Segments of the same track form clusters. Furthermore, the same trend is observed as the plot using the ideal signals in Figure 3.2, regarding which track clusters are close to each other in drums, bass, and piano.*

Table 4.1: *Track ID prediction accuracy. The lines "proposed" and "baseline" show the results using the proposed method and using the separated signals as input to the individual networks.*

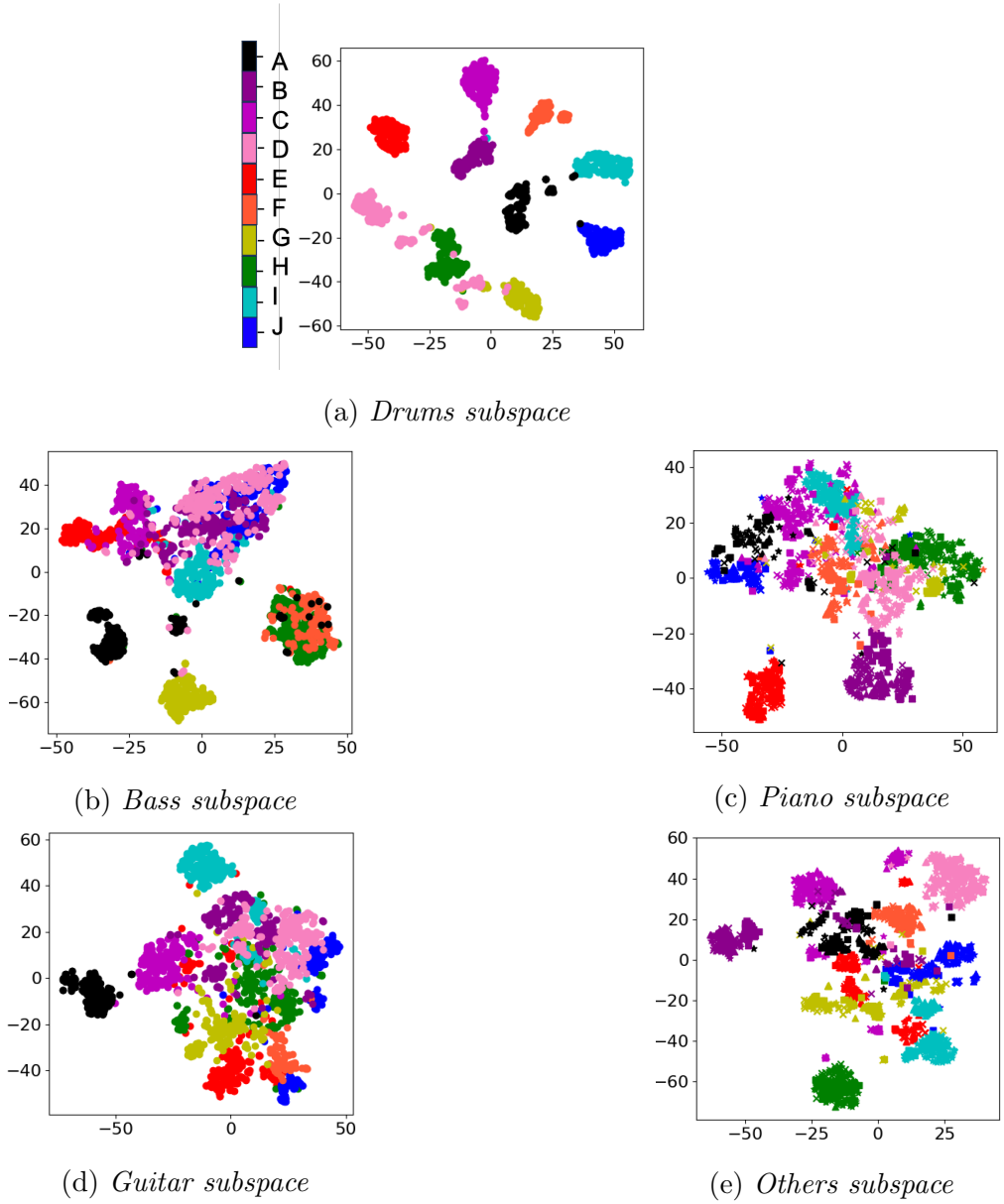
	Input: 3 s of data		Input: 5 s of data		Input: 10 s of data	
instrument	proposed[%]	baseline[%]	proposed[%]	baseline[%]	proposed[%]	baseline[%]
drums	84.73	94.00	86.84	95.24	88.91	94.62
bass	51.01	51.78	59.20	61.54	64.87	70.32
piano	77.21	39.40	82.00	43.27	84.30	45.30
guitar	76.50	–	80.18	–	82.70	–
others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

prediction but worse than the conventional method. The results (c)–(e) indicate that the subspaces trained without pseudo musical pieces (Figure 4.2) do not effectively distinguish between instruments, with only the drum subspace showing high similarity scores. This suggests that all subspaces are learning similar features, which raises concerns discussed in Section 4.3.3 that they may be optimized under the same criteria, preventing proper separation of instrument-specific features. 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 separation 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 4.8 and Figure 4.9. It can be seen that the tracks with the same part-level labels are close to each other.

Table 4.2: *Part-level label prediction accuracy using the pseudo piece. 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. "w/" indicates that the corresponding method is applied.* † *When learning without the pseudo musical pieces, the basic triplet sampling method is shown in Figure 4.2, and the additional triplet cannot be created. Only the results of the five-second segment are shown.*

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



(a) *Drums subspace*

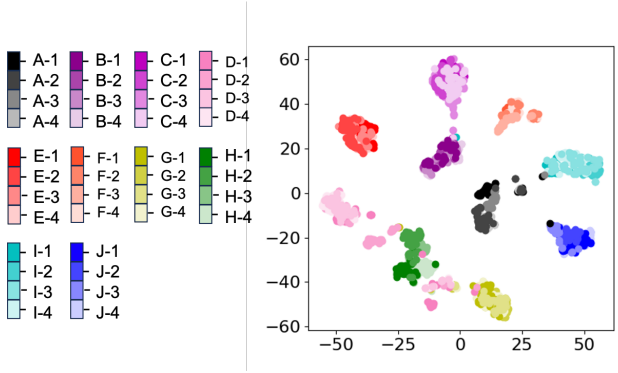
(b) *Bass subspace*

(c) *Piano subspace*

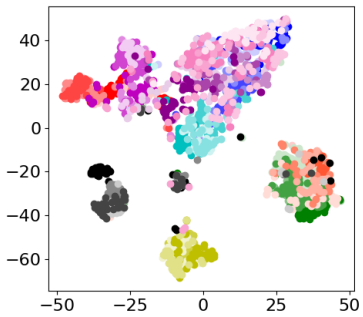
(d) *Guitar subspace*

(e) *Others subspace*

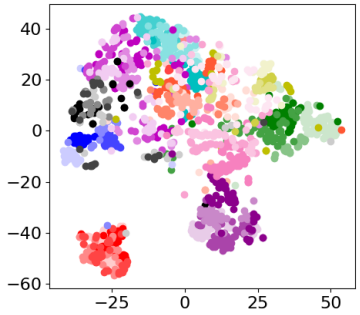
Figure 4.8: *The examples of embeddings with pseudo musical pieces visualized in two dimensions. The 640-dimensional embeddings were extracted from five-second segments in the part of the test pseudo musical pieces, masked to extract each subspace, and dimensionally reduced with t-SNE [105]. In each instrument, the same color represents the same part-level label. For example in (a), a segment with label* $(\mathrm{A}, \mathrm{B})^{(\mathrm{dr,else})}$ *and a segment with label* $(\mathrm{A}, \mathrm{C})^{(\mathrm{dr,else})}$ *are plotted with the same color as they have the same drum label. Track IDs are displayed next to the color bars after being replaced with alphabetic characters to align with the main text explanation. "A", "B", "C", "D", "E", "F", "G", "H", "I", "J" correspond to IDs 1876, 1881, 1882, 1886, 1889, 1899, 1893, 1898, 1902, and 1957 in Slakh2100.*



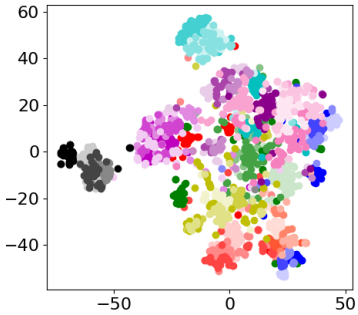
(a) *Drums subspace*



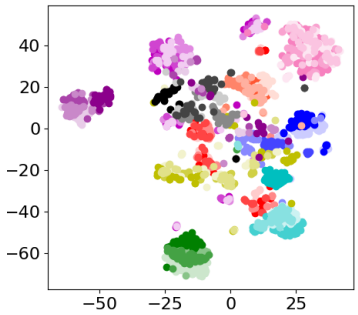
(b) *Bass subspace*



(c) *Piano subspace*



(d) *Guitar subspace*



(e) *Others subspace*

Figure 4.9: *The same results as in Figure 4.8 but different color coding. Only segments with the same label for all instruments are plotted in the same color. For example in (a), a segment with label* $(\mathrm{A}, \mathrm{B})^{(\mathrm{dr,else})}$ *and a segment with label* $(\mathrm{A}, \mathrm{C})^{(\mathrm{dr,else})}$ *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 4.4.2, the same color in Figure 4.8 was regarded as the same label, but segments with the same color in this figure were not used for reference.*

Table 4.3: *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 4.2 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.*

	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

Instrumental Sound Identification Accuracy

Table 4.3 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 4.10.

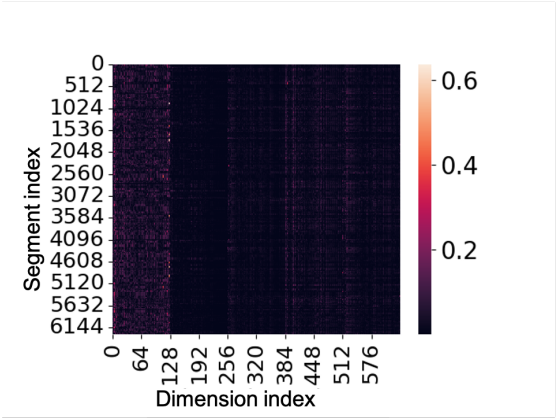
Correlation Analysis of Learned Similarity Measures

Table 4.4 shows the correlation between the distance matrix when an individual instrumental part signal is input and the distance matrix when a music track containing that instrumental part signal is input and masked. It can be seen that the correlation is higher when pseudo musical pieces were used in training, which suggests that training with pseudo musical pieces can help each subspace represent the target instrumental feature. A visualization of the two distance matrices is shown in Figure 4.11 taking the drums as an example. We can see a similar pattern in the similarity measures, both when the instrumental part signals are input and when the music tracks containing them are input and masked.

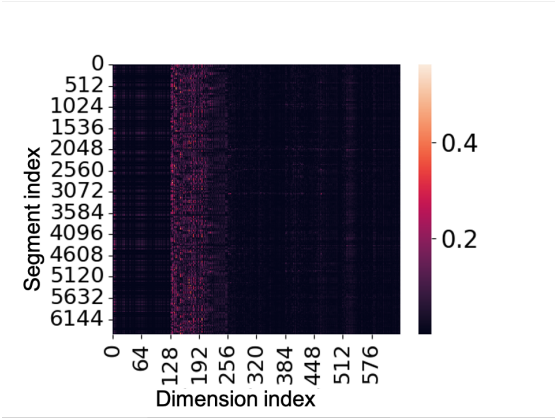
Table 4.5 shows the correlation between pairs of distance matrices across the subspaces. The results with the individual network with ideal individual instrumental part signal input, shown for reference, show low correlations, indicating that the correlation of embedding for each instrument should inherently be low. However, training 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.

Subjective Evaluation

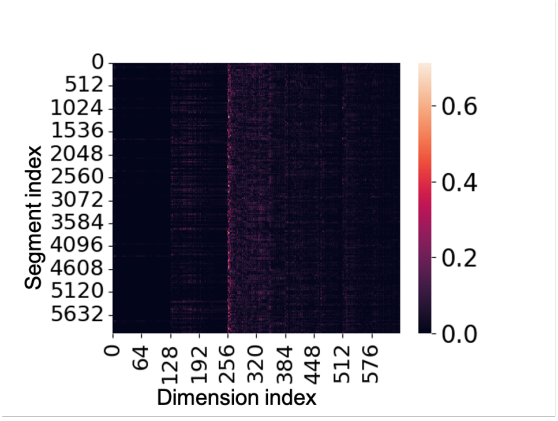
The matching rate with the responses on *overall* was high for rhythm and melody, but low for timbre. In other words, different labels were assigned to the same sample set when focusing on timbre compared to focusing on *overall*. Therefore, we evaluated the model using these two types of responses.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



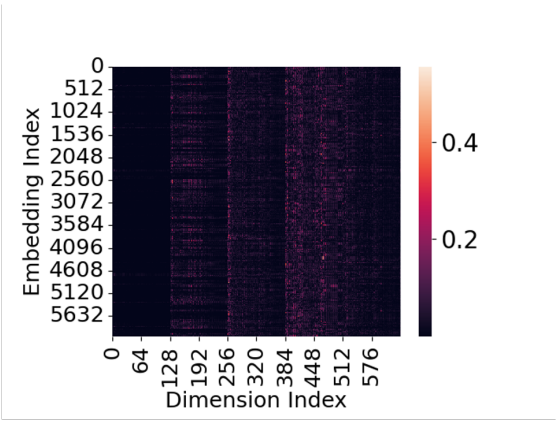
(a) *Drums subspace*



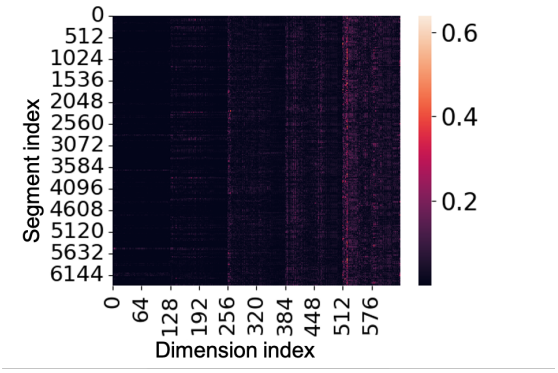
(b) *Bass subspace*



(c) *Piano subspace*



(d) *Guitar subspace*

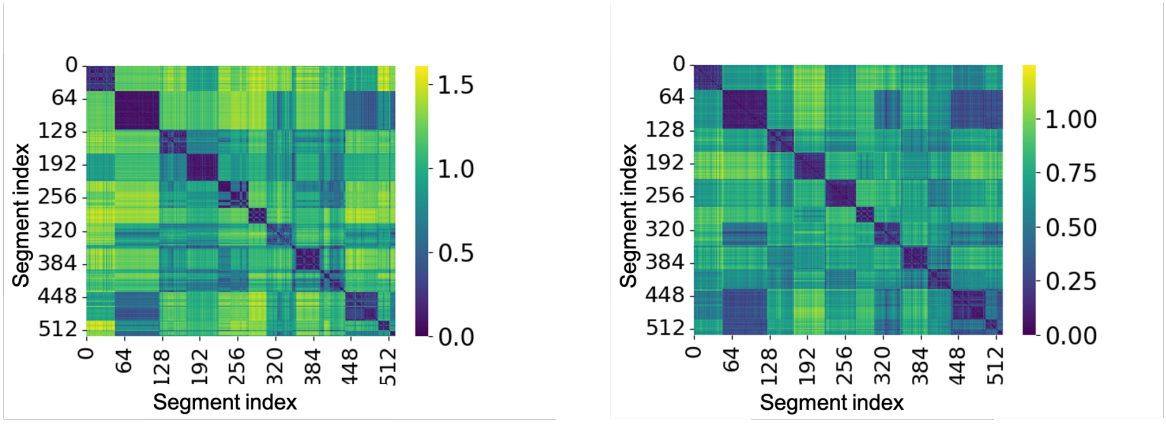


(e) *Others subspace*

Figure 4.10: *Visualization of the absolute values of test embeddings. For each instrument, absolute values are taken for the output embeddings (without masking) when inputting five-second segments of the individual instrument signals in the test set, and stacked vertically. The vertical axis is the index of segments, and the horizontal axis is the index of dimensions.*

Table 4.4: *Correlation between distance matrices in the same subspace with individual instrumental part signal input and with music track input. Each line name can be rephrased corresponding to the results in Table 4.2 as follows: "wo/psd" to norm+basic+pre, and "w/ psd" to norm+psd+basic+add+pre.*

	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



(a) Using the drums subspace embedding with drums signal input

(b) Using the drums subspace embedding with music track input

Figure 4.11: *Distance matrices between the embeddings 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".*

Table 4.5: *Correlation between pairs of distance matrices across the subspaces. The "wo/psd" and "w/psd" represent the same mean as Table 4.4.*

	w/ ideal instrumental part signals (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 4.6: *Matching rate between the model's results and participants' results focusing on overall and timbre, respectively.*

	Evaluation with responses on **overall** (%)				
	drums	bass	piano	guitar	others
baseline	56.5±3.3	56.0±3.2	58.4±3.4	–	–
proposed	56.7±3.3	68.3±3.1	57.7±3.4	61.7±3.2	60.0±3.1
	Evaluation with responses on **timbre** (%)				
	drums	bass	piano	guitar	others
baseline	61.6±4.6	55.0±4.9	59.1±5.5	–	–
proposed	66.1±4.5	69.3±4.7	70.6±5.2	73.8±4.5	64.7±4.8

instrument on the two perspectives are shown Table 5.3 in Chapter 5.)

The results are shown in Table 4.6. Compared with the baseline, the matching rates of the proposed method for the drums and bass are comparable, and that for the piano is better.

Accuracy is higher in drums, piano, and guitar in the evaluation using responses focusing on timbre compared with using responses focusing on *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.

4.5 Conclusions and Discussions

In this chapter, we proposed a part-level similarity estimation method using mixed signals as input in one network, which extracts a single similarity embedding space with separated 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 learning strategies and improvement from our method using source separation in perceptual evaluation. It is also suggested that the model has learned a similarity criterion primarily based on timbre. This point will be further discussed in the following chapter.

5 Part-Level Perceptual Similarity Analysis

In this chapter, we investigate perceptual music similarity focusing on each instrumental part. Section 5.1 provides an introduction, Section 5.2 describes the design of our listening test, and Section 5.3 presents the analysis methods and results.

5.1 Introduction

We proposed deep metric learning methods using triplet loss for estimating part-level similarity, in contrast to previous studies that targeted track-level similarity. Ideally, this method would employ a sampling strategy in which samples that are perceptually similar to an anchor are treated as positive samples, while perceptually dissimilar samples are treated as negative samples. However, collecting such perceptual annotations for training at scale is impractical. Instead, we adopted an unsupervised sampling strategy based on the assumption that temporally distinct segments within an instrumental part are perceptually similar. This unsupervised sampling approach raises several points that warrant careful consideration. First, it is unclear whether the assumption that temporally distinct segments within an instrumental part are always perceptually similar holds from a human listening perspective. Second, it remains unclear to what extent the learned similarity reflects perceptual similarity, given that its

sampling strategy differs from the ideal one in terms of data distribution, for example, by limiting the diversity of positive samples. Considering these issues, to enable the development of a perceptually grounded part-level music similarity estimation model, it is necessary to conduct (1) an analysis of perceptual characteristics and (2) a perceptual evaluation of the model estimations. To achieve these, subjective evaluation experiments through listening tests are required.

For many years, work has been done on perceptual music similarity through listening tests. Studies that clarify the factors listeners prioritize when perceiving music [11, 69, 72] contribute to the design of methods that automatically estimate similarity based on listeners' perception. Addressing the construction of ground truth labels [11, 69, 72] is essential for model evaluation. However, no listening tests have been conducted in which participants evaluate the part-level similarity.

To address this limitation, we conducted an ABX-style listening test for the part-level similarity with 632 participants and collected 26,898 responses. The perceptual similarity was evaluated from four perspectives: timbre, rhythm, melody, and *overall*, with *overall* denoting a comprehensive assessment integrating the other three. For developing a perceptually grounded music similarity model, we aim to clarify the following points through our analysis: (1) the difference among part-level similarities in perception; (2) the correspondence between part-level and track-level similarity in perception; (3) the impact of timbre, rhythm, and melody on part-level perceptual music similarity; (4) the corresponcdance between the output of the deep learning models and perceptual similarity in terms of timbre, rhythm, and melody; (5) the validity of the assumption underlying the unsupervised data sampling methods for music similarity estimation, namely that the similarity between temporally distinct segments within the same track is higher than that between segments from different tracks. Furthermore,

the responses collected in this listening test are available as the dataset[1].

5.2 Listening Test Design

In this section, we describe the listening test. The listening test was conducted on the web page shown in Figure 5.1 with the participants recruited through crowdsourcing. Following previous work that targeted non-musicians [7], we aimed to collect evaluations from a large number of participants. To investigate the part-level perceptual music similarities, we conducted a listening test based on the ABX test on the similarity of sample sets consisting of instrumental parts and original music tracks. Participants provide responses for each *sample set*, which refers to a set of three audio samples labeled A, B, and X. A collection of sample sets that participants respond to in a single listening test session is called an *evaluation set*. For each sample set, participants provide responses from four perspectives: timbre, rhythm, melody, and *overall* for each sample set. These perspectives were determined based on previous findings indicating that music similarity is influenced by genre, tempo, and timbre [72], by rhythm and pitch [11], and by melodic [12, 69] and timbral [77, 78] characteristics. Each sample set consisted of either three different samples from the same instrument category (e.g., all three are drum sounds) or three different music tracks. One evaluation set included instrumental parts from five different categories (drums, bass, piano, guitar, others) and original music tracks.

[1] https://github.com/zume06/Inst-Sim-ABX-Dataset

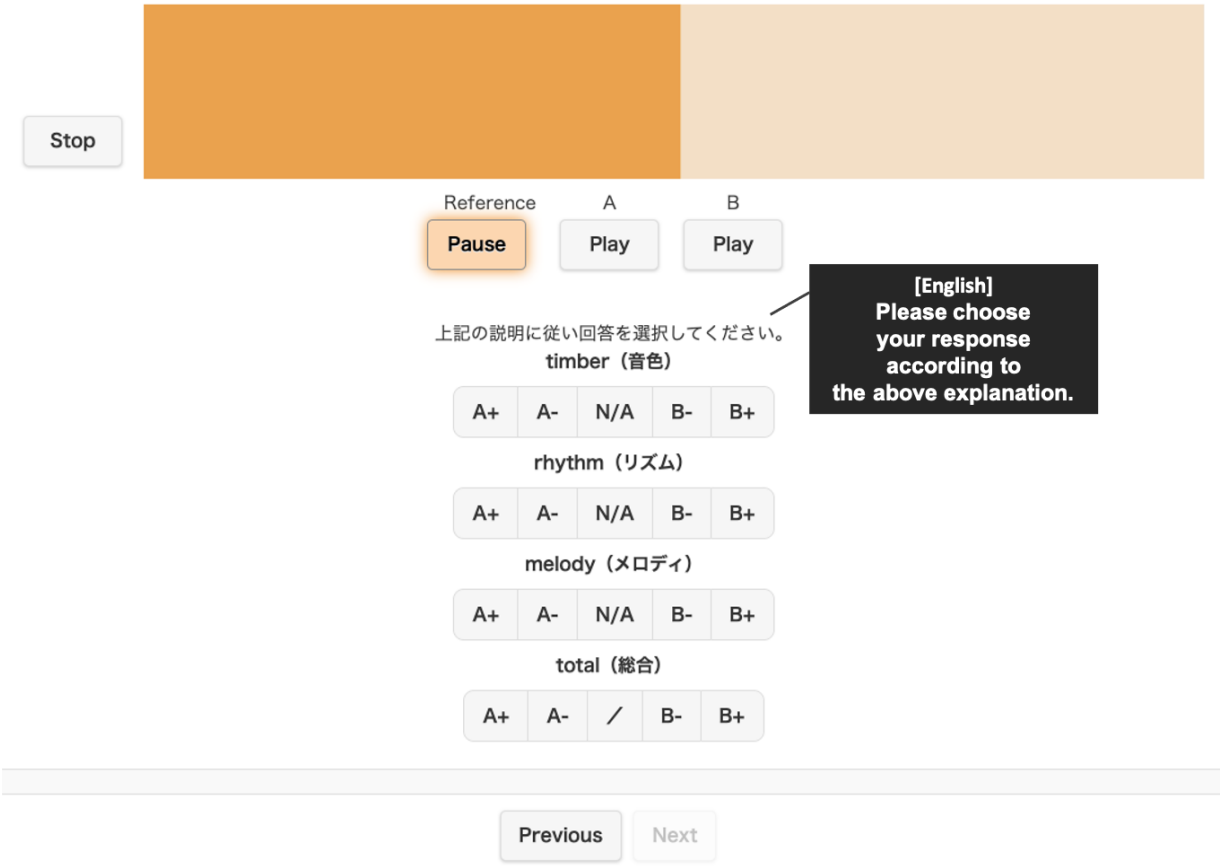


Figure 5.1: *Image of the web page used in the listening test. Each sample set has play buttons for three samples and response buttons for four perspectives: timbre, rhythm, melody, and the* total *similarity (i.e., the overall similarity across these three perspectives; hereinafter referred to as "overall" in the paper). In this figure, "Reference" corresponds to X as used in this paper. The actual screen shows the instructions for the procedure of the listening test above this. In addition to the content described in Section 5.2, the instructions state that participants are required to listen to each sample from the beginning to the end. Furthermore, the "Next" button remains disabled until all samples have been played and responses satisfying the specified constraints (will be described in Section 5.2) have been selected.*

5.2.1 Evaluation Procedure for One Sample Set

As shown in Figure 5.1, participants were presented with three audio samples, 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*.

In each response, the participants were allowed to select N/A from up to two perspectives, except for *overall*. The following instruction was provided to the participants as cases where N/A can be selected: "A and B are similar/dissimilar to X of equal degree" or "The presented instrumental track has no element corresponding to the perspective, e.g., drums have no melody."

5.2.2 Sample Selection

An overview of sample selection is shown in Figure 5.2. We created two types of sample set: "$\chi\alpha\beta$" and "$\chi\chi'\gamma$". For $\chi\alpha\beta$, we randomly selected three different music tracks $\{\chi_i, \alpha_i, \beta_i\}$ from the test set of the Slakh2100 dataset. This combination is referred to as a *music triplet*. We randomly captured five-second segments from each instrumental part contained in the three tracks of that music triplet. Five-second segments were also randomly extracted from each original music track. Then, we obtained one sample set for each instrument or music track, {X, A, B}=$\{\chi_{ij}[t_{ij}^{(\chi)} : t_{ij}^{(\chi)} + 5], \alpha_{ij}[t_{ij}^{(\alpha)} : t_{ij}^{(\alpha)} + 5], \beta_{ij}[t_{ij}^{(\beta)} : t_{ij}^{(\beta)} + 5]\}$. The subscript j represents each category of instrument ($j = 0, ..., 5$), where 0 represents drums, 1 bass, 2 piano, 3 guitar, 4 others, and 5 original music track. This selection was repeated four times ($i = 0, ..., 3$), and 24 sample sets were created (6 categories × 4 repetitions).

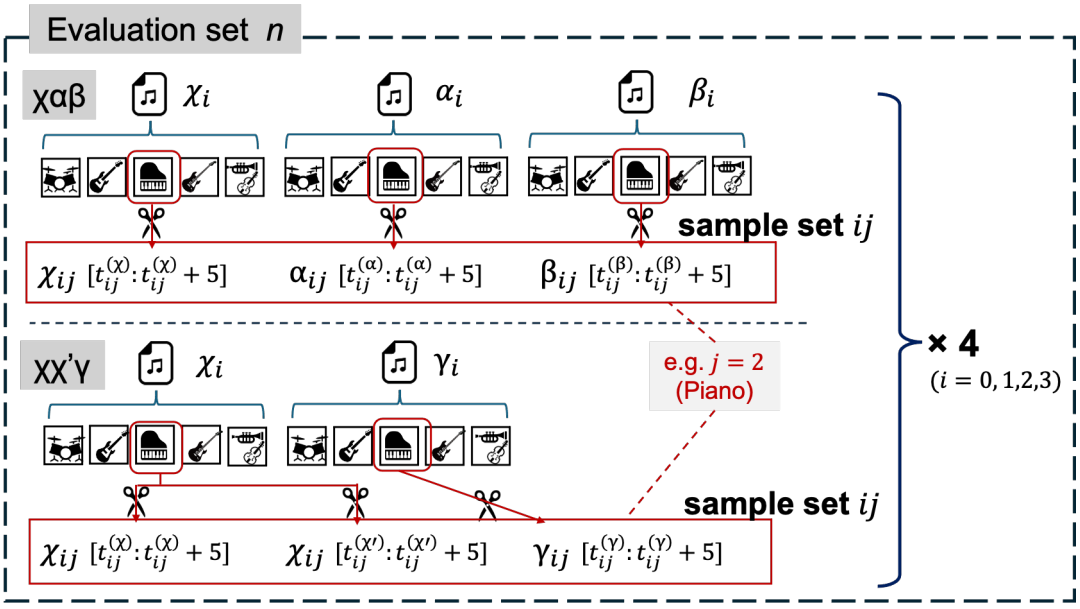


Figure 5.2: *How to create one evaluation set. Examples of how to make one sample set for the piano part are also shown with a red line for* $\chi\alpha\beta$ *and* $\chi\chi'\gamma$*. Sample sets are created in the same manner as for other instrumental parts and the original music track, and the procedure is repeated four times.*

For $\chi\chi'\gamma$, we used the same X as in $\chi\alpha\beta$, and one of the other two samples was taken from temporally distinct segments of the same music track as X. Namely, we randomly selected a music track γ_i excluding χ_i and replaced α_i and β_i with χ_i and γ_i, respectively, and the process was repeated in the same manner as for $\chi\alpha\beta$. The 24 sample sets $\{\chi_{ij}[t_{ij}^{(\chi)} : t_{ij}^{(\chi)} + 5], \chi_{ij}[t_{ij}^{(\chi')} : t_{ij}^{(\chi')} + 5], \gamma_{ij}[t_{ij}^{(\gamma)} : t_{ij}^{(\gamma)} + 5]\}, (t_{ij}^{(\chi)} \neq t_{ij}^{(\chi')}, i = 0, ..., 3, j = 0, ..., 5)$ were created.

Then, 48 sample sets were created as one evaluation set. This procedure was repeated with random sample selection, resulting in 60 evaluation sets. Hereafter, time indices are omitted for simplicity. To distinguish $\chi_{ij}[t_{ij}^{(\chi)} : t_{ij}^{(\chi)} + 5]$ from $\chi_{ij}[t_{ij}^{(\chi')} : t_{ij}^{(\chi')} + 5]$, we denote the latter as χ'_{ij}.

5.2.3 Setup of the Listening Test

The presented audio samples were 136 music tracks and the instrumental parts that compose them. They were the tracks remaining from the 151 tracks in the test set of the redux subset of the Slakh2100 dataset [102], excluding tracks that do not contain enough instrumental parts. Fifty sample sets, consisting of 48 sample sets created as explained in Section 5.2.2 and two dummy sample sets, were shuffled and presented to the participants individually. The participants were not informed whether the sample set presented was $\chi\alpha\beta$, $\chi\chi'\gamma$, or a dummy. The dummy sample sets were of two types: A is exactly the same audio as X, and B is exactly the same audio as X. In $\chi\chi'\gamma$, one of the two orderings, $\{\chi', \gamma\}$ or $\{\gamma, \chi'\}$, was randomly assigned to {A, B}. Since this study is aimed at establishing a system for use by general users rather than experts, listening tests were conducted by recruiting participants through a crowdsourcing platform, CrowdWorks [112]. Participation was fully voluntary and anonymous, and no personally identifiable information was collected. Participants were free to withdraw from the experiment at any time. The average duration of the experiment was approximately 30 minutes per single session (one participant evaluated one evaluation set), and participants were compensated according to the standard rates of the platform.

5.2.4 Response Aggregation

The following responses were excluded as they were inappropriate for analysis: all responses from participants who did not select the same sample as X in the dummy tests; responses that took less than 15 seconds including listening time; responses with blanks in the technical problems; and duplicate responses from the same participant within each evaluation set. After excluding them, by counting the set of evaluations on

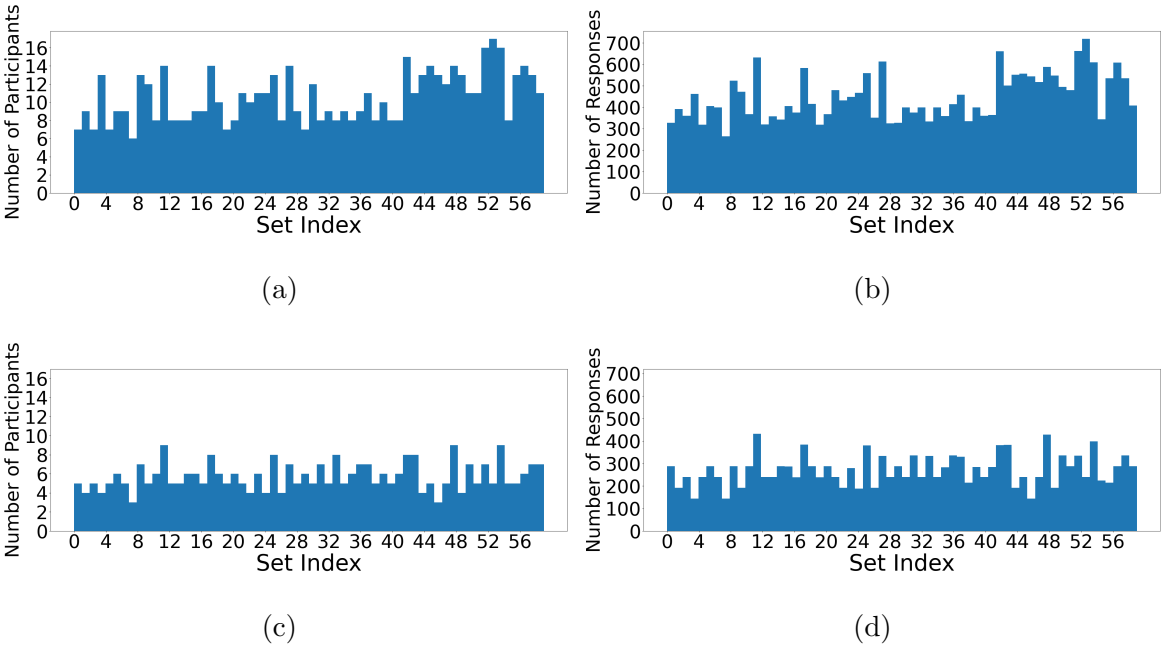


Figure 5.3: *Number of participants providing valid responses and number of valid responses to each evaluation set. (a) Number of all participants providing valid responses, (b) Number of all valid responses, (c) Number of participants who provided valid responses to the complete evaluation sets, (d) Number of valid responses to the complete evaluation sets.*

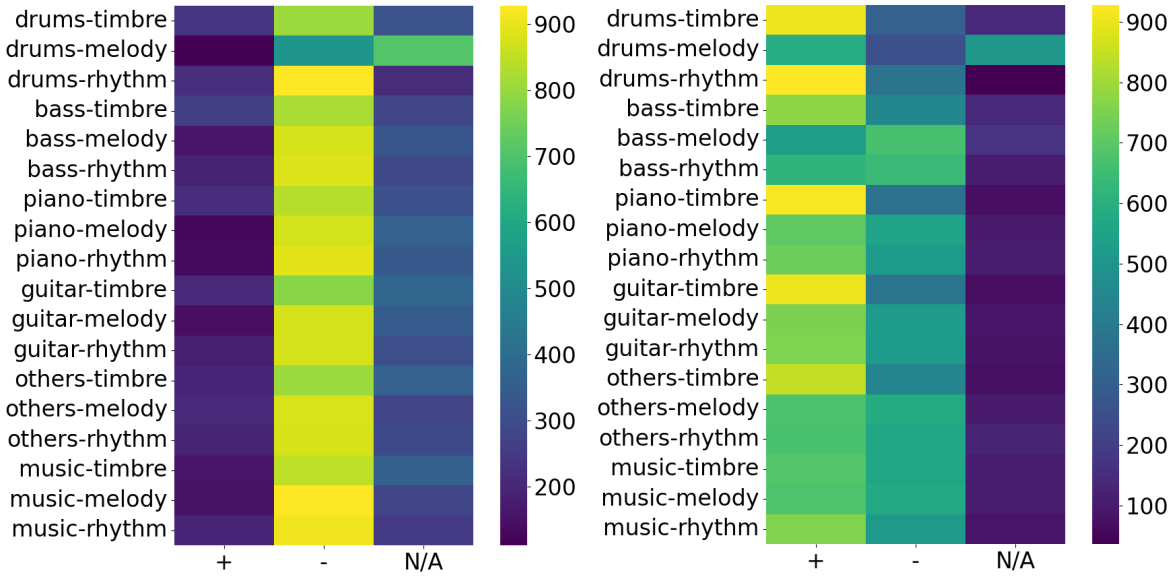
timbre, rhythm, melody, and *overall* as a single response, we obtained 26,898 valid responses from a total of 632 participants (281 unique participants). For all 60 evaluation sets, valid responses from at least six different participants were obtained. Note that this count also includes responses given only for instrumental parts. This is because the listening test was conducted in two phases: in the first phase, only evaluations of individual instrumental parts were collected to obtain instrument labels, namely, responses to 42 sample sets (5 categories × 4 repetitions × 2 types, plus 2 dummy sets); in the second phase, participants responded to *complete evaluation sets*, which consisted of 50 sample sets including the original music tracks. The total number of valid participants who responded to the complete evaluation sets was 346 (167 unique participants); the total number of valid responses to the complete evaluation sets was 16317. Figure 5.3 shows a histogram of the number of participants providing valid responses and the number of valid responses to each evaluation set.

In addition, the number of strong responses (A+ or B+), the number of weak responses (A- or B-), and the number of N/A responses are shown in Figure 5.4. In the case of drums, there were many N/A responses for melody. This is assumed to be because many participants evaluated that "drums have no melody," which was given as an example in the instructions.

5.3 Analysis

5.3.1 Differences across Instrumental Parts

As in Figure 5.5, when a single participant evaluated the similarity of triplets composing the same music track for each instrumental part, the relative similarity relationship, i.e., which X–A or X–B is perceived as more similar, can vary across parts. To examine



(a) $\chi\alpha\beta$ (b) $\chi\chi'\gamma$

Figure 5.4: *Heatmaps showing the number of strong responses (A+ or B+), the number of weak responses (A- or B-), and the number of N/A responses for each instrument and each perspective. Note that only the responses to the complete evaluation set are shown here to enable comparison between the original music tracks and each instrumental part.*

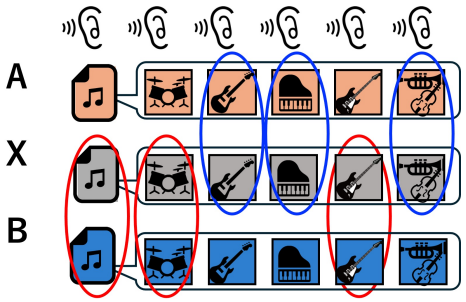
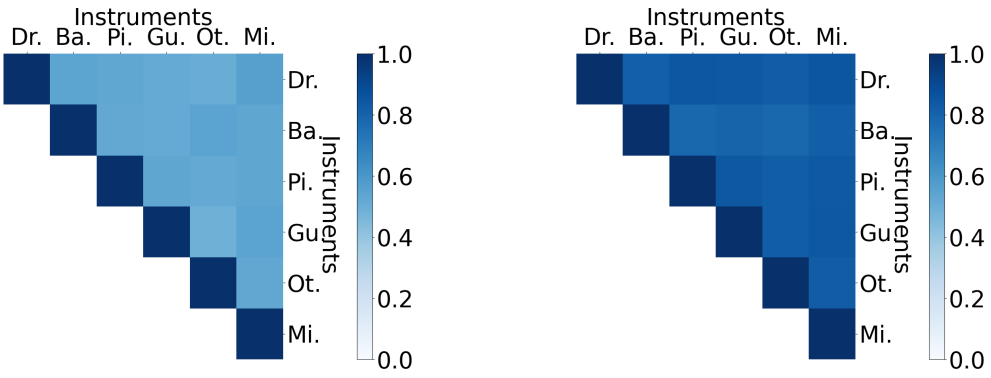


Figure 5.5: *Overview of the analysis in Section 5.3.1. Highlight the pair (i.e., X-A or X-B) that a participant perceives as more similar when they listen to the instrumental part sets that compose the same music triplet. In this example, when focusing on the drum part, B is more similar to X than A is. Additionally, the similarity judgments for the drum part, guitar part, and track level are considered to match.*

this, we assigned a value of 1 if the relative relationship matched between a pair of instrumental parts and 0 otherwise. The agreement rate was then computed as the average of these values across all participants and all responses.

The results are shown in Figure 5.6. In $\chi\chi'\gamma$, the matching rates between instruments are high because many participants chose different segments of the same music tracks as X regardless of the type of instrument. In $\chi\alpha\beta$, the values are comparable among all instruments, i.e., all of the off-diagonal values are around 0.5. In other words, there is no instrumental part pair that consistently exhibits the same tendency across all music triplets and all participants. This indicates that the criteria for music similarity change depending on the instrumental part focused on. It is also suggested that no specific instrument has a consistent and significant influence on the similarity perception of music tracks (mixed sounds). These results suggest that it is reasonable to compute similarity separately for each part. A more detailed analysis of this point is provided in the next section.



(a) $\chi\alpha\beta$ *(b)* $\chi\chi'\gamma$

Figure 5.6: *Heatmaps of the average matching rates of evaluations for each instrument pair. "Dr.", "Ba.", "Pi.", "Gu.", "Ot." and "Mi." represent drums, bass, piano, guitar, others, and mixed sound (i.e., music track) respectively.*

5.3.2 Part- vs. Track-Level Correspondence

In this section, we examine which instrumental parts predominantly contribute to the determination of the track-level music similarity, as in Figure 5.7. (Here, we focus only on $\chi\alpha\beta$.) As shown in Section 5.3.1, the perceived similarity between music tracks varies depending on the instrumental part being focused on. This result raises the question of which instrumental parts listeners prioritize when determining the track-level similarity. For example, as in Figure 5.7, if a participant perceives that music track A is more similar to X for the bass, piano, and other parts, but music track B is more similar to X for the drums and guitar, and evaluated that music track B is more similar to X for the original music track (i.e., mixed sounds), then we can consider that the drums and guitar are dominant in that music triplet for that participant.

We calculated the matching rates with the track-level evaluation for each instrumental part evaluation, and Figure 5.8 shows the results for each music triplet and for

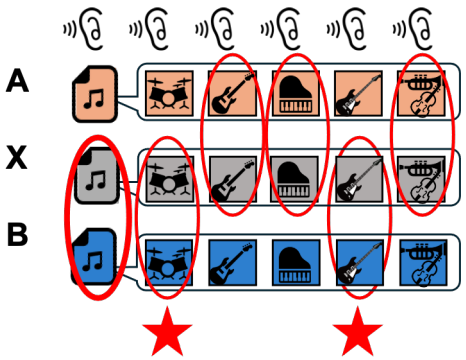
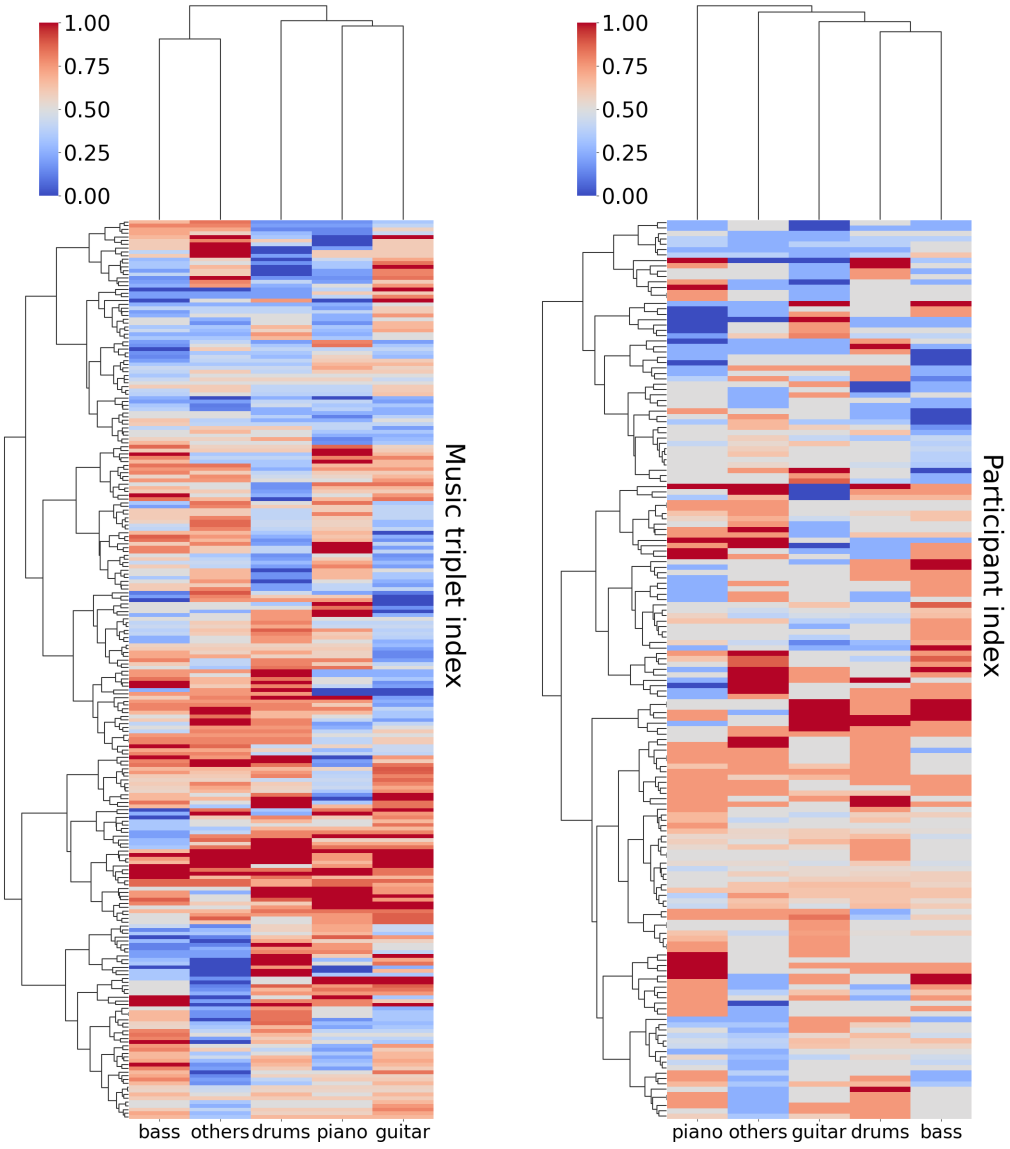


Figure 5.7: *Overview of the analysis in Section 5.3.2. Highlight the pair that a participant perceives as more similar when they listen to the instrumental part sets that compose the same music triplet. In this example, the drums and guitar are considered dominant because their evaluations are consistent with the track-level similarity.*

each participant, along with dendrograms obtained using Ward's hierarchical clustering [113]. In Figure 5.8(a), light colors indicate that participants were divided in their evaluation of the dominant part. Dark colors represent a high level of agreement among participants: the red indicates that the participants perceived the part as dominant, whereas the blue indicates that they perceived it as not dominant. In Figure 5.8(b), light colors indicate that the participant's evaluation changes depending on the music triplet. Dark colors indicate a high level of consistency in a participant's evaluations across different music triplets: the red indicates that the participant consistently perceived the part as dominant, whereas the blue indicates that the part was consistently perceived as not dominant. It can be seen that the dominant instrumental parts vary across both music triplets and participants. Clusters that corresponded to patterns associated with particular combinations of instruments were formed. For example, in (a), when the dendrogram is cut to yield four clusters, the bottom cluster represents a group of music triplets where the *other* part is consistently disregarded among participants.

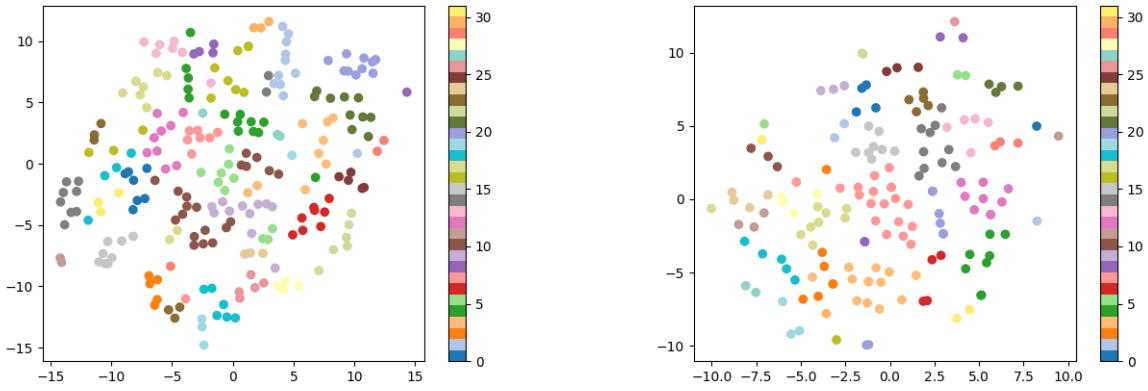
Furthermore, this cluster is divided at the next branch into two subclusters: one in which the bass part is regarded, and another in which it is disregarded. We also present a two-dimensional plot obtained by applying t-SNE for dimensionality reduction, with colors indicating clusters identified using Ward's method in Figure 5.9. These results suggest that the instrumental parts that listeners focus on can be grouped according to either the music triplet or the listener.

Additionally, comparing the two heatmaps in Figure 5.8, (a): based on music triplets and (b): based on participants, we can see that (a) contains more darker-colored cells. This suggests that the evaluations of the same music triplet show high agreement across participants in terms of whether the instrumental part is perceived as dominant. To quantify this, we tested whether there is a significant difference between the variability of evaluations across different participants for the same music triplet and the variability of evaluations across different music triplets from the same participant. Specifically, for each instrumental part, we first construct data in which evaluations were coded as 1 if the track-level and part-level evaluations matched, and 0 if they did not. Then, we examined whether the mean of the variances of the data within each music triplet was smaller than the mean of the variances of the data within each participant, using the Mann–Whitney U test [114]. The results are summarized in Table 5.1. For all instruments, the within-triplet variance was lower than the within-tparticipant variance on average, and the difference was statistically significant ($p < 0.01$) for all instruments except the piano. In other words, the tendency for different listeners to perceive similarly to a given music triplet is stronger than the tendency for the same listener to consistently focus on a particular instrument. This result implies that which instrumental parts are perceived as dominant is more likely to depend on music triplets than on individuals. Taken together with all of this section's results, certain instrumental



(a) *For each music triplet* (b) *For each participant*

Figure 5.8: *Heatmaps showing the matching rates between track-level evaluations and the evaluations for each instrumental part, represented as a 5-dimensional vector. (a) shows the matching rates calculated across all participants who responded to one music triplet, and (b) shows the matching rates calculated across all music triplets responded to by one participant. In this figure, the vectors are reordered based on hierarchical clustering using Ward's method [113], with dendrograms displayed on the left and top sides. Horizontally, clustering is performed by instrumental parts. Vertically, clustering is performed by music triplets in (a) and by participants in (b).*



(a) *For each music triplet* (b) *For each participant*

Figure 5.9: *Two-dimensional plots of the five-dimensional vectors shown in Figure 5.8 reduced using t-SNE. The colors indicate the 32 clusters obtained by cutting the dendrogram from Ward's hierarchical clustering, as presented in Figure 5.8.*

parts have a strong influence on music similarity, and while there are individual differences, there is a considerable level of agreement across multiple listeners. Therefore, estimating which instruments are dominant and leveraging this information in track-level similarity estimation is expected to improve the accuracy of general track-level similarity measures.

5.3.3 Impact of Each Perspective

As explained in Section 5.2, participants evaluated music similarity based on four perspectives; timbre, rhythm, melody, and *overall* for one sample set. The participants must choose either A or B in the *overall*, even when the evaluation is divided among the three perspectives. By examining which perspective the participants followed in selecting the *overall*, we investigated which perspective is important for humans to

Table 5.1: *The mean of the variances of the data within each participant (averaged across participants) and the mean of the variances of the data within each music triplet (averaged across music triplets), where the data represent binary values (1 or 0) indicating whether the evaluations matched the track-level evaluations. The p-values for testing whether the former is smaller than the latter are also shown. Since the number of evaluations per music triplet and per participant differed in range, we filtered the data to include only music triplets and participants with between 4 and 9 evaluations, where the ranges overlapped. After this filtering, the average number of evaluations per music triplet was 5.74, and the average number of evaluations per participant was 4.68. We calculated the means after the filtering.*

	drums	bass	piano	guitar	others
mean variance in a triplet	0.21496	0.22861	0.22288	0.22473	0.22137
mean variance in a participant	0.25918	0.24173	0.23684	0.25223	0.24672
p-value	0.00000	0.00906	0.09272	0.00001	0.00170

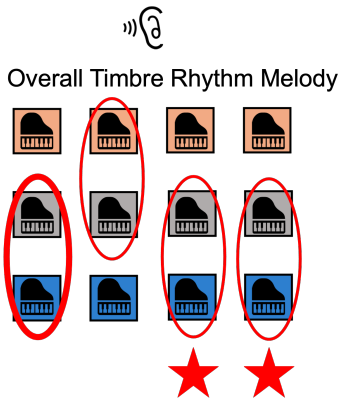


Figure 5.10: *Overview of the analysis in Section 5.3.3. Highlight the pair that a participant perceives as more similar for each perspective. In this example, the rhythm and melody are considered important because their evaluations are consistent with the overall similarity.*

listen to for each instrumental sound.

Matching rates of three perspectives with *overall* are shown in Table 5.2. In drums, many of the responses for melody are N/A, and the matching rate with *overall* for melody is low. For all instruments, the matching rate of timbre is as high as 70% or more, but that of rhythm and melody are even higher than that of timbre except for drums. This means that rhythm and melody tend to have a larger impact on the perceptual music similarity than timbre in each instrumental sound, except for melody in drums.

5.3.4 Correspondence with Deep Features

Models for Evaluation

Our Developed Model

We developed an instrumental-part-level similarity learning method with a single net-

Table 5.2: *Matching rate with overall for each perspective (%)*

	drums	bass	piano	guitar	others	mix
timbre	74.7	74.5	75.6	75.1	75.3	74.3
rythm	**84.4**	**80.1**	78.4	**80.6**	79.8	**82.0**
melody	56.7	79.0	**80.7**	**81.0**	**83.1**	**82.8**

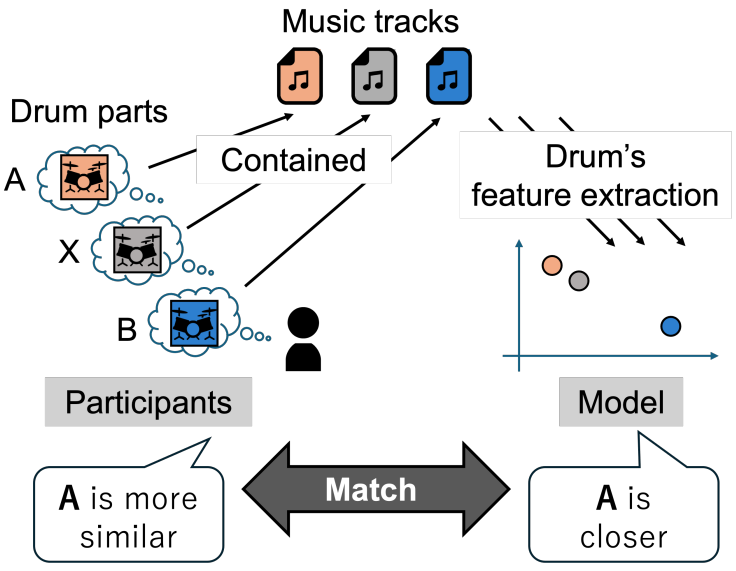


Figure 5.11: *How to evaluate the models.*

work that takes original music tracks (in Chapter 4). We designed a similarity embedding space with separated subspaces for each instrumental part using Conditional Similarity Networks (CSNs) [26]. To separate the embedding space, a mask is applied to all dimensions except for the dimension corresponding to the notion to be considered in the triplet loss calculation. That is, for example, when learning the drum space, a binary mask that retains only the dimensions assigned to drums is applied, and the distance between the masked feature representations is used for metric learning with the triplet loss. When learning the drum space, it is necessary to sample data based on (dis)similarity, focusing on the drum part. To address this, we introduced *pseudo musical piece*, which extends unsupervised learning under the assumption that segments within the same track are similar. The model input was a mel spectrogram converted from the original music track signal, and the output was a 640-dimensional vector in which five instruments (drums, bass, piano, guitar, and others) were each assigned a 128-dimensional subspace. The model was trained using the training data from the Slakh2100 dataset [102]. The network consisted of ten convolutional layers, followed by temporal average pooling and a fully connected layer.

MERT [59] with Temporally Average Poolings

We also conducted a performance evaluation using MERT [59], a pretrained model based on self-supervised learning that can extract musical features. Unlike our model, MERT does not have a mechanism for extracting the distinct features of each instrumental part from original music tracks, so we input the instrumental part signals instead. We downsampled the signals to 24 kHz, extracted 3D tensors with 25 layers × T time steps × 1024 feature dimensions from the intermediate layers of the pre-trained model [2]. We then applied average pooling along the layer and time axes

[2]https://huggingface.co/m-a-p/MERT-v1-330M

to obtain 1024-dimensional representations. Noted that in this type of framework, where individual instrumental part signals are used as inputs, it is generally necessary to use estimated signals obtained from source separation of the original music track, as in our model, because true individual instrumental part signals are typically not available at inference. However, in this experiment, to eliminate the influence of separation performance and simplify the discussion, we used estimated signals at their ideal upper-bound performance, i.e., stems, as inputs.

Performance Evaluation Proceedure

We calculated whether A or B was closer to X using the models for the same set as used in this listening test and then calculated the matching rate between the models' results and the subjective evaluation results, as shown in Figure 5.11 (same as the subjective evaluation experiment in Chapter 4). The calculation of distances using the model was performed as follows: The music tracks originally containing the instrumental parts used in the listening test were input into the model, the instrumental-part level feature representations were extracted from them, and then the distances were measured between them. Sample sets with low agreement among participants indicate that both A and B are equally (dis)similar to X, making them unsuitable for this performance evaluation method. Therefore, only evaluations with agreement equal to or higher than 80% among participants were used in the evaluation. This filtering focuses on reliable perceptual judgments by emphasizing high-consensus cases, while trading off broader sample coverage. To avoid excessively reducing the number of evaluation samples, we set the consensus threshold to 80% in this study, balancing reliability and coverage. Note that if N/A accounts for the largest percentage, that sample set was excluded from the evaluation set.

The numbers of unique sample sets and evaluations for them used for this performance evaluation are shown in Table 5.3. We can see that there is more agreement among participants in their evaluations in $\chi\chi'\gamma$ than in $\chi\alpha\beta$. Moreover, there are fewer sample sets for timbre, rhythm, and melody compared with *overall* because the participants cannot select N/A in *overall*, but can in the others, and here sample sets that received N/A from 80% or more of participants were omitted from the evaluation.

Results

The performance evaluation results of our model are shown in Figure 5.12. Although $\chi\alpha\beta$ is less accurate than $\chi\chi'\gamma$, accuracy is higher in drums, piano, and guitar in the performance evaluation using evaluations focusing on timbre compared with using those focusing on *overall*. In guitar, accuracy is higher even in the melody from *overall*. In all instruments, rhythm tends to show an equal or lower accuracy than the other perspectives. This suggests that frequency-related features are captured by the model, but rhythmic features are not. This is considered to be due to the network architecture, which contains the temporal average pooling.

Note that although the drum melody also shows a relatively high value, this result is based on a significantly smaller number of samples than for other results. Furthermore, we consider that this evaluation is based on the perceived similarity of drum patterns. The number of evaluations for the melody of drums after filtering is significantly smaller than that for the others because many participants selected N/A for the melody of the drums, as illustrated in Figure 5.4. Among the 281 unique participants, 39 selected N/A for all evaluations to the melody of drums. In other words, 242 participants provided at least one non-N/A evaluation for drum melody across the sample sets. Furthermore, there were no sample sets for which all participants responded with N/A. However, even

Table 5.3: *Number of unique sample sets and number of evaluations used for performance evaluation. Only sample sets with an agreement rate of 80% or higher among participants and evaluations for them were included.*

(a) *Number of unique sample sets of* $\chi\alpha\beta$

	drums	bass	piano	guitar	others
overall	92	95	83	90	95
timbre	49	44	37	42	43
rhythm	56	47	26	33	38
melody	4	41	21	25	47

(b) *Number of unique sample sets of* $\chi\chi'\gamma$

	drums	bass	piano	guitar	others
overall	212	180	210	209	192
timbre	212	192	214	206	197
rhythm	201	164	192	192	185
melody	211	177	203	200	188

(c) *Number of evaluations of* $\chi\alpha\beta$

	drums	bass	piano	guitar	others
overall	912	949	836	925	977
timbre	463	420	330	412	414
rhythm	517	433	233	317	355
melody	32	363	186	238	437

(d) *Number of evaluations of* $\chi\chi'\gamma$

	drums	bass	piano	guitar	others
overall	2143	1792	2082	2096	1926
timbre	1926	1745	2044	1993	1892
rhythm	1983	1555	1800	1819	1721
melody	1391	1589	1905	1898	1786

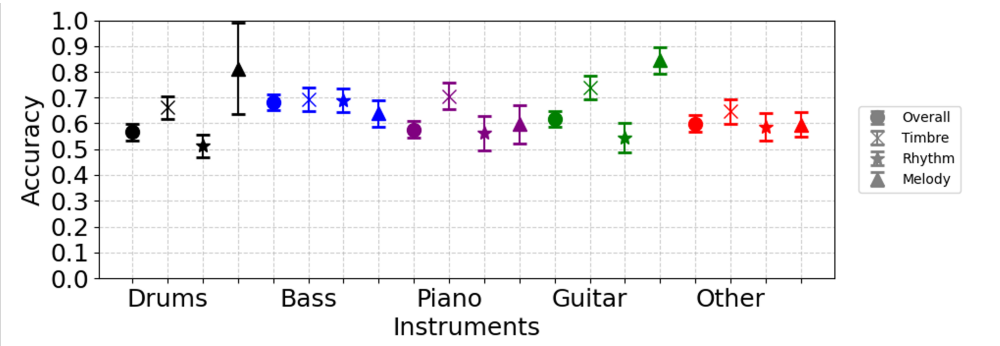
so, the number of sample sets where more than or equal to 80% of participants agreed on a non-NA evaluation was only four in $\chi\alpha\beta$ (shown in Table 5.3). In other words, there were only a few sample sets in which participants exhibited higher sensitivity to differences in melody. In these sample sets, one of the two (A or B) had a drum pattern that was highly similar to X and easily distinguishable pitch contrasts, such as repeated alternations between bass and snare drums, whereas the other had a different drum pattern from X. We consider that these differences were perceived as melodic differences.

The performance evaluation results of MERT with average poolings are shown in Figure 5.13. Note that, as mentioned in Section 5.3.4, since this performance was based on inputting ideal instrumental part signals (stems), a direct comparison between this result and that of our previous work is not possible. On the other hand, similar trends are observed between the model in our previous work and MERT with temporal and channel-wise average pooling, namely that performance on rhythm is lower than on the other perspectives. These findings suggest that temporal average pooling in deep learning models leads to a loss of rhythm information in any instrumental part.

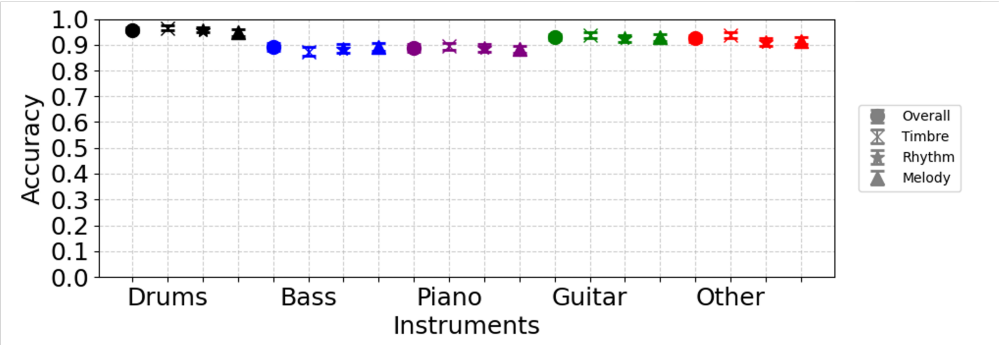
In conjunction with the finding in our previous work that listeners placed more importance on rhythm or melody than on timbre (Table 5.2), we consider that if we can design a model that captures the structure of the time direction so that rhythm can be considered, it will be possible to obtain a music similarity that is also compatible with listeners' perception on *overall*.

5.3.5 Within- and Between-Piece Similarity

In this section, we examine whether the similarity between temporally distinct segments within the same music track is perceived to be higher than that between different

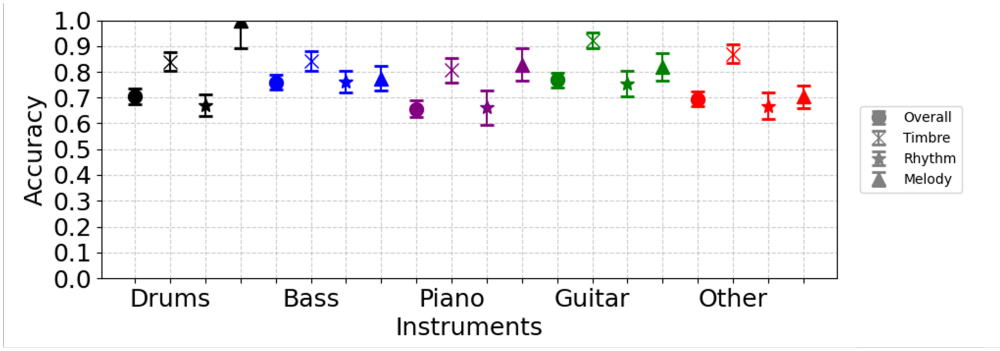


(a) $\chi\alpha\beta$

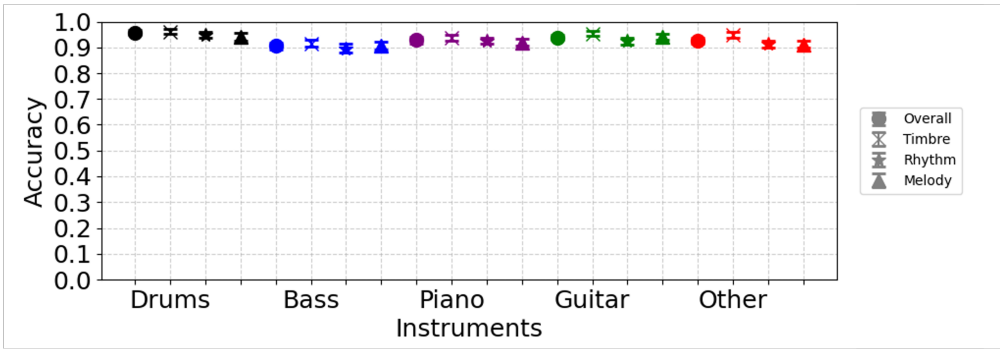


(b) $\chi\chi'\gamma$

Figure 5.12: *Graphs of accuracy and the 95% confidence intervals of our model for each instrument and each perspective. The 95% confidence intervals were calculated using the Clopper–Pearson method [115]. Colors represent instruments, and symbols represent perspectives.*



(a) $\chi\alpha\beta$



(b) $\chi\chi'\gamma$

Figure 5.13: *Graphs of accuracy and the 95% confidence intervals [115] of MERT for each instrument and each perspective. Colors represent instruments, and symbols represent perspectives.*

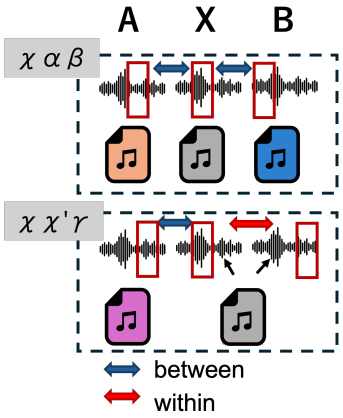


Figure 5.14: *Explanation of between-music similarity and within-music similarity. In* $\chi\chi'\gamma$*, one sample is a different segment of the same music track as X. Thus, one of the similarities to evaluate is the within-music similarity.*

music tracks. We employed the unsupervised learning method with triplet loss, assuming that the similarity between temporally distinct segments within the same music track is higher than that between different music tracks in our study (described in Chapter 3 and 4). As shown in Figure 5.14, the participants compare two between-music similarities in $\chi\alpha\beta$, whereas they compare the within-music similarity with the between-music similarity in $\chi\chi'\gamma$. The data selection setting in $\chi\chi'\gamma$ is the same as our unsupervised learning method. Table 5.4 shows the rate at which the segment from the same music track as X, namely, χ', was selected in $\chi\chi'\gamma$. The 95% confidence intervals were calculated using the Clopper–Pearson method [115]. Considering all evaluations, including N/A, the percentages of evaluations exceed 70% for all except the melody of drums. When N/A evaluations are excluded, the percentages of evaluations exceed 80% for all, including the melody of drums. The results show that the similarity between randomly extracted segments within the same music track is perceived to be significantly higher than that between different music tracks. This suggests that our

Table 5.4: *Percentage of evaluations selecting the segment from the same music track (χ') in $\chi\chi'\gamma$, with the 95% confidence intervals[%]*

(a) *Percentage of χ' among all evaluations including N/A*

	drums	bass	piano	guitar	others	music
overall	91.49–93.62	83.91–86.77	89.02–91.43	89.0–91.41	86.48–89.13	87.82–91.15
timbre	82.4–85.37	78.16–81.39	86.24–88.91	84.8–87.59	83.12–86.04	79.57–83.76
rhythm	86.94–89.55	72.87–76.37	79.64–82.79	80.71–83.8	76.53–79.87	81.73–85.73
melody	57.89–61.85	73.33–76.81	82.37–85.34	82.04–85.04	79.05–82.24	81.96–85.94

(b) *Percentage of χ' among evaluations excluding N/A*

	drums	bass	piano	guitar	others	music
timbre	92.17–94.32	86.8–89.55	91.63–93.8	89.53–91.94	88.6–91.12	86.57–90.19
rhythm	89.9–92.25	79.61–82.90	86.69–89.42	86.17–88.93	84.24–87.21	87.38–90.87
melody	90.63–93.37	83.13–86.22	88.7–91.22	88.14–90.72	85.05–87.91	89.3–92.55

previous method is appropriate for realizing the goal of learning similarity that aligns with human perception. Furthermore, we can see from Figure 5.4 that the number of participants who perceived a strong similarity is higher for $\chi\chi'\gamma$ than for $\chi\alpha\beta$. These indicate that within-music similarity is perceived to be significantly higher than between-music similarity.

5.4 Conclusions and Discussions

In this chapter, we investigated how part-level perceptual similarity is perceived and related to the track-level similarity. We conducted a large-scale listening test and obtained 26,898 responses from 632 participants from four perspectives, i.e., timbre,

rhythm, melody, and *overall*. We conducted an analysis and obtained the following insights. (1) The relative similarity relationships among triplet tracks differ depending on the instrumental part being focused on. This suggests that it is reasonable to compute similarity separately for each part, as listeners have different criteria for music similarity perception across instrumental parts. (2) In relation to (1), some parts show consistency with the track-level similarity, while others do not. The dominant instrumental part in the music track, which is consistency with the track-level perception, varies across music triplets and listeners. Moreover, the variance across participants for the same music triplet was smaller than the variance within a participant across different triplets, suggesting that differences among music triplets have a stronger influence than individual differences among participants. This indicates that certain instrumental parts tend to be more salient for listeners' perception depending on the music track. (3) Rhythm and melody tend to have a larger impact on perceptual music similarity for each instrumental part than timbre. (4) However, the deep learning models with temporal average pooling capture the features related to timbre compared to rhythm. This suggests that temporal average pooling leads to the loss of rhythmic information. (5) The similarity between temporally distinct segments within the same music track is perceived to be significantly higher than that between segments from different tracks, supporting the unsupervised learning approach used. These findings provide important insights for future research on part-level similarity.

6 Application: Part-Level Music Retrieval Interface

Finally, we consider how the part-level similarity can be applied. To this end, we propose a retrieval interface, *MixQuery*, that effectively leverages part-level similarity. Section 6.1 provides an introduction, Section 6.2 describes the system, Section 6.3 explains the implementation, and Section 6.4 presents the subjective evaluation experiments.

6.1 Introduction

Given the complexity of popular music, as discussed in Chapter 1, flexible music retrieval systems are required to find desired songs from a large collection. Most content-based music retrieval systems compute similarity based on audio features by analyzing the entire audio signal [3, 27, 28], without allowing users to adjust what musical aspects to prioritize. However, the musical aspects that listeners care about can differ depending on preferences and contexts. Therefore, various studies have proposed methods for estimating musical similarity from different perspectives. Some focus on audio-based aspects such as vocal timbre, musical timbre, rhythm, and chord progression [116]. Others incorporate non-audio information, including text-based features (*e.g.*, tags [117] and lyrics [92, 93, 118]), co-occurrence in blogs or playlists, and user evaluations [119]. These approaches have enriched the understanding of music simi-

larity by considering diverse modalities. Nevertheless, no studies have addressed the problem of interactively exploring similarity based on multiple instrumental parts (*e.g.*, vocals, drums, bass) while incorporating users' part-level preferences into a musical retrieval system.

To overcome this limitation, we propose *MixQuery*, a music retrieval system that allows users to create personalized queries by selecting some of four instrumental parts (separated stems of vocals, bass, drums, and other) based on their preferences. MixQuery enables a novel type of retrieval in which a user's track-level preference for a song is broken down into part-level preferences and then reconstructed. As illustrated in Figure 6.1, MixQuery guides users through an interactive workflow. The users first select a favorite song from a song list (Step ①), listen to its source-separated stems, and then choose a temporal segment where the users find the sound of the instrumental part appealing. This selected segment, referred to as a *stem query*, is added to a query set (Step ②-1). The users can continue adding multiple stem queries from other favorite songs and can also browse and add similar stem queries suggested by the system (Step ②-2). Finally, the system retrieves songs by aggregating the similarities between each stem query in the query set and the corresponding instrumental part of every song in a large music collection. With MixQuery, for example, users can discover songs where the vocal part resembles one song while the drum part resembles another. By combining interactive stem selection with instrumental part-level similarity aggregation, MixQuery enables users to explore a large music collection in a flexible and personalized manner.

To assess the effectiveness of MixQuery, we conduct a subjective evaluation to verify whether the retrieved songs reflected the intended similarity to the selected stem queries. We designe an experiment comparing top-ranked retrieval results with lower-

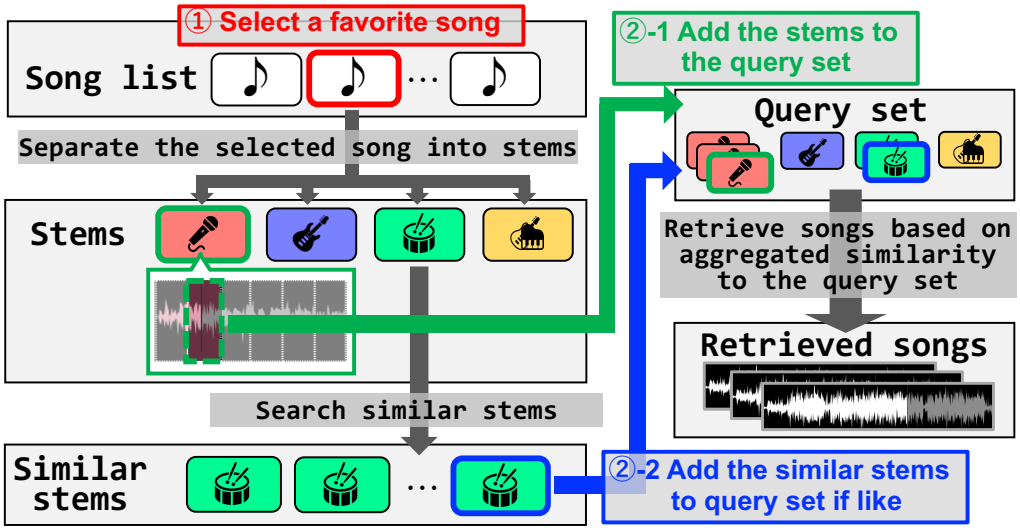


Figure 6.1: *Interactive workflow of MixQuery.*

ranked ones, using the query sets containing two different types of instrumental parts.

6.2 System

MixQuery is a music retrieval system in which users can create personalized queries by selecting and combining multiple instrumental parts from various favorite songs. By iteratively selecting preferred temporal segments as stem queries and aggregating them, the system allows users to find songs that match their part-level preferences.

The user interface of MixQuery is shown in Figure 6.2. It consists of four main steps:

Step ① The users select a song from the song list.

Step ② The users listen to four stems of the selected song, select a preferred temporal segment as a stem query, and optionally search for similar stem queries.

Step ③ The selected segments are added to the query set.

Step ④ Based on the query set, the system displays songs retrieved according to the aggregated similarities.

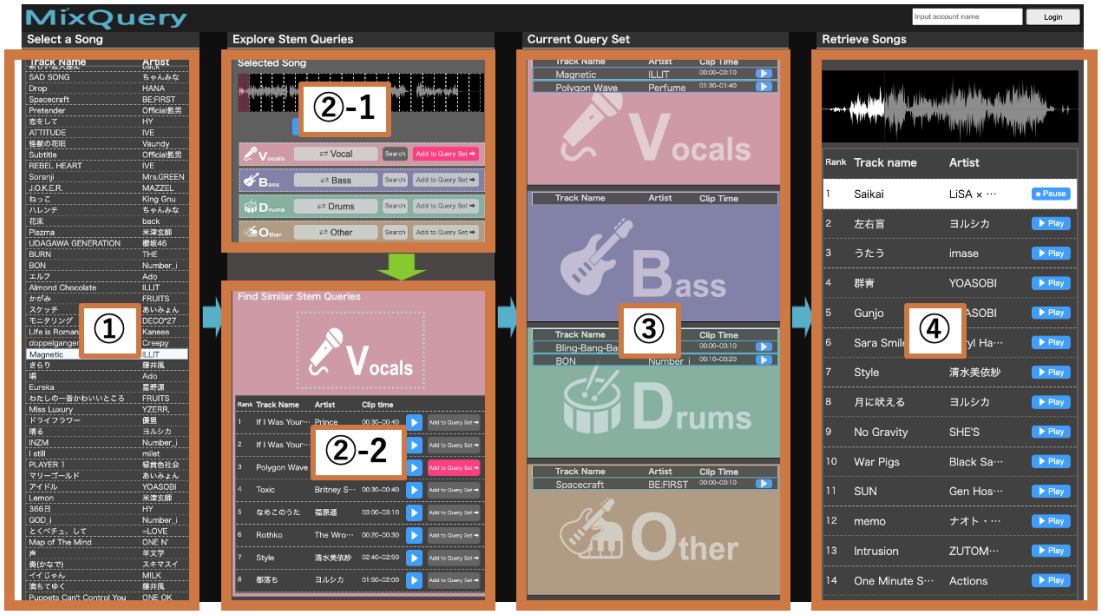


Figure 6.2: *The user interface of MixQuery consists of four interactive areas: song selection (①), stem query exploration (②), query set construction (③), and retrieval results (④).*

The users can repeat the process of selecting songs and stems, and the retrieval results are updated dynamically. Details of each step are described below.

6.2.1 Explore a Favorite Segment of a Song

After selecting a favorite song from the song list (Step ①), the users explore a particularly preferred stem and its specific time segment within the song (Step ②-1). Since the four stems the users can select are vocals, drums, bass, and other, the term *other* refers to all sounds except vocals, drums, and bass, such as guitars and pianos.

As shown in Figure 6.3, when the users play back the song as well as its separated stems and find a specific segment with a preferred sound to use as a stem query, they can add it to the query set (Step ③) by clicking the "Add to Query Set" button.

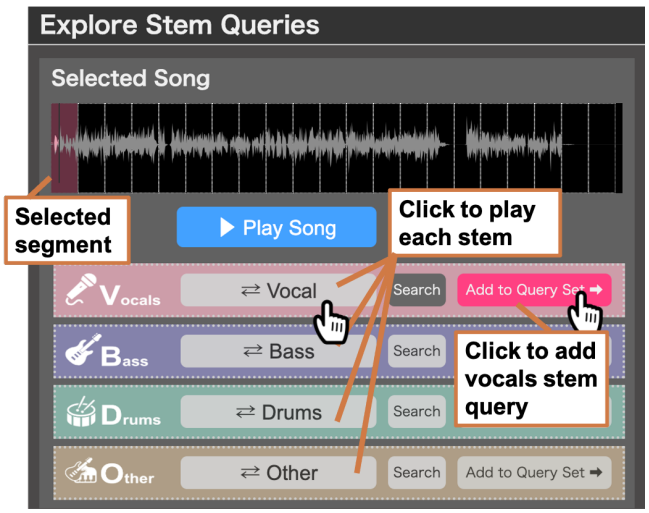


Figure 6.3: *Step ②-1 in Figure 6.2. The users can explore stem queries. In this example, the first segment of vocals was selected. The color of the "Add to Query Set" button turned pink as this vocal segment was added to the query set.*

Each segment is a 10-second portion obtained by sequentially dividing the song from its beginning. Since instrumental performance or singing style can change throughout the song, this allows the users to select the segment with their favorite sound for each instrument, providing a key advantage in query construction.

6.2.2 Find Similar Stem Queries

As an optional feature, the stem segment selected in Step ②-1 can be used to search for similar stem queries from other songs (Step ②-2). The system identifies the 100 most similar 10-second stem segments to the selected segment (in Step ②-1), all belonging to the same instrumental part. These are displayed as in Figure 6.4. The users can listen to these segments interactively and add any that they find appealing to the query set as a stem query. This allows the users to build the query set not only from their preferred songs selected in Step ②-1 but also from songs they were

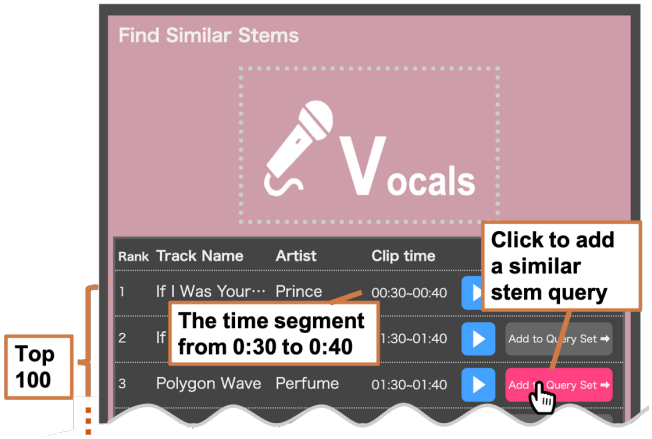


Figure 6.4: *Step ②-2 in Figure 6.2. The users can find similar stem queries. The top 100 similar segments are displayed with their song name, artist name, and time. In this example, vocals stem queries similar to the one selected in Figure 6.3 are presented, and the third one is added to the query set.*

previously unaware of or unfamiliar with. In other words, the system helps the users refine their part-level preferences by identifying and suggesting stem queries similar to their selected segment.

6.2.3 Aggregate Stem Queries in the Query Set

In Step ③, the system aggregates the stem queries added in Steps ②-1 and ②-2 to retrieve songs. Whenever a new stem query is added to the query set, the retrieval results in Step ④ are updated automatically. This process runs in the background without user interaction. The current query set is always displayed in Step ③, and the retrieval results are generated based on it, as shown in Figure 6.5.

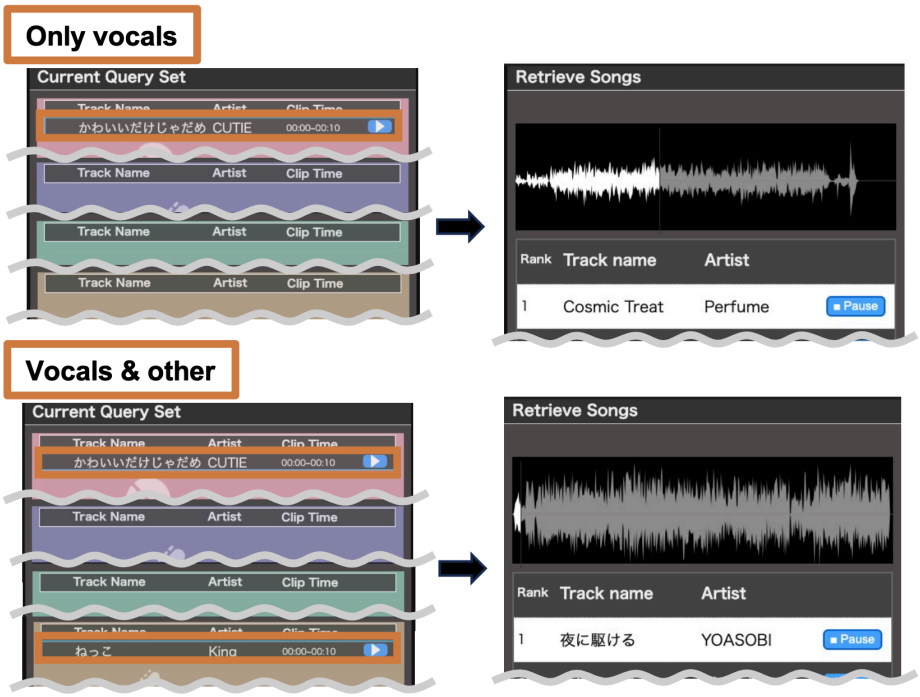


Figure 6.5: *Examples of the query set (Step ③) and the retrieval results (Step ④). In the upper example, when only the vocal of a female group was added to the query set, a female group's dance music-style song was shown as the top result. Next, after the other from another song, which includes piano accompaniment, was added to the query set as in the lower example, the top result was changed into a song with female vocals and piano accompaniment.*

6.2.4 Enjoy Listening to Retrieved Songs

The system displays the top 100 songs (Step ④) retrieved based on the current query set (Step ③). A key feature of the system is that retrieved songs can be played back starting from the point most similar to the aggregated query.

6.3 Implementation

This section describes the implementation of the proposed system and the data used for evaluation.

6.3.1 Dataset

Two datasets were prepared: one for user selection, and the other for similar stem queries search as well as retrieval display.

The first dataset was created for user selection, containing a total of 85 songs, which were used as the song list displayed in Step ①. Although MixQuery is designed to allow users to freely select their favorite songs in practical use in the future, in this study, we curated a fixed set of songs to verify its effectiveness. We selected recent popular songs from the top 100 songs on the Billboard Japan daily chart [1] (as of March 3, 2025) so that users can be familiar with many of them. Billboard charts are commonly used in the analysis of popular music [120, 121]. To minimize artist bias, particularly in vocal characteristics, we limited each artist to a maximum of five songs, keeping only the highest-ranked ones. After this filtering, we obtained audio for 85 songs available on YouTube.

The second dataset was created for song retrieval, with a total of 3,108 songs. These songs served as target songs for both the similar stem query search in Step ② and the retrieval results displayed in Step ④. Each song was divided into 10-second segments from the beginning for use in the system. It is important to include as many target songs as possible in the retrieval dataset. If users are dissatisfied with the retrieval results, it can be unclear whether the issue is system accuracy or the absence of desired songs.

[1] https://www.billboard-japan.com/charts/detail?a=hot100

To mitigate this, we built a large dataset using public research datasets, which consists of 100 songs from the Popular Music subset of the RWC Music Database (RWC-MDB-P) [122], 150 songs from MUSDB18 [123], 1,911 songs listed in the SecondHandSongs (SHS) Dataset [124] (only consisting of original songs available on YouTube), and 947 popular Japanese songs obtained from publicly available three YouTube playlists based on play count rankings, 2024 hits, and 1990s hits (available as of December 2024).

6.3.2 Automatic Sound Source Separation

To obtain stems for each instrumental part, we applied music source separation to all songs in the dataset. We used HT Demucs [125] to separate each song audio into vocals, bass, drums, and other stems. Each separated stem was then divided into 10-second segments for user interaction.

6.3.3 Feature Extraction

In MixQuery, distances are computed between each stem query and the corresponding stems in all target songs, using feature representations extracted from the models as similarity metrics. These distances are used both in Step ②-2 to find similar stem queries and in Step ④ to calculate the similarity between the query set and each target song. Therefore, it is essential to extract meaningful feature representations that capture the acoustic characteristics of individual stems.

To achieve this, we adopted a self-supervised learning framework based on MERT [59], a large pre-trained transformer-based model designed for music understanding. Each 10-second audio segment was downsampled to 24 kHz and passed through the frozen

MERT model[2], from which we extracted 25 transformer-layer features by averaging across the time dimension at each layer, resulting in 25 embeddings of 1024 dimensions. Since MERT is not originally trained to capture similarities between isolated musical parts (*e.g.*, vocals, drums), we trained a separate 1D-convolutional network (with kernel size 1) for each stem type on top of the MERT features to produce a single 1024-dimensional embedding per stem.

To train the 1D-convolutional networks, we used the 3,108 target songs described in Section 6.3.1. The networks were trained separately for each stem type i (vocals, bass, drums, or other) using a triplet loss. Given an anchor segment, a positive sample from a different time in the same song and a negative sample from a different song were used to encourage embeddings of acoustically similar segments to be closer than dissimilar ones, measured by cosine distance. This triplet learning method was the same as used in Chapter 3. It was shown that the method enables learning a similarity metric corresponding to perceptual similarity focused on timbre in Chapter 5. The optimizer used was Adam [126] with a learning rate of 0.0005, and training was performed for 100 epochs.

After training, we used our models to compute feature representations for all 10-second segments of each stem in the dataset. For each stem type i, this resulted in a set of embeddings that were used in both similar stem queries search and retrieval.

6.3.4 Retrieval Function

MixQuery includes two retrieval mechanisms: (1) searching similar stem queries in Step ②, and (2) retrieving songs based on the constructed query set in Step ④.

In Step ②, similar stem queries are retrieved by computing the cosine distance

[2]https://huggingface.co/m-a-p/MERT-v1-330M

between the feature embedding of the input segment with stem type i and those of segments of stem type i in the target songs. Since both segments are of the same stem type, the comparison reflects within-stem similarity.

In Step ④, the system retrieves songs by comparing the set of stem queries (the query set) with each song in the target dataset. This process consists of two stages, as illustrated in Figure 6.6. First, for each stem type i that appears in the query set, a query representation $y^{(i)}$ is computed by averaging the feature embeddings of all added stem queries of type i:

$$y^{(i)} = \frac{1}{N_i} \sum_{n=1}^{N_i} x_n^{(i)} \tag{6.1}$$

where N_i is the number of stem queries of type i included in the query set, and $x_n^{(i)}$ is the n-th feature embedding of stem type i in the query set.

Next, for each song in the target dataset, the cosine distance $d(y^{(i)}, y'^{(i)})$ is computed between each averaged query embedding $y^{(i)}$ and the feature representation $y'^{(i)}$ of a 10-second segment of stem type i in the target song. These distances are then aggregated by averaging across all stem types present in the query set. The final distance D between the query set and a song is defined as:

$$D = \frac{1}{|I|} \sum_{i \in I} d(y^{(i)}, y'^{(i)}), \tag{6.2}$$

where I is the set of stem types included in the query set. This distance reflects how closely a target song aligns with the users' overall preferences, computed by aggregating the part-level preferences specified in the query set. For each target song, distances are calculated over all 10-second segments, and the minimum value is used as the song's final score. Songs are then ranked in ascending order and presented to the users.

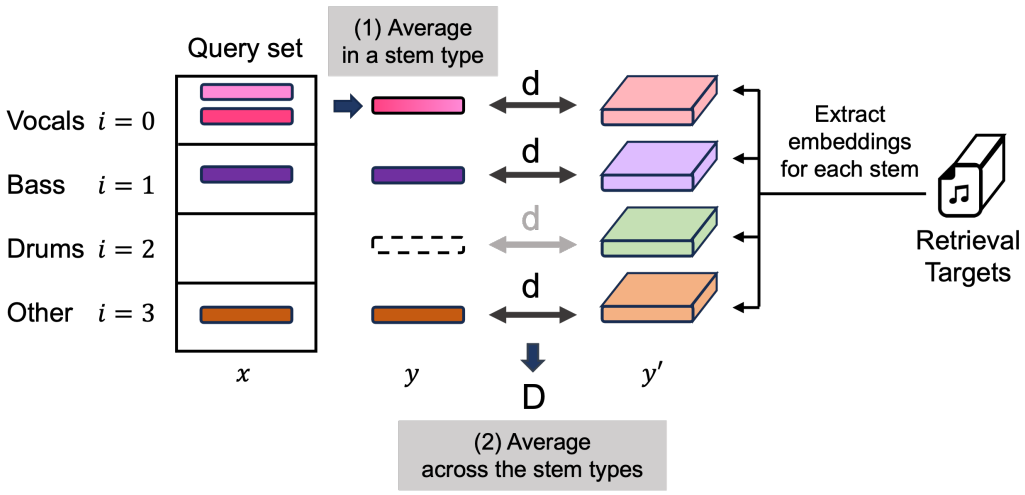
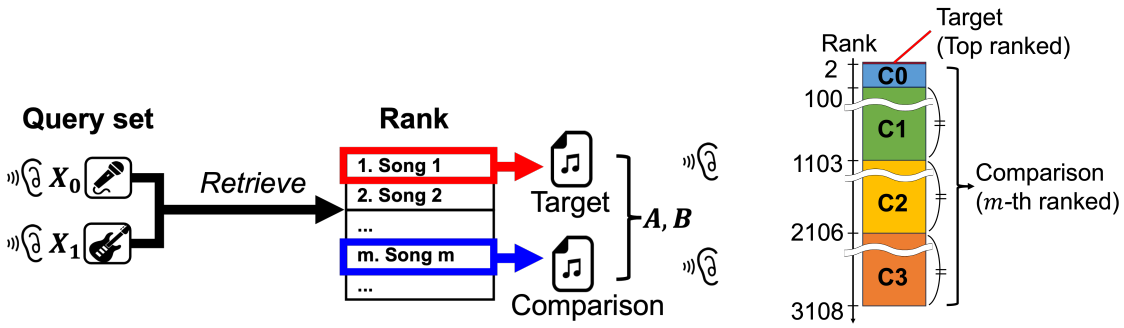


Figure 6.6: *How to calculate the aggregated distance D used for a retrieval function in Step ④. In this example, two distinct vocal segments have been added as stem queries, while one stem query each has been added for bass and other, and none have been added for drums. The vocal stem queries are averaged, and then the average distance D is calculated across the three stems excluding drums.*

6.4 Experimental Evaluation

To evaluate whether our system's similarity metric aligns with human perception, we conducted an ABX-style experiment. Ideally, evaluating this system requires ground-truth labels indicating that, for example, the song most similar to the vocals of song A and the drums of song B. However, such ground-truth data do not exist, making objective evaluation metrics such as MAP@k or MRR, generally used in MIR research [127], inapplicable. Moreover, high performance on objective evaluation does not necessarily imply that the system is useful to humans since the crucial question is whether the system aligns with human perception.

Therefore, we performed a subjective evaluation of the system in this study. The ABX experiment specifically assesses whether top-ranked retrieval results are percep-



(a) *Overview*

(b) *Strategy*

Figure 6.7: *How to select samples for the subjective evaluation experiment. The overview is shown in (a). The types of strategies for sampling comparisons, C0 to C3, are shown in (b).*

tually more similar to the stem queries than lower-ranked results. Fifteen participants (12 male, 3 female; all university or graduate students in their 20s who regularly listen to popular music) took part in the experiment.

6.4.1 Subjective Evaluation Method

Eight distinct query sets were used as X in the ABX experiment, and each query set consisted of two stem segments X_0 and X_1 (*e.g.*, vocals and drums) manually selected from different songs in the first dataset described in Section 6.3.1 (85 songs, used as the song list in Step ①). The stem type combinations for all eight query sets are summarized in Table 6.1. These segments were selected through careful auditory inspection by the author to ensure that each stem displayed distinctive musical characteristics, such as prominent vocal timbres, recognizable drum sounds, or noticeable instrumental textures.

This experiment assessed whether the song ranked 1st by the system was perceived

as more similar to the query set than a lower-ranked result, specifically the m-th ranked song ($m > 1$). Figure 6.7-(a) illustrates the procedure: the query set was used to retrieve a ranked list of songs. Two candidates were then selected for perceptual comparison: the top-ranked result (labeled *target*) and a lower-ranked comparison song (labeled *comparison*). The target and comparison were randomly assigned to A and B, respectively. Participants first listened to the query set (X), which consisted of two 10-second stem segments. Then they listened to 10-second segments of both songs (A and B), the target and the comparison, and chose the one that more closely matched the stems in the query set, focusing on both instrumental parts included in the query set.

To sample a lower-ranked song for the comparison candidate, we employed four sampling strategies: (C0) rank m was randomly selected from 2–100, and (C1)–(C3) from below 100, which were divided into three equal parts: (C1) from 101–1103, (C2) from 1104–2106, and (C3) 2107–3108 (as illustrated in Figure 6.7-(b)). Each query set was assigned one of these four strategies. The comparison strategies for all eight query sets are also summarized in Table 6.1. Every participant answered the same ABX experiment set.

6.4.2 Results

Figure 6.8 shows the subjective evaluation results. We defined an answer as *correct* if the top-ranked song (target) was selected by a participant and *accuracy* as the proportion of correct answers. The 95% confidence intervals were calculated using the Wilson score interval [128].

Conditions C2 and C3 exhibit higher accuracy than C0 and C1. This suggests that the song ranked first is more similar to the query set than the songs ranked well below

Table 6.1: *Stem combinations of the query sets, as well as sampling strategies and actual ranks of the comparisons, for the eight sets in the experiment. "w/" indicates that the corresponding stem is included.*

		0	1	2	3	4	5	6	7
query set	vocals	w/	w/		w/		w/		
	bass					w/	w/	w/	w/
	drums	w/		w/	w/			w/	
	other		w/	w/		w/			w/
comparison	strategy	C0	C0	C1	C1	C2	C2	C3	C3
	rank	6	24	249	762	1936	1944	2482	2616

the middle. Meanwhile, when the comparison candidate is sampled from ranks 2–100 (C0), the average accuracy hovers around 50% (i.e., the chance level). This implies that among the top 100 songs, there is no substantial difference in perceived similarity relative to the top-ranked song. From a practical standpoint, returning around 100 results seems reasonable, although the moderate accuracy of C1 indicates that the precise cutoff may warrant further investigation.

6.5 Conclusions and Discussions

This chapter described MixQuery, a music retrieval system that allows users to iteratively refine queries using four instrumental parts. We implement an interface of a similarity-based music retrieval system that can combine multiple instrumental parts taken from different favorite songs. We demonstrated the effectiveness of our system through the subjective experiment, showing that the retrieved songs were judged to resemble the instrumental parts used in the query sets. This work enabled us to examine practically relevant applications of part-level similarity, indicating its potential

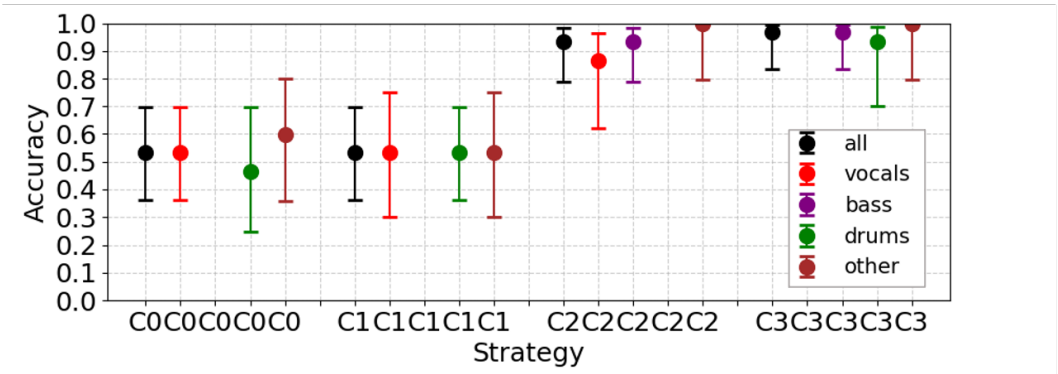


Figure 6.8: *Results of the subjective evaluation experiment; average accuracies and 95% confidence intervals. The black color indicates the results calculated from all the answers. The other colors indicate the results calculated from only the answers that include each stem. Red, purple, green, and brown indicate vocals, bass, drums, and other, respectively. Note that, for example, for the query set 0 in Table 6.1, which includes both vocals and drums, each participant provided one response. In this figure, however, that response is included in both the vocals and drums results. Since each strategy does not include all stems, as Table 6.1 shows, the stems with no results are left blank.*

to broaden the usefulness of similarity-based music retrieval.

7 Conclusions

7.1 Summary of This Thesis

This thesis explored research on part-level similarity of music, i.e., music similarity focusing on individual instrumental parts. The study was conducted from the following perspectives: constructing deep learning-based similarity estimation models and analyzing perceptual similarity, along with perceptually evaluating the models. Additionally, as an application of part-level similarity, a music retrieval interface was implemented.

Chapter 2 reviewed related work relevant to this thesis. Section 2.1 focused on content-based MIR, introducing studies closely related to Chapters 3 and 4. While data-driven approaches that learn similarity spaces in latent representations have been explored and shown to be effective, these methods take entire music tracks as input, without explicitly considering individual instrumental parts. Section 2.2 summarized prior work on perceptual music similarity in relation to Section 5. Previous studies have demonstrated the multidimensional nature of music similarity and have collected ground truth data through subjective listening experiments. However, perceptual similarity has not been investigated at the level of instrumental parts contained within music tracks. Section 2.3 reviewed music retrieval systems related to Chapter 6. Although retrieval systems that consider multiple perceptual perspectives or focus on vocal components have been proposed, no existing system allows users to focus on

individual instrumental parts beyond vocals and to flexibly combine multiple instrumental parts in query-based retrieval.

Chapter 3 proposed the first deep learning-based similarity estimation method. In this method, individual instrumental parts were separated at the signal level and then input into feature extraction models to obtain part-level feature representations. The Euclid distances between these representations were defined as the part-level similarity. Specifically, music tracks were input to a source separation model, and the estimated signals were processed by instrument-specific feature extractors to obtain the features. Experimental evaluation revealed that the accuracy of the extracted features is influenced by the performance of the source separation.

Chapter 4 proposed the second deep learning-based similarity estimation method, in which the separation of instrumental parts was performed at the feature representation level. The input was a music track, and without performing source separation, feature representations corresponding to each instrumental part were extracted directly. Using CSNs proposed initially for image domains, feature spaces representing the features of individual instrumental parts were learned for each subspace. Additionally, methods to facilitate learning, such as pseudo musical pieces, auxiliary losses, and pre-training, were proposed. Experimental evaluation showed improved accuracy compared to the source separation-based method, as well as the effectiveness of the proposed learning facilitation techniques. Correspondence with perceptual similarity was also confirmed.

Chapter 5 analyzed perceptual similarity. We investigated how perceptual music similarity at the part level is perceived and how it relates to track-level similarity. A large-scale listening experiment was conducted, collecting 26,898 responses from 632 participants, evaluating four aspects: timbre, rhythm, melody, and overall. The analysis suggested the following: (1) It is reasonable to compute similarity separately

for each part, as listeners have different criteria for music similarity perception across instrumental parts. (2) Estimating the importance of each instrumental part for a given music track and weighting part-level similarities accordingly has the potential to improve the accuracy of track-level similarity estimation. (3) Temporal average pooling leads to loss of rhythmic information, which is essential for similarity perception. This finding constitutes an improvement over the proposed method and can be leveraged in future model development. (4) The unsupervised learning approach used in our method is reasonable.

Chapter 6 presented a retrieval system as an application. The system, MixQuery, realized the retrieval using part-level similarity aggregation. Users can iteratively refine queries, freely adding their favorite part from any song. The experimental evaluation showed that the top results retrieved by the system using queries composed of multiple instrumental parts were appropriate given the combination of parts considered.

7.2 Future Work

As presented in Section 7.1, this thesis revealed several important findings regarding part-level similarity. However, there remain several issues that have not yet been fully addressed. This section discusses these remaining challenges as future work.

7.2.1 Rhythm-Aware Feature Extraction

Although our perceptual analysis indicated that rhythm plays an important role in part-level similarity, the current models failed to capture sufficient rhythmic information. Future work should focus on developing feature extraction methods that are more sensitive to rhythm. This could be further explored in future work by incorporating

attention-based temporal modeling or by learning representations through downstream rhythm-related tasks (e.g., beat tracking), among other possible approaches.

7.2.2 Evaluation and Training Using Real Instrumental Sounds

The limitation of the datasets used in this study is that they consist solely of MIDI-generated sounds. Since most real-world music is performed with acoustic instruments, experiments conducted only with MIDI data are insufficient. Among the potential limitations, the impact on source separation performance is particularly significant. In fact, the separation performance of the data used for training and evaluation in Chapter 3 was lower than the publicly reported performance of the Spleeter model (see Table 3.1 and [103]), which is likely due to the use of MIDI-generated sounds. Furthermore, it was shown that source separation quality affects the performance of feature extractors in Chapter 3. For these reasons, it is necessary to evaluate the models using recordings of real instruments.

7.2.3 Modeling Full-Length Tracks

In this study, 3-, 5-, and 10-second segments were used for inference and evaluation. A typical popular song has a duration of approximately several minutes; in fact, the original music tracks in the dataset in our experiments have an average duration of about four minutes before being divided. However, listeners typically consume music by listening to a track from start to finish, namely, a *full-length track*, and music retrieval systems usually treat a single track as the basic unit of retrieval. Therefore, modeling music only at the segment level may not fully reflect practical usage scenarios. In our retrieval interface, we handled this issue by computing similarity for each 10-

second segment, allowing users to select their favorite segments, and presenting results based on the most similar segment. Nevertheless, interactions across different time segments were not considered. Music tracks typically have a structured form, such as verses and choruses, and such global temporal structure and progression constitute an important aspect of musical content. Ignoring these temporal relationships limits the ability to fully model music. However, due to computational cost and memory constraints, directly using full-length tracks as model inputs is often impractical. As a result, representations are typically computed on shorter temporal segments and then aggregated. However, simple aggregation methods such as average pooling tend to obscure musically meaningful structural characteristics, such as section-wise progression, indicating the need for more sophisticated aggregation strategies. In addition, annotating full-length tracks is particularly expensive due to the long listening time required, resulting in a lack of fine-grained annotated datasets for full-length music.

Recently, full-length track music information retrieval has become a growing trend. For example, CoLLAP [129] addresses these challenges by segmenting full-length tracks into shorter clips and aggregating their representations with an attention-based mechanism guided by long-form language descriptions generated by music language models, enabling the preservation of temporal dynamics beyond simple pooling. Similarly, FUTGA [130] addresses full-length track modeling by leveraging synthetic, temporally structured music captions generated with large language models, enabling fine-grained and time-aware understanding of musical structure. Future work should explore modeling part-level music similarity for full-length tracks.

7.2.4 Application to Track-Level Similarity Estimation

We showed that some parts show consistency with the track-level similarity, while others do not in Chapete 5. Then, we found that the dominant instrumental part in the music track, which is consistency with the track-level perception, varies across music triplets and listeners, and the influence of differences of music triplets is more substantial. It suggests that estimating the importance of each instrumental part for a given track and weighting part-level similarities accordingly has the potential to improve the accuracy of track-level similarity estimation. However, this study has not yet investigated how the importance of each instrument is determined. Identifying and predicting such importance and examining its effect on track-level similarity estimation remain important directions for future work.

7.2.5 Evaluation of System Usability

Finally, while a retrieval interface was implemented as an application, the evaluation was limited to listening tests of retrieved songs. The user experience and overall usability of the interface remain unassessed. Future work should include formal user studies to evaluate whether the interface provides a satisfactory and effective retrieval experience.

7.2.6 Extension of the Framework to Other Musical Perspectives

In this thesis, a representation learning framework was proposed in which the embedding space is decomposed into multiple subspaces, each corresponding to an in-

strumental part, and several auxiliary strategies were introduced to effectively support this decomposition. While instrumental parts served as a practically meaningful application in this work, the framework itself is not limited to instrumental parts. It may be applicable to other perspectives of music similarity. As discussed in Chapter 5, different musical perspectives, timbre, rhythm, and melody capture distinct characteristics of music and contribute differently to human perception. These attributes could potentially be treated as separate perspectives for our framework. Exploring such extensions would enable a more general and flexible representation of music similarity and may further enhance interpretability as well as applicability to a wider range of music retrieval systems.

Acknowledgments

I wish to express my utmost gratitude to my advisor, Prof. Tomoki Toda of Nagoya University, for his exceptional guidance and steadfast support throughout my research. I have been deeply encouraged by his considerate choice of words, which always showed care not to hurt or discourage us. In addition, his precise advice and guidance in research planning allowed me to proceed toward achieving my research goals. From him, I have learned not only the technical aspects of research but also how to approach research with sincerity. I have also learned from him the true spirit of being an educator. Having had the opportunity to pursue my doctoral studies under Prof. Tomoki Toda has been one of the greatest treasures of my life.

I am also grateful to Dr. Wen-Chin Huang of Nagoya University, whose advice and daily discussions have greatly inspired me, and to Ms. Nami Noro, our lab's secretary, for her kind administrative support. I would like to express my deep gratitude to Dr. Li Li of CyberAgent, Inc., a former postdoctoral researcher of Nagoya University. She generously devoted a great deal of time to teach me many technical skills when I was still a master's student and had little experience. She also listened to my concerns about being a researcher, providing encouragement and guidance. She has been a constant source of inspiration to me and serves as a role model for the kind of researcher I aspire to become. I am also grateful to Dr. Yusuke Yasuda of the National Institute of Informatics, a former designated assistant professor at Nagoya University, for his

advice on my research and his organization of our server administration staff team. I would like to extend my appreciation to all the students in Toda Laboratory for their friendship and collaboration. In particular, I am thankful to Takehiro Imamura, who worked closely with me on my research and discussed our research together.

I am also grateful to Dr. Masataka Goto, Dr. Tomoyasu Nakano, Dr. Kosetsu Tsukuda, Dr. Kento Watanabe, Dr. Tian Cheng, and Dr. Takayuki Nakatsuka from the Media Interaction Group, National Institute of Advanced Industrial Science and Technology. I had the privilege of research with them during my technical training, which led to fruitful collaborations and co-authored publications.

I would like to express my gratitude to everyone in the speech, audio, and music signal processing community who have warmly welcomed me at conferences, engaged in lively discussions, and provided me with many valuable opportunities.

I would like to thank the professors, staff members, and fellow teaching assistants at the Center for Artificial Intelligence, Mathematical and Data Science, Nagoya University, for their kind support and cooperation during my time as a teaching assistant.

I am also thankful to my mentors and colleagues at the companies where I worked part-time research assistant or engineer, for providing me with valuable practical experience and insightful discussions.

Finally, I would like to express my heartfelt gratitude to my family and friends for their warm encouragement, understanding, and continuous support throughout this journey.

References

[1] IFPI, "Global music report 2025," 2025. [Online]. Available: https://www.ifpi.org/wp-content/uploads/2024/03/GMR2025_SOTI.pdf

[2] Apple Inc, "Apple music," 2025. [Online]. Available: https://www.apple.com/jp/apple-music/

[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] 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.

[5] M. Schedl, H. Zamani, C.-W. Chen, Y. Deldjoo, and M. Elahi, "Current challenges and visions in music recommender systems research," *International Journal of Multimedia Information Retrieval*, vol. 7, pp. 95–116, 2018.

[6] Ò. Celma Herrada, *Music recommendation and discovery in the long tail*. Universitat Pompeu Fabra, 2009.

[7] D. P.W.Ellis, B. Whitman, A. Berenzweig, and S. Lawrence, "The quest for ground truth in musical artist similarity." in *Proceedings of International Conference on Music Information Retrieval 2002*, 2002, pp. 13–17.

[8] J. S. Downie, "Music information retrieval," *Annual Review of Information Science and Technology*, vol. 37, no. 1, pp. 295–340, 2003.

[9] I. Deliège, "Introduction: Similarity perception - categorization - cue abstraction," *Music Percept*, vol. 18, no. 3, pp. 233–243, 2001.

[10] G. C. Cupchik, M. Rickert, and J. Mendelson, "Similarity and preference judgments of musical stimuli," *Scandinavian Journal of Psychology*, vol. 23, no. 1, pp. 273–282, 1982.

[11] S. McAdams and D. Matzkin, "Similarity, invariance, and musical variation," *Annals of the New York Academy of Sciences*, vol. 930, pp. 62—67, 2001.

[12] A. Volk and P. van Kranenburg, "Melodic similarity among folk songs: An annotation study on similarity-based categorization in music," *Musicae Scientiae*, vol. 16, no. 3, pp. 317–339, 2012.

[13] E. Pampalk, S. Dixon, and G. Widmer, "Exploring music collections by browsing different views," *Computer Music Journal*, vol. 28, no. 2, pp. 49–62, 2004.

[14] S. Stober and A. Nürnberger, "Musicgalaxy — an adaptive user-interface for exploratory music retrieval," in *Proceedings of Sound and Music Computing Conference 2010*, 2010, pp. 23–30.

[15] Y.-J. Luo, K. Agres, and D. Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture varia-

tional autoencoders," in *Proceedings of International Society for Music Information Retrieval 2019*, 2019, pp. 746–753.

[16] 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 *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 2021*, 2021, pp. 111–115.

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

[18] S. Essid, G. Richard, and B. David, "Instrument recognition in polyphonic music based on automatic taxonomies," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 68–80, 2005.

[19] T. Takahashi, S. Fukayama, and M. Goto, "Instrudive: A music visualization system based on automatically recognized instrumentation," in *Proceedings of International Society for Music Information Retrieval 2018*, 2018, pp. 561–568.

[20] G. Madison, F. Gouyon, F. Ullén, and K. Hörnström, "Modeling the tendency for music to induce movement in humans: first correlations with low-level audio descriptors across music genres," *Journal of experimental psychology. Human perception and performance*, vol. 37, no. 5, pp. 1578–1594, 2011.

[21] O. Senn, L. Kilchenmann, T. Bechtold, and F. Hoesl, "Groove in drum patterns as a function of both rhythmic properties and listeners' attitudes," *PLoS ONE*, vol. 13, 2018.

[22] J. A. Hockman and M. E. P. Davies, "Computational strategies for breakbeat classification and resequencing in hardcore, jungle and drum and bass," in *Proceedings of Digital Audio Effects 2015*, 2015.

[23] J. Herbst, "Historical development, sound aesthetics and production techniques of metal's distorted electric guitar," *Metal Music Studies*, vol. 3, 2017.

[24] H. Fujihara, M. Goto, T. Kitahara, and H. G. Okuno, "A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity-based music information retrieval," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 638–648, 2010.

[25] T. Nakano, M. Sasaki, M. Kishi, M. Hamasaki, M. Goto, and Y. Hijikata, "A music exploration interface based on vocal timbre and pitch in popular music," in *Proceedings of Computer Music Multidisciplinary Research 2023*, 2023.

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

[27] M. Schedl, E. Gómez, and J. Urbano, "Music information retrieval: Recent developments and applications," *Foundations and Trends in Information Retrieval*, vol. 8, no. 2–3, pp. 127–261, 2014.

[28] P. Knees, M. Schedl, and M. Goto, "Intelligent user interfaces for music discovery," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 165–179, 2020.

[29] J. T. Foote, "Content-based retrieval of music and audio," in *Proceedings of Multimedia storage and archiving systems II 1997*, vol. 3229, 1997, pp. 138–147.

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

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

[32] 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.

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

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

[35] J.-J. Aucouturier and F. Pachet, "Music similarity measures: What's the use?" in *Proceedings of International Conference on Music Information Retrieval 2002*, vol. 7, 2002, pp. 339–340.

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

[37] S. Dieleman and B. Schrauwen, "Multiscale approaches to music audio feature learning," in *Proceedings of International Society for Music Information Retrieval 2013*, 2013, pp. 116–121.

[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 *Proceedings of International Society for Music Information Retrieval 2013*, 2013.

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

[40] 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.

[41] D. Wolff and T. Weyde, "Learning music similarity from relative user ratings," *Information Retrieval*, vol. 17, pp. 109–136, 2014.

[42] 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 *Proceedings of International Conference on Modern Circuits and Systems Technologies 2022*, 2022, pp. 1–4.

[43] 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.

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

[45] I.-Y. Jeong and K. Lee, "Learning temporal features using a deep neural network and its application to music genre classification." in *Proceedings of International Society for Music Information Retrieval 2016*, 2016, pp. 434–440.

[46] Y. M.G. Costa, L. S. Oliveira, and C. N. Silla Jr, "An evaluation of convolutional neural networks for music classification using spectrograms," *Applied Soft Computing*, vol. 52, pp. 1–37, 2017.

[47] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proceedings of IEEE International conference on acoustics, speech and signal processing 2017*. IEEE, 2017, pp. 2392–2396.

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

[49] S. Hizlisoy, S. Yildirim, and Z. Tufekci, "Music emotion recognition using convolutional long short term memory deep neural networks," *Engineering Science and Technology, an International Journal*, vol. 24, no. 3, pp. 760–767, 2021.

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

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

[52] S. Sigtia and S. Dixon, "Improved music feature learning with deep neural networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 2014*. IEEE, 2014, pp. 6959–6963.

[53] 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.

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

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

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

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

[58] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis, "Mulan: A joint embedding of music audio and natural language," in *Proceedings of International Society for Music Information Retrieval 2022*, 2022, pp. 559–566.

[59] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Xiao, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi, W. Huang, Z. Wang, Y. Guo, and J. Fu, "MERT: Acoustic music understanding model with large-scale self-supervised training," in *Proceedings of International Conference on Learning Representations 2024*, 2024.

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

[61] T. LH. Li, A. B. Chan, and A. HW. Chun, "Automatic musical pattern feature extraction using convolutional neural network," *Lecture Notes in Engineering and Computer Science*, vol. 2180, 2010.

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

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

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

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

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

[67] J. Lee, J. Park, K. L. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," in *Proceedings of Sound and Music Computing Conference 2017*, 2017, pp. 220–226.

[68] X. Li, H. Xianyu, J. Tian, W. Chen, F. Meng, M. Xu, and L. Cai, "A deep bidirectional long short-term memory based multi-scale approach for music dynamic emotion prediction," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 2016*, 2016, pp. 544–548.

[69] T. Eerola, T. Järvinen, J. Louhivuori, and P. Toiviainen, "Statistical features and perceived similarity of folk melodies," *Music Perception: An Interdisciplinary Journal*, vol. 18, no. 3, pp. 275–296, 2001.

[70] A. Lamont and N. Dibben, "Motivic structure and the perception of similarity," *Music Perception*, vol. 18, no. 3, pp. 245–274, 2001.

[71] A. Novello, M. F. McKinney, and A. Kohlrausch, "Perceptual evaluation of music similarity," in *Proceedings of International Conference on Music Information Retrieval 2006*, 2006, pp. 246–249.

[72] A. Novello, M. M.F. McKinney, and A. Kohlrausch, "Perceptual evaluation of inter-song similarity in western popular music," *Journal of New Music Research*, vol. 40, pp. 1–26, 2011.

[73] A. Berenzweig, B. Logan, D. P.W. Ellis, and B. Whitman, "A large-scale evaluation of acoustic and subjective music similarity measures," *Computer Music Journal*, vol. 28, pp. 63–76, 2003.

[74] R. Typke, M. den Hoed, J. de Nooijer, F. Wiering, and R. C. Veltkamp, "A ground truth for half a million musical incipits," *Journal of Digital Information Management*, vol. 3, no. 1, pp. 34–39, 2005.

[75] D. Müllensiefen and K. Frieler, "Modelling experts' notions of melodic similarity," *Musicae Scientiae*, vol. 11, pp. 183–210, 2007.

[76] T. Eerola, R. Ferrer, and V. Alluri, "Timbre and affect dimensions: Evidence from affect and similarity ratings and acoustic correlates of isolated instrument sounds," *Music Perception: An Interdisciplinary Journal*, vol. 30, no. 1, pp. 49–70, 2012.

[77] S. McAdams, S. Winsberg, S. Donnadieu, G. D. Soete, and J. Krimphoff, "Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes," *Psychological Research*, vol. 58, pp. 177–192, 1995.

[78] S. Lakatos, "A common perceptual space for harmonic and percussive timbres," *Perception & Psychophysics*, vol. 62, no. 7, pp. 1426–1439, 2000.

[79] T. Kageyama, K. Mochizuki, and Y. Takashima, "Melody retrieval with humming," in *Proceedings of International Computer Music Conference 1993*, 1993, pp. 349–351.

[80] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search, and retrieval of audio," *IEEE MultiMedia*, vol. 3, no. 3, pp. 27–36, 1996.

[81] E. Pampalk, A. Rauber, and D. Merkl, "Content-based organization and visualization of music archives," in *Proceedings of ACM International Conference on Multimedia*, 2002, pp. 570–579.

[82] P. Knees, M. Schedl, T. Pohle, and G. Widmer, "An innovative three-dimensional user interface for exploring music collections enriched with meta-information from the web," in *Proceedings of ACM International Conference on Multimedia*, 2006.

[83] D. Lübbers and M. Jarke, "Adaptive multimodal exploration of music collections," in *Proceedings of International Conference on Music Information Retrieval 2009*, 2009.

[84] S. Leitich and M. Topf, "Globe of music: Music library visualization using geosom," in *Proceedings of International Conference on Music Information Retrieval 2007*, 2007.

[85] P. Lamere and D. Eck, "Using 3d visualizations to explore and discover music," in *Proceedings of International Conference on Music Information Retrieval 2007*, 2007.

[86] M. Hamasaki, M. Goto, and T. Nakano, "Songrium: Browsing and listening environment for music content creation community," in *Proceedings of Sound and Music Computing Conference 2015*, 2015, pp. 23–30.

[87] G. Tzanetakis, G. Essl, and P. Cook, "Automatic musical genre classification of audio signals," in *Proceedings of International Symposium on Music Information Retrieval 2001*, 2001.

[88] S. Vembu and S. Baumann, "A self-organizing map based knowledge discovery for music recommendation systems," in *Proceedings of International Symposium on Computer Music Modeling and Retrieval 2004*, 2004.

[89] I. Andjelkovic, D. Parra, and J. O ' Donovan, "Moodplay: Interactive music recommendation based on artists ' mood similarity," *International Journal of Human-Computer Studies*, vol. 121, pp. 142–159, 2019.

[90] B. Vad, D. Boland, J. Williamson, R. Murray-Smith, and P. B. Steffensen, "Design and evaluation of a probabilistic music projection interface," in *Proceedings of International Society for Music Information Retrieval 2015*, 2015, pp. 134–140.

[91] R. van Gulik and F. Vignoli, "Visual playlist generation on the artist map," in *Proceedings of International Conference on Music Information Retrieval 2005*, 2005, pp. 520–523.

[92] S. Sasaki, K. Yoshii, T. Nakano, M. Goto, and S. Morishima, "Lyricsradar: A lyrics retrieval system based on latent topics of lyrics," in *Proceedings of International Society for Music Information Retrieval 2014*, 2014, pp. 585–590.

[93] K. Tsukuda, K. Ishida, and M. Goto, "Lyric jumper: A lyrics-based music exploratory web service by modeling lyrics generative process," in *Proceedings of International Society for Music Information Retrieval 2017*, 2017, pp. 544–551.

[94] J. Serra, E. Gómez, and P. Herrera, "Audio cover song identification and similarity: background, approaches, evaluation, and beyond," in *Advances in music information retrieval*, 2010, pp. 307–332.

[95] S. H. Doh, M. Won, K. Choi, and J. Nam, "Toward universal text-to-music retrieval," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing 2023*, 2023, pp. 1–5.

[96] K. Mao, L. Shou, J. Fan, G. Chen, and M. S. Kankanhalli, "Competence-based song recommendation: Matching songs to one's singing skill," *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 396–408, 2015.

[97] K. Itoyama, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Query-by-Example music information retrieval by score-informed source separation and remixing technologies," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, Article ID 172961, pp. 1–14, 2010.

[98] B. McFee and G. R. G. Lanckriet, "Hypergraph models of playlist dialects," in *Proceedings of International Conference on Music Information Retrieval 2012*, 2012, pp. 343–348.

[99] K. Watanabe and M. Goto, "Query-by-Blending: A music exploration system blending latent vector representations of lyric word, song audio, and artist," in *Proceedings of International Society for Music Information Retrieval 2019*, 2019, pp. 144–151.

[100] S. Bostandjiev, J. O'Donovan, and T. Höllerer, "TasteWeights: A visual interactive hybrid recommender system," in *Proceedings of ACM conference on Recommender systems 2012*, 2012, pp. 35–42.

[101] M. Millecamp, N. N. Htun, Y. Jin, and K. Verbert, "Controlling spotify recommendations: Effects of personal characteristics on music recommender user interfaces," in *Proceedings of User Modeling, Adaptation and Personalization 2018*, 2018, pp. 101–109.

[102] 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 *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics 2019*, 2019, pp. 45–49.

[103] 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.

[104] R. Scheibler, "SDR — medium rare with fast computations," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 2022*, 2022, pp. 701–705.

[105] 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.

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

[107] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen, P. Nguyen, and R. Pang, "Hierarchical generative modeling for controllable speech synthesis," in *Proceedings of International Conference on Learning Representations 2019*, 2019.

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

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

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

[111] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in *Proceedings of International Society for Music Information Retrieval 2017*, 2017.

[112] "Crowdworks," https://crowdworks.jp.

[113] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[114] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 1947.

[115] C. J. Clopper and E. S. Pearson, "The use of confidence or fiducial limits illustrated in the case of the binomial," *Biometrika*, vol. 26, no. 4, pp. 404–413, 1934.

[116] T. Nakano, K. Yoshii, and M. Goto, "Musical similarity and commonness estimation based on probabilistic generative models of musical elements," *International Journal of Semantic Computing*, vol. 10, no. 1, pp. 27–52, 2016.

[117] M. Kamalzadeh, C. Kralj, T. Möller, and M. Sedlmair, "TagFlip: Active mobile music discovery with social tags," in *Proceedings of ACM Conference on Intelligent User Interfaces 2016*, 2016, pp. 19–30.

[118] T. Nakano and M. Goto, "LyricListPlayer: A consecutive-query-by-playback interface for retrieving similar word sequences from different song lyrics," in *Proceedings of Sound and Music Computing Conference 2016*, 2016, pp. 344–349.

[119] P. Knees and M. Schedl, "A survey of music similarity and recommendation from music context data," *ACM Transactions on Multimedia Computing, Communications*, vol. 10, no. 1, pp. 2:1–2:21, 2013.

[120] M. Mauch, R. M. MacCallum, M. Levy, and A. M. Leroi, "The evolution of popular music: USA 1960–2010," *Royal Society Open Science*, vol. 2, no. 5, pp. 1–10, 2015.

[121] E. Zangerle, R. Huber, M. Vötter, and Y.-H. Yang, "Hit song prediction: Leveraging low- and high-level audio features," in *Proceedings of International Society for Music Information Retrieval 2019*, 2019, pp. 319–326.

[122] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in *Proceedings of International Society for Music Information Retrieval 2002*, 2002, pp. 287–288.

[123] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," 2017.

[124] T. Bertin-Mahieux, D. P.W. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of International Conference on Music Information Retrieval 2011*, 2011, pp. 591–596.

[125] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing 2023*, 2023, pp. 1–5.

[126] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations 2015*, 2015.

[127] Y. Liu, "Deep learning based music recommendation systems: A review of algorithms and techniques," *Applied and Computational Engineering*, vol. 109, pp. 17–23, 2024.

[128] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, 1927.

[129] J. Wu, W. Li, Z. Novack, A. Namburi, C. Chen, and J. McAuley, "CoLLAP: Contrastive long-form language-audio pretraining with musical temporal structure augmentation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 2025*, 2025, pp. 1–5.

[130] J. Wu, Z. Novack, A. Namburi, H.-W. Dong, C. Chen, J. Dai, and J. McAuley, "FUTGA-MIR: Enhancing fine-grained and temporally-aware music understanding with music information retrieval," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 2025*, 2025, pp. 1–5.

List of Publications

Journal Papers

1. Y. Hashizume, L. Li, A. Miyashita, and T. Toda, "Learning Separated Representations for Instrument-based Music Similarity", APSIPA Transactions on Signal and Information Processing: vol. 14: no. 1, e16, 32 pages, 2025.

2. Y. Hashizume and T. Toda, "Investigation of Part-level Perceptual Music Similarity by Large-scale Listening Test", APSIPA Transactions on Signal and Information Processing: accepted.

3. T. Imamura, Y. Hashizume, W.-C. Huang, and T. Toda, "Music Similarity Representation Learning Focusing on Individual Instruments with Source Separation and Human Preference", APSIPA Transactions on Signal and Information Processing: vol. 14: no. 4, e303, 29 pages, 2025.

International Conferences

1. Y. Hashizume, L. Li, and T. Toda, "Music similarity calculation of individual instrumental sounds using metric learning", in Proceedings of Asia Pacific Signal

and Information Processing Association Annual Summit and Conference, pp. 33-38, 2022.

2. Y. Hashizume and T. Toda, "Investigation of perceptual music similarity focusing on each instrumental part," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pp.1–5, 2025.

3. T. Imamura, Y. Hashizume, and T. Toda, "Multi-task learning approaches for music similarity representation learning based on individual instrument sounds," in Proceedings of Asia Pacific Signal and Information Processing Association Annual Summit and Conference, pp. 1–6, 2024.

Domestic Conferences

1. 橋爪 優果, 李 莉, 戸田 智基, "各楽器音源に着目した距離学習に基づく楽曲間類似度計算," 音講論, 2-9-12, pp. 1207–1208, 2022.

2. 橋爪 優果, 李 莉, 戸田 智基, "各楽器音に着目した楽曲間類似度学習, " 情報処理研報, vol. 2022-MUS-134, no. 46, pp. 1–6, 2022.

3. 橋爪 優果, 李 莉, 戸田 智基, "各楽器音源に着目した楽曲間類似度学習の評価," 音講論, 3-1-5, pp. 1517–1518, 2022.

4. 橋爪 優果, "各楽器音に着目した楽曲間類似度学習, " 第 26 回東海地区音声関連研究室修士論文中間発表会, 2022.

5. 橋爪 優果, 李 莉, 宮下 敦志, 戸田 智基, "個別楽器音に基づいた楽曲間類似度のための分離表現学習," 情報処理研報, vol. 2023-MUS-137, no. 9, pp. 1–7, 2023.

6. 橋爪 優果, “個別楽器音に基づいた楽曲間類似度のための分離表現学習,” 日本音響学会東海支部 50 周年記念行事, 2023.

7. 橋爪 優果, 宮下 敦志, 李 莉, 戸田 智基, “多視点楽曲検索に向けた楽曲分離表現学習,” 人工知能学会全国大会論文集, 1O4-OS-29a-01, pp. 1–4, 2024.

8. 橋爪 優果, 戸田 智基, “各楽器パートに焦点を当てた知覚的楽曲間類似度の調査,” 音講論, 2-1-18, pp. 1369–1370, 2024.

9. 橋爪 優果, “ICASSP2025 における音楽情報処理の動向,” 信学技報, vol. 125, no. 36, EA2025-3, pp. 13–17 2025.

10. 今村 剛大, 橋爪 優果, 戸田 智基, “個別楽器音に基づく楽曲間類似度表現学習における音源分離の活用法,” 情報処理研報, vol. 2024-MUS-140, no. 54, pp. 1–7, 2024.

11. 今村 剛大, 橋爪 優果, ホワン ウェンチン, 戸田 智基, “個別楽器音に基づく知覚的楽曲間類似度表現学習,” 情報処理研報, vol. 2025-MUS-142, No. 7, pp. 1–9, 2025.

Awards

1. Japanese Society for Artificial Intelligence Annual Conference Award, 2023.

2. The Tokai Chapter of Japan Acoustical Society Outstanding Presentation Award, 2022.