MULTI-INSTRUMENT MUSIC TRANSCRIPTION BASED ON DEEP SPHERICAL CLUSTERING OF SPECTROGRAMS AND PITCHGRAMS

Keitaro Tanaka[1,*] **Takayuki Nakatsuka**[1] **Ryo Nishikimi**[2]
Kazuyoshi Yoshii[2] **Shigeo Morishima**[3]
[1] Waseda University, Japan [2] Kyoto University, Japan
[3] Waseda Research Institute for Science and Engineering, Japan
*phys.keitaro1227@ruri.waseda.jp

ABSTRACT

This paper describes a clustering-based music transcription method that estimates the piano rolls of arbitrary musical instrument parts from multi-instrument polyphonic music signals. If target musical pieces are always played by particular kinds of musical instruments, a way to obtain piano rolls is to compute the pitchgram (pitch saliency spectrogram) of each musical instrument by using a deep neural network (DNN). However, this approach has a critical limitation that it has no way to deal with musical pieces including undefined musical instruments. To overcome this limitation, we estimate a condensed pitchgram with an existing instrument-independent neural multi-pitch estimator and then separate the pitchgram into a specified number of musical instrument parts with a deep spherical clustering technique. To improve the performance of transcription, we propose a joint spectrogram and pitchgram clustering method based on the timbral and pitch characteristics of musical instruments. The experimental results show that the proposed method can transcribe musical pieces including unknown musical instruments as well as those containing only predefined instruments, at the state-of-the-art transcription accuracy.

1. INTRODUCTION

The problem of estimating the fundamental frequencies of multiple periodic signals, which is called multi-pitch estimation (MPE) [1], is an important task of music information retrieval (MIR) since it plays a basic role in automatic music transcription (AMT), which is a task of converting music signals into a symbolic form of music notation [2]. The conventional approaches to MPE primarily focused on transcribing single-instrument music signals. The accuracy of this single-instrument MPE (SI-MPE) has been greatly improved by deep learning. Recently, some studies have extended SI-MPE and have tackled the problem of multi-instrument MPE (MI-MPE) for further generalization. An MI-MPE is a task which estimates the pitchgrams (pitch saliency spectrograms) of every musical instrument from a music signal consisting of multiple instruments. The difficulty of MI-MPE in addition to SI-MPE is the necessity of estimating the corresponding instrument part which the pitchgram belongs to. To alleviate this difficulty, previous studies [3, 4] for MI-MPE limited their target musical instruments to a small number of predefined instruments. One of the solutions to this problem is applying a classification technique to MI-MPE and separate the music signal into each pitchgram.

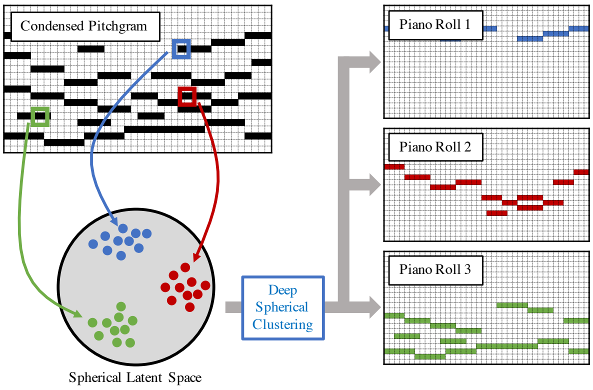

Figure 1. Each bin of a condensed pitchgram is embedded on a spherical latent space taking into account the timbral characteristics. Piano rolls of each instrument part is obtained by deep spherical clustering on the space.

These classification-based methods have been successful [3, 4] in the framework of supervised learning, especially for classical music where the constituent instruments are mostly fixed. However, in modern music (*e.g.*Pops and EDMs) where a larger number of instruments often appear, it would be ideal to have no limit on target instruments in order to achieve better AMT.

In the field of speech separation, several studies [5–7] have attempted a similar task of separating arbitrary speakers. When handling arbitrariness of the target sources in DNNs, technical problems related to permutations arise. Specifically, DNNs deterministically map inputs to a defined set of sources in each dimension, and thus does not allow permutation between different targets. To solve this permutation problem, a method called deep clustering

Attribution: K. Tanaka, T. Nakatsuka, R. Nishikimi, K. Yoshii, and S. Morishima, “Multi-Instrument Music Transcription Based on Deep Spherical Clustering of Spectrograms and Pitchgrams”, in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.,* Montréal, Canada, 2020.

has been proposed that treats speech separation for arbitrary speakers as a clustering problem, rather than a classification problem [5]. This approach avoids the above-mentioned problem and achieves optimal clustering at the same time by constructing an affinity matrix.

In this paper, we propose a new method to estimate the piano rolls of arbitrary musical instrument parts from multi-instrument polyphonic music signals based on deep clustering (Figure 1). We estimate a condensed pitchgram which shows all played pitches, with an existing instrument-independent neural multi-pitch estimator, and then separate the pitchgram into a specified number of musical instrument parts with a deep spherical clustering technique. Also, by considering the spectrogram in addition to the pitchgram in clustering phase, the optimal part estimation can be performed based on both the timbral and pitch characteristics of the instruments contained in the music signal. Furthermore, since there is a complementary relationship between MPE and sound source separation [8–10], we propose a joint spectrogram and pitchgram clustering method which can improve the transcription accuracy.

To verify that our method can transcribe arbitrary musical instruments, we conducted experiments of MI-MPE for various musical instruments. Experimental results show that the method can successfully handle a wide variety of instruments including those unseen during training. Although our method does not set any limitation on applicable instruments, the results suggest that it performs comparably to the state-of-the-art classification method [3].

Our main contribution of this study is the proposal of a new clustering-based method to transcribe arbitrary musical instrument parts from a music signal. To our knowledge, this is the first attempt of MI-MPE at frame-level without any restriction on used instruments. Furthermore, we show that the deep clustering method can be applied to tasks other than speech separation, and describe its potential in several sound related tasks.

2. RELATED WORK

In this section, we limit our scope to studies related to MI-MPE and methods dealing with arbitrariness of DNNs. Brief explanations of each study will be provided in the following subsections.

2.1 Multi-instrument Multi-pitch Estimation

Although AMT has been well studied, it still remains a challenging task [11]. Among the various tasks associated to AMT, MI-MPE is particularly difficult because it requires to simultaneously perform SI-MPE and instrument part estimation for each estimated note [2].

MI-MPE has commonly been tackled as a problem of stream-level transcription: grouping estimated notes and making continuous pitch contours for each part. Duan *et al.* [12] proposed a constrained clustering approach against the result of MPE. The clustering is performed under the constraint of consistency in each part of uniform discrete cepstrum. Their method can be used in complement with various MPE algorithms [13–15], and does not require any source model trained with isolated recordings of the underlying instruments. Following this study, Arora *et al.* [16] took a similar approach. They used probabilistic latent component analysis for MPE and source-specific feature extraction, and hidden Markov random fields for clustering into each instrument part. These two methods can deal with a variety of instruments, but due to their algorithms, each instrument must be a monophonic instrument which plays only one note at a time. Unlike these methods, Benetos *et al.* [17] focused on the differences in the sounds played by each instrument. They used spectral templates that correspond to sound states, supported by the shift-invariant probabilistic latent component analysis method. To conduct MPE for each instrument, they controlled the order of these templates by using hidden Markov model-based temporal constraints.

In recent years, some studies have tackled MI-MPE as a frame-level simultaneous MPE and musical instrument recognition problem. Wu *et al.* [3] proposed a DNN model based on the DeepLabV3+ [18] and U-Net structure [19]. They considered MI-MPE as a semantic segmentation problem on the time-frequency bins generated from music signals, where each object class represents a certain musical instrument. Most recently, Cerberus Network was proposed by Manilow *et al.* [4]. This model was built upon the preceding Chimera Network [20] which was developed for speech separation, adding a module that produces separated piano rolls for each instrument. The drawback of these methods is that only musical instruments included in the predefined set can be transcribed. In order to apply classification-based methods to source separation, output classes and object instances must be represented explicitly. Therefore, it is difficult to use these methods in the general case.

2.2 Arbitrariness with DNNs

In order to allow extraction of a piano roll of arbitrary instruments from an audio signal, the prediction itself must take place in a process where the instruments are unidentified, *i.e.*, individual piano rolls are referred to as instrument one, two, three instead of their specific identity such as piano, guitar, violin, etc. However, when doing so by using a DNN based approach, a problem of permutation arises as previously mentioned. Specifically, when the piano rolls of individual instruments are extracted but the instrument type is unknown, the loss between predicted piano roll and ground truth cannot be calculated straightforwardly since the correspondence of instrument type between these two remains unknown.

A similar problem has been addressed in the studies for speaker-independent speech separation [5–7], whose goal is to separate a piece of audio consisting of multiple people speaking simultaneously into individual speakers audio. Unless given an image or video of the target speaker, correspondence between separated audio and ground truth audio cannot be established, and hence the task poses the

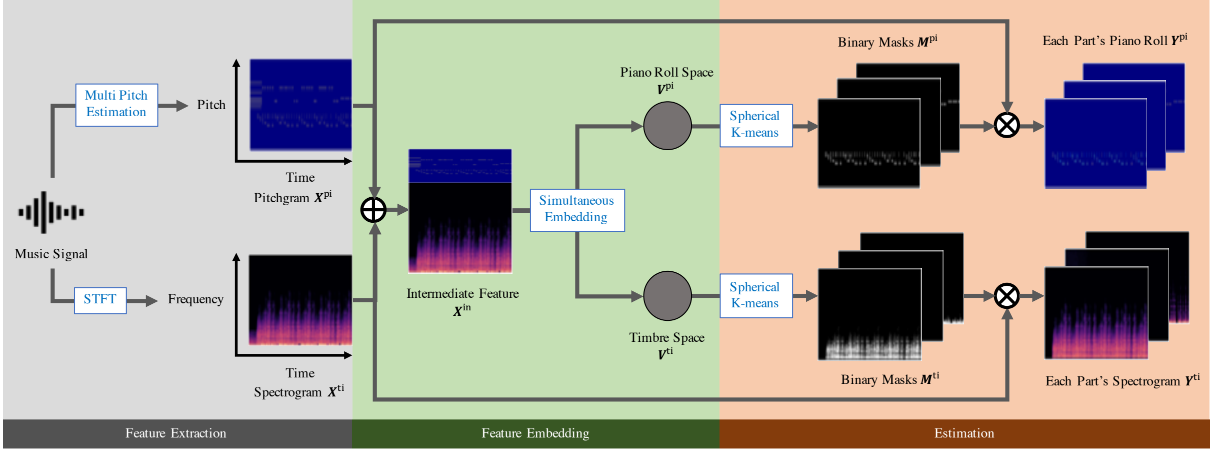
Multi Pitch Estimation
Pitch
Time
Pitchgram X^{pi}
Music Signal
STFT
Frequency
Time
Spectrogram X^{ti}
Intermediate Feature X^{in}
Simultaneous Embedding
Piano Roll Space V^{pi}
Timbre Space V^{ti}
Spherical K-means
Spherical K-means
Binary Masks M^{pi}
Binary Masks M^{ti}
Each Part's Piano Roll Y^{pi}
Each Part's Spectrogram Y^{ti}
Feature Extraction
Feature Embedding
Estimation

Figure 2. Overview of the proposed method. We extract two audio features $\boldsymbol{X}^{\mathrm{pi}}$ and $\boldsymbol{X}^{\mathrm{ti}}$ from an input music signal and concatenate them as an intermediate feature $\boldsymbol{X}^{\mathrm{in}}$. The feature $\boldsymbol{X}^{\mathrm{in}}$ is mapped into two latent space, piano roll space $\boldsymbol{V}^{\mathrm{pi}}$ and timbre space $\boldsymbol{V}^{\mathrm{ti}}$. We generate binary masks $\boldsymbol{M}^{\mathrm{pi}}$ and $\boldsymbol{M}^{\mathrm{ti}}$ from each space by clustering. These masks are applied to corresponding features, and we obtain piano rolls for each musical instrument part $\boldsymbol{Y}^{\mathrm{pi}}$ and separated spectrograms $\boldsymbol{Y}^{\mathrm{ti}}$.

same problem of permutation. To address this problem, methods such as permutation invariant training (PIT) [6], and deep clustering [5] have been proposed recently.

PIT tackled the permutation problem by calculating a loss function for all possible pairs of predicted values and ground truth, while optimization is only conducted for the pair with minimum loss. Although its implementation is simple and it can be combined with other learning techniques, its computational complexity remains considerably high. In details, when N sources are included in the target mixture, $N!$ possible permutations must be calculated in their algorithm.

On the other hand, deep clustering avoids the permutation problem by optimizing an embedded representation of the desired output, so that the class separation can be conducted via clustering in the embedded space at inference time. Given a $X \times D$ matrix $\boldsymbol{A}$ as the embedded representation, where X is the time-frequency index and D is the embedding dimension, the affinity matrix $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ is calculated. In the same manner, the affinity matrix $\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}$ is obtained for the ground truth data $\boldsymbol{B}$ which is a $X \times N$ matrix, where N represents the number of speakers. The optimization is conducted to minimize the distance between the two affinity matrices $||\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}||_F^2$. Here, deep clustering succeeds in circumventing the permutation problem as $(\boldsymbol{A}\boldsymbol{P})(\boldsymbol{A}\boldsymbol{P})^{\mathrm{T}} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ for any $D \times D$ permutation matrix $\boldsymbol{P}$. Furthermore, since optimization is conducted on the transformed $X \times X$ matrix, the target data may include any number of sources. For these advantages, we adopt the deep clustering method in our framework as described in Section 3.

3. PROPOSED METHOD

This section describes our proposed clustering-based method for the transcription of arbitrary musical instrument parts (Figure 2). Our framework consists of three parts: a feature extraction part, a feature embedding part to obtain piano roll space and timbre space, and an estimation part based on deep spherical clustering. We first pretrain the feature extraction part and the feature embedding part individually for the stabilization of early learning stages, then optimize both parts in conjunction through the entire learning.

3.1 Problem Configuration

Let $\boldsymbol{S} = \{\boldsymbol{s}_k \in \mathbb{R}^l\}_{k=1}^K$ be a set of mixture audio signals, where $l = 44.1\ [\mathrm{kHz}] \times 10\ [\mathrm{sec}]$ is a length of the signal, and K is the number of mixture audio signals. We assume that each $\boldsymbol{s}_n$ consists of three instrument parts. Let $\boldsymbol{Y}^{\mathrm{pi}} = \{\boldsymbol{y}_n^{\mathrm{pi}} \in [0,1]^{T \times C}\}_{n=1}^{N+1}$ be a set of pitchgrams of piano rolls, where T is the number of time frames, C is the number of constant-Q transform (CQT) frequency bins and N is the number of musical instrument parts. Our goal is to train a DNN f that maps $\boldsymbol{S}$ to $\boldsymbol{Y}^{\mathrm{pi}}$. Here, we incorporated two key ideas into f for the performance improvement and the stable training. Let $\boldsymbol{Y}^{\mathrm{ti}} = \{\boldsymbol{y}_n^{\mathrm{ti}} \in \mathbb{R}^{T \times F}\}_{n=1}^N$ be a set of corresponding spectrograms of the piano rolls, where F is the number of short-time Fourier transform (STFT) frequency bins. We train f that maps $\boldsymbol{S}$ to not only $\boldsymbol{Y}^{\mathrm{pi}}$, but also $\boldsymbol{Y}^{\mathrm{ti}}$ for improving the performance of a transcription. To achieve this with the stable training, we introduce an intermediate supervision, which consists of two semantic features. Let $\boldsymbol{X}^{\mathrm{pi}} \in [0,1]^{T \times C}$ and $\boldsymbol{X}^{\mathrm{ti}} \in \mathbb{R}^{T \times F}$ be a set of pitch characteristics and a set of timbral characteristics, respectively. We divided the f into two networks: feature extraction network g that maps $\boldsymbol{S}$ to $\boldsymbol{X}^{\mathrm{pi}}$, and feature embedding network h that maps the concatenation of $\boldsymbol{X}^{\mathrm{pi}}$ and $\boldsymbol{X}^{\mathrm{ti}}$ to $\boldsymbol{Y}^{\mathrm{pi}}$ and $\boldsymbol{Y}^{\mathrm{ti}}$. Firstly, the network g and h are trained individually for the stable training, and then our full network $f\ (= h \circ g)$ are jointly trained for the overall optimization.

3.2 Feature Extraction

In the feature extraction stage, a pitchgram and spectrogram are obtained from the input music signal, as pitch and timbral characteristics of each instrument are important for estimating piano rolls of each musical instrument part. For pitch, we computed a condensed pitchgram of the given music signal, in a form similar to a logarithmic frequency spectrogram using an instrument-independent neural multi-pitch estimator [21]. This network receives harmonic constant-Q transform (HCQT) as its input and outputs a condensed pitchgram denoted by $\boldsymbol{X}^{\mathrm{pi}}$. A value of each pitchgram bin is proportional to its salience.

We computed an STFT spectrogram $\boldsymbol{X}^{\mathrm{ti}}$ of the given music signal. Although there are other possible representations for timbral characteristics, we use STFT following the original deep clustering [5]. To reduce variations in total volume of input signals, the STFT spectrogram is normalized so that each time-frequency bin has a mean of zero and a standard deviation of one. Details are in Section 4.1.

3.3 Feature Embedding

We adopt joint learning of piano roll transcription and sound source separation. They are known to have a complementary relationship and have been reported to improve performance when they are learned simultaneously [8–10]. Following this knowledge, we propose a network based on deep spherical clustering that allows joint learning of transcription and separation. To learn the obtained pitch and timbral feature at the same time, we concatenate them along each frequency axis. This input feature $\boldsymbol{X}^{\mathrm{in}} \in \mathbb{R}^{T\times(C+F)}$ is used as the input to our network. The network maps the input feature $\boldsymbol{X}^{\mathrm{in}}$ to two separate latent spaces: piano roll space $\boldsymbol{V}^{\mathrm{pi}}$ and timbre space $\boldsymbol{V}^{\mathrm{ti}}$. The structure of our network is shown in Figure 3, where D and D' are the embedded dimensions of piano roll and timbre space. It consists of a three layer Bidirectional Long short-term memory (BLSTM), a fully connected (FC) layer for each space with tanh activation, and finally L^2 normalization. L^2 Normalization is conducted so that the piano roll space and timbre space respectively form a D and D' dimensional hypersphere.

The binary masks are made from the two latent spaces and applied to the pitchgram and the spectrogram later. In order to generate masks by clustering, all time-frequency bins have to be located ideally on the spherical latent spaces, *i.e.*, bins of the same source are close and bins of different sources are far apart. This can be achieved by constructing the affinity matrix of each space, $\boldsymbol{V}^{\mathrm{pi,ti}}{\boldsymbol{V}^{\mathrm{pi,ti}}}^{\mathrm{T}}$. Since $\boldsymbol{V}^{\mathrm{pi,ti}}$ is L^2 normalized, $TC \times TC$ or $TF \times TF$ matrix $\boldsymbol{V}^{\mathrm{pi,ti}}{\boldsymbol{V}^{\mathrm{pi,ti}}}^{\mathrm{T}}$ show cosine similarity of all time-frequency bins. Let $TC \times (N+1)$ matrix $\hat{\boldsymbol{M}}^{\mathrm{pi}}$ and $TF \times N$ matrix $\hat{\boldsymbol{M}}^{\mathrm{ti}}$ represent correct masks, where N is the number of musical instrument parts. We assume that each time-frequency bin is attributed to only one source. If more than one source share the same bin, the bin is assigned to the dominant source which has the largest volume (MIDI velocity) or the largest power spectrogram. $\hat{\boldsymbol{M}}^{\mathrm{pi,ti}}$ thus take binary value, one for assigned bin and zero for the opposite, and affinity matrix $\hat{\boldsymbol{M}}^{\mathrm{pi,ti}}{\hat{\boldsymbol{M}}^{\mathrm{pi,ti}}}^{\mathrm{T}}$ also have binary value. We can train this network using $\hat{\boldsymbol{M}}^{\mathrm{pi,ti}}{\hat{\boldsymbol{M}}^{\mathrm{pi,ti}}}^{\mathrm{T}}$ as target affinity matrix of $\boldsymbol{V}^{\mathrm{pi,ti}}{\boldsymbol{V}^{\mathrm{pi,ti}}}^{\mathrm{T}}$.

Note that we prepare an extra dimension for $\hat{\boldsymbol{M}}^{\mathrm{pi}}$. Because the condensed pitchgram $\boldsymbol{X}^{\mathrm{pi}}$ is a prediction, $\boldsymbol{X}^{\mathrm{pi}}$ may include misestimations, *i.e.*, false negatives and false positives. Among them, false positives should be treated as exceptions because they have no true instrumental attribution. We therefore prepare an additional dimension for bins which are silent in the ground truth, thus true negatives and false positives are put in this dimension. We also retain $\boldsymbol{X}^{\mathrm{ti}}$ bins whose magnitude is greater than the original maximum magnitude minus 40 dB. This prevents the network from considering about small power bins too much.

3.4 Training Strategy

Training of the multi-pitch estimator is conducted by minimizing the cross entropy loss shown in Eqn (1),

$$\mathcal{L}_{DS} = -\hat{\boldsymbol{X}}^{\mathrm{pi}}\log(\boldsymbol{X}^{\mathrm{pi}}) - (1-\hat{\boldsymbol{X}}^{\mathrm{pi}})\log(1-\boldsymbol{X}^{\mathrm{pi}}) \quad (1)$$

where $\hat{\boldsymbol{X}}^{\mathrm{pi}}$ and $\boldsymbol{X}^{\mathrm{pi}}$ represent the ground truth condensed pitchgram and the estimated condensed pitchgram. Both have values ranging from zero to one. Training of simultaneous embedding part is conducted to minimize Eqn (2).

$$\mathcal{L}_{DC}{}^{\mathrm{pi,ti}} = ||\boldsymbol{V}^{\mathrm{pi,ti}}{\boldsymbol{V}^{\mathrm{pi,ti}}}^{\mathrm{T}} - \hat{\boldsymbol{M}}^{\mathrm{pi,ti}}{\hat{\boldsymbol{M}}^{\mathrm{pi,ti}}}^{\mathrm{T}}||_F^2 \quad (2)$$

To reduce computational costs, we used a variation of Eqn (2) in practice.

$$\begin{aligned}\mathcal{L}_{DC}{}^{\mathrm{pi,ti}} = {} & ||{\boldsymbol{V}^{\mathrm{pi,ti}}}^{\mathrm{T}}\boldsymbol{V}^{\mathrm{pi,ti}}||_F^2 - 2||{\boldsymbol{V}^{\mathrm{pi,ti}}}^{\mathrm{T}}\hat{\boldsymbol{M}}^{\mathrm{pi,ti}}||_F^2 \\ & + ||{\hat{\boldsymbol{M}}^{\mathrm{pi,ti}}}^{\mathrm{T}}\hat{\boldsymbol{M}}^{\mathrm{pi,ti}}||_F^2 \quad (3)\end{aligned}$$

Direct construction of the original affinity matrix is avoided in Eqn (3) because TC and TF are much greater than D and D' [5]. Using these two kinds of losses, the total loss function is described as Eqn (4).

$$\mathcal{L}_{total} = \mathcal{L}_{DS} + \alpha\mathcal{L}_{DC}{}^{\mathrm{pi}} + \beta\mathcal{L}_{DC}{}^{\mathrm{ti}} \quad (4)$$

α and β in Eqn (4) are parameters to decide weights of each loss. We set them both at 0.000001 in our experiment.

For the stabilization of early learning stage, we first pretrained multi-pitch estimator and simultaneous embedding network respectively using the loss in Eqn (1) and Eqn (3). After pretraining, global optimization was conducted through end-to-end training by Eqn (4). We used Adam optimizer [22] for every training.

3.5 Estimation

At inference time, we generate two binary masks $\{\boldsymbol{M}_i^{\mathrm{pi}}\}_{i=1,\ldots,N+1}$ and $\{\boldsymbol{M}_j^{\mathrm{ti}}\}_{j=1,\ldots,N}$ for $\boldsymbol{X}^{\mathrm{pi}}$ and $\boldsymbol{X}^{ti}$ respectively from learned latent spaces $\boldsymbol{V}^{\mathrm{pi}}$ and $\boldsymbol{V}^{\mathrm{ti}}$. Mask generation is conducted by clustering the embedded features. Here, since the two spaces are hyperspherical shaped, we execute clustering by means of spherical k-means [23] though original deep clustering simply uses k-means. Because spherical k-means groups features based

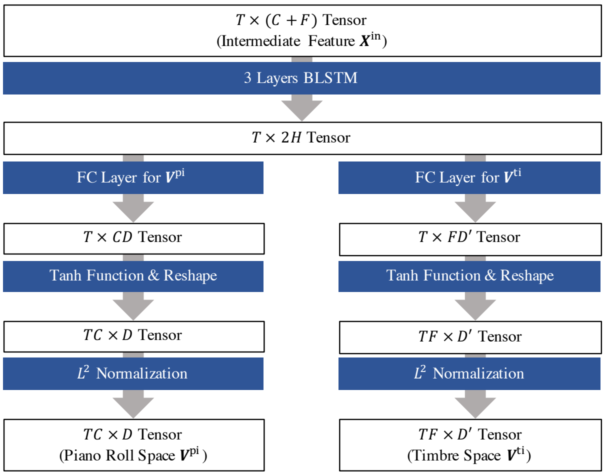

Figure 3. Details of simultaneous embedding part in Figure 2. $2H$ is the number of hidden nodes in BLSTM.

on their distance on a hypersphere, i.e. cosine distance, this should be applied for our purpose rather than k-means. Piano rolls of each musical instrument part and silent part $\{\boldsymbol{Y}_i^{\mathrm{pi}}\}_{i=1,\ldots,N+1}$ are calculated by Eqn (5).

$$\boldsymbol{Y}_i^{\mathrm{pi}} = \boldsymbol{X}^{\mathrm{pi}} \otimes \boldsymbol{M}_i^{\mathrm{pi}} \tag{5}$$

Additionally, spectrograms of each part $\{\boldsymbol{Y}_j^{\mathrm{ti}}\}_{j=1,\ldots,N}$ are obtained by Eqn (6), which can be converted to separated sounds of each instrument via inverse STFT.

$$\boldsymbol{Y}_j^{\mathrm{ti}} = \boldsymbol{X}^{\mathrm{ti}} \otimes \boldsymbol{M}_j^{\mathrm{ti}} \tag{6}$$

In Eqn (5) and Eqn (6), element wise product is described as $\otimes$. Since $\boldsymbol{M}_j^{\mathrm{ti}}$ is only for retained bins of $\boldsymbol{X}^{\mathrm{ti}}$, other bins are shared with all sources.

4. EVALUATION

4.1 Data

We used the Slakh2100-orig dataset [24] for our evaluation. The dataset contains 1500 training tracks, 375 validation tracks, and 225 test tracks. Each track is composed of multiple instruments, and the dataset consists of both mixed and separated sound sources with their MIDI data. It contains twelve kinds of instruments: piano, bass, guitar, drums, strings, synth pad, reed, brass, organ, pipe, synth lead, and chromatic percussion. We eliminated drums and chromatic percussion from the data to focus on instruments where pitch is important, *i.e.*, we used the other ten instruments for the experiment. To demonstrate the capability of estimating the piano rolls of arbitrary musical instrument part, we only used seven instruments (piano, bass, guitar, strings, synth pad, reed, and brass) for the training and validation data. We evaluated the performance using test data; above seven for the closed condition (seen instruments), and ten for the open condition (unseen instruments).

Training samples are constructed by cutting the tracks into ten seconds segments. Ground truth for condensed pitchgram is prepared by overlaying the MIDI data for the constituent sound sources. To make the mixture signal and the ground truth of condensed pitchgram, we overlaid both cut sound sources and MIDI data. Here, segments that do not have instrument sound for more than five seconds are omitted. The mixture MIDI data are binarized and gaussian blurred according to [21]. The musical recordings are mono-channel and their sampling rates are 44.1kHz. We computed STFT using Hann window with a size of 2048 time frames ≈ 50ms. The hop size is 512 frames ≈ 11ms for both STFT and HCQT. HCQT is computed for harmonics of {0.5, 1, 2, 3, 4, 5} with the minimum frequency 32.7Hz (C1) over six octaves. Our implementation uses the librosa library [25]. In total, 11 hours of training data and 3 hours of validation data were generated. For test data, 6 hours of data were generated for each condition.

4.2 Experimental Conditions

We evaluated the frame-level accuracy of transcriptions for each instrument part in the mixture. For the experiment, we fixed the number of mixed instruments to three, *i.e.*, $N = 3$. The transcription accuracy is evaluated by precision, recall, and F-measures. We count the pitchgram bin of a certain instrument as correct when binary values of estimation result and ground truth match with a correct part attribution. These metrics are calculated with Eqn (7),

$$P = \frac{TP}{TP + FP},\ R = \frac{TP}{TP + FN},\ F = \frac{2PR}{P + R} \tag{7}$$

where TP, FP, and FN are the number of true positive, false positive, and false negative, respectively. These values are calculated by the mir_eval [26] library.

To compare with the existing state-of-the-art classification-based method, we reimplemented [3] with eight output classes: the seven known instruments above and a non-instrument class. For a fair comparison between our clustering approach and the existing classification approach while considering the correctness of part attribution, we set evaluation conditions as follows:

1. In the clustering approach, part attribution is not conducted explicitly. Thus, clusters are assigned to each instrument source by optimizing the F-measure.
2. In the classification approach under the closed condition, estimated parts are directly used as part assignments.
3. In the classification approach under the open condition, by design, part attribution cannot be conducted for unknown instrument sources. Thus, estimated parts are reassigned to each instrument source included in the audio by optimizing the F-measure.

4.3 Experimental Results

The experimental results are shown in Table 1. Our proposed method outperformed the state-of-the-art classification-based method [3] in the transcription of unknown instruments under the open condition. Furthermore, the F-measure score of unknown instruments

	Closed condition						Open condition					
	[3]			Our method			[3]			Our method		
Instrument	P	R	F	P	R	F	P	R	F	P	R	F
Piano	51.28	46.50	**45.87**	62.02	39.61	44.07	52.51	48.04	**47.37**	61.87	38.90	43.64
Bass	73.75	58.79	**64.04**	39.72	50.78	42.24	74.27	59.66	**64.67**	40.59	51.88	43.23
Guitar	46.64	36.72	37.69	52.91	35.45	**39.46**	44.59	37.12	37.25	53.45	36.50	**40.32**
Strings	55.27	56.79	**52.74**	66.35	48.74	52.40	53.21	56.97	**52.05**	65.31	48.40	52.04
Synth pad	43.72	44.80	**42.07**	49.65	35.12	38.70	44.42	46.89	**43.91**	51.99	36.58	40.81
Reed	28.53	33.90	29.27	29.87	37.37	**31.53**	26.92	31.72	27.53	28.87	35.46	**30.04**
Brass	35.24	25.12	24.50	37.10	30.23	**29.53**	37.66	25.67	25.89	36.78	30.64	**30.26**
Organ	—	—	—	—	—	—	20.14	19.01	16.89	36.62	28.57	**29.11**
Pipe	—	—	—	—	—	—	22.62	27.13	23.02	38.37	39.49	**35.22**
Synth lead	—	—	—	—	—	—	20.58	17.44	17.59	29.41	25.11	**24.98**

Table 1. Comparative results of MI-MPE on the Slakh2100-orig dataset [24] with classification-based method by [3] and our method. P, R, and F are precision, recall and F-measure, respectively, defined in Eqn (7).

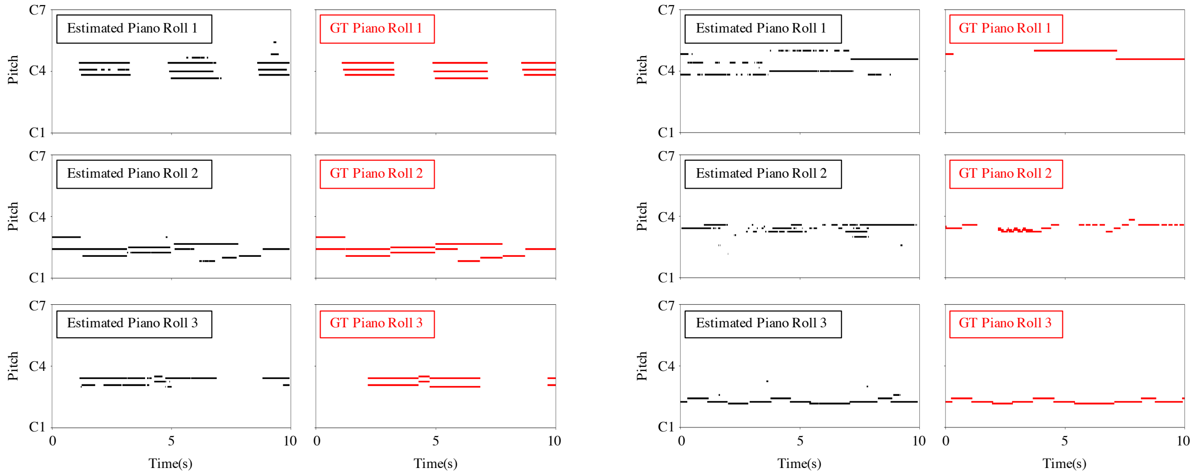

Figure 4. Transcribed piano rolls of each instrument part from the mixture signals. The left pairs are successful cases (Track01879) and the right pairs are failure cases (Track01878). The left column shows the estimated piano rolls (black) and the right column shows the ground truths (red). Each row shows the corresponding part, respectively.

was comparable to that of known instruments in our method, while the score significantly decreased in the classification-based method. Our method also succeeded in transcribing known instruments under both conditions at an accuracy equivalent to the classification-based method.

Examples of estimated piano rolls using our method are illustrated in Figure 4. In the successful cases, although some errors are present, it can be seen that our proposed method well-conducted pitch estimation and instrument assignment. In the failure cases, some notes which have to appear in piano roll two are transcribed in piano roll one around three seconds, in addition to many misestimations.

4.4 Discussion

Our method can obtain separated sounds of each instrument part in addition to their piano rolls; however, matching the estimated piano rolls and the instrument part labels still have to be done manually. One of the most interesting directions of this research is the automation of this process. Also, we assume that each time-frequency bin is attributed to only one source as mentioned in Section 3.3 though different instruments may share the same bin in practice. To deal with this case, another direction is to introduce the von Mises-Fisher (vMF) distribution [27, 28] into the hyperspherical latent space and perform soft clustering based on this distribution.

5. CONCLUSION

This paper presented a method for transcription of arbitrary musical instrument parts based on deep spherical clustering. Timbral and pitch characteristics of the music signal are simultaneously considered in the transcription, through joint clustering of a pitchgram and a spectrogram. The experimental results showed that the proposed method is capable of transcribing musical pieces including musical instruments not in training data. We plan to automate the matching process and introduce the vMF distribution into the hyperspherical latent space for future work.

6. ACKNOWLEDGEMENTS

This paper is supported by the JST ACCEL Grant Number JPMJAC1602, JSPS KAKENHI Grant Number JP16H01744, and JP19H04137.

7. REFERENCES

[1] M. G. Christensen, P. Stoica, A. Jakobsson, and S. H. Jensen, "Multi-pitch estimation," *Signal Processing*, vol. 88, no. 4, pp. 972–983, 2008.

[2] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine (SPM)*, vol. 36, no. 1, pp. 20–30, 2019.

[3] Y. Wu, B. Chen, and L. Su, "Polyphonic music transcription with semantic segmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 166–170.

[4] E. Manilow, P. Seetharaman, and B. Pardo, "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 771–775.

[5] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 31–35.

[6] D. Yu, M. Kolbæk, Z. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 241–245.

[7] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 246–250.

[8] S. Ewert, B. Pardo, M. Muller, and M. D. Plumbley, "Score-informed source separation for musical audio recordings: An overview," *IEEE Signal Processing Magazine (SPM)*, vol. 31, no. 3, pp. 116–124, 2014.

[9] Z. Duan and B. Pardo, "Soundprism: An online system for score-informed source separation of music audio," *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, vol. 5, no. 6, pp. 1205–1215, 2011.

[10] T. Nakano, K. Yoshii, Y. Wu, R. Nishikimi, K. W. Edward Lin, and M. Goto, "Joint singing pitch estimation and voice separation based on a neural harmonic structure renderer," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019, pp. 160–164.

[11] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Challenges and future directions," *Journal of Intelligent Information Systems (JIIS)*, vol. 41, 2013.

[12] Z. Duan, J. Han, and B. Pardo, "Multi-pitch streaming of harmonic sound mixtures," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 22, no. 1, pp. 138–150, 2014.

[13] Z. Duan, B. Pardo, and C.Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 18, no. 8, pp. 2121–2133, 2010.

[14] M. Wu, D. Wang, and G. J. Brown, "A multipitch tracking algorithm for noisy speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 11, no. 3, pp. 229–241, 2003.

[15] Z. Jin and D. Wang, "Hmm-based multipitch tracking for noisy and reverberant speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 19, no. 5, pp. 1091–1102, 2011.

[16] V. Arora and L. Behera, "Multiple f0 estimation and source clustering of polyphonic music audio using plca and hmrfs," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 23, no. 2, pp. 278–287, 2015.

[17] E. Benetos and S. Dixon, "Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model," *The Journal of the Acoustical Society of America (JASA)*, vol. 133, pp. 1727–1741, 2013.

[18] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," in *eprint arXiv:1706.05587*, 2017.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 9351, pp. 234–241, 2015.

[20] Y. Luo, Z. Chen, J. R. Hershey, J. Le Roux, and N. Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 61–65.

[21] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f_0 estimation in polyphonic music," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 63–70.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2014.

[23] I. S. Dhillon, J. Fan, and Y. Guan, “Efficient clustering of very large document collections,” *Data Mining for Scientific and Engineering Applications*, vol. 2, 2001.

[24] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, “Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019.

[25] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *Python in Science Conference (SciPy)*, 2015, pp. 18–24.

[26] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, “mir_eval: A transparent implementation of common mir metrics,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[27] S. Gopal and Y. Yang, “Von mises-fisher clustering models,” *International Conference on Machine Learning (ICML)*, vol. 1, pp. 269–302, 2014.

[28] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, “Clustering on the unit hypersphere using von mises-fisher distributions,” *Journal of Machine Learning Research (JMLR)*, vol. 6, pp. 1345–1382, 2005.