Joint Music Segmentation and Clustering Based on Self-Attentive Contrastive Learning of Multifaceted Self-Similarity Representation

Tsung-Ping Chen and Kazuyoshi Yoshii, *Senior Member, IEEE*

***Abstract*—This paper describes a method of music structure analysis that aims to partition a music recording into musically meaningful segments and group similar segments with the same label. A basic approach to this task is to extract latent acoustic features with a deep neural network (DNN) and then perform segmentation and clustering based on self-similarity matrices over those features. The performance of this approach, however, is essentially limited because the latent features are not necessarily optimal for the following step, and the self-similarity matrices have often been hand-crafted based on prior knowledge of musical sections. To overcome this limitation, we propose a jointly-trainable network that has a feature extraction subnetwork followed by segmentation and clustering branches. The extraction subnetwork is implemented with a Transformer encoder, whose multi-head self-attention mechanism is expected to learn multifaceted self-similarity matrices in a data-driven manner. The clustering branch is implemented by deep-unfolding the expectation-maximization (EM) algorithm of a Gaussian mixture model and thus has no trainable parameters. The segmentation branch is introduced for supervised boundary detection, encouraging the temporal continuity of labels estimated by the clustering branch. The evaluation results show the effectiveness of the joint optimization and the superiority of the proposed method over state-of-the-art methods.**

***Index Terms*—Clustering, contrastive learning, music structure analysis, segmentation, transformer encoder.**

I. Introduction

MUSIC structure analysis (MSA) aims to find musically meaningful segments that compose a music signal and to categorize the segments by their relationships. In the field of music information retrieval (MIR), it has a wide variety of applications including music summarization for effective trial listening with large-scale music collections. *Segmentation* and *labeling* are two typical subtasks of MSA, which aim to estimate musical boundaries and classify the segments, respectively. For the labeling task, *clustering* and *classification* frameworks have been addressed in previous research, where either semiotic labels (e.g., A, B, and C) [1], [2], [3] or semantic labels (e.g., intro, verse, and chorus) [4], [5] are used to categorize musical segments. Considering the absence of a standard taxonomy and a specification for the semantic labeling of music structure, we hence focus on the semiotic labeling using a clustering framework in this paper.

Fig. 1. The overview of music structure analysis. The frontend extracts latent features from a music signal, and the backend performs segmentation and clustering of the feature sequence.

A basic approach to MSA consists of a *feature extraction* frontend and a *segmentation-and-clustering* backend (Fig. 1). In the frontend, low-level features such as mel-spectrograms [6], chromagrams [7], and tempograms [8] are extracted from music signals as effective clues for MSA. Based on the acoustic features, higher-level representations can be further generated by using non-negative matrix factorization (NMF) [9], [10], hidden Markov models (HMMs) [11], or deep neural networks (DNNs) [12], [13]. In the backend, the self-similarity matrix (SSM) derived from the frontend representations is a building block of music segmentation [14], [15] and clustering [16], [17], [18], [19]. The main limitation of this two-stage framework is that the performance of MSA relies heavily on the compatibility between the frontend and backend.

To address this issue, we propose a holistic approach to MSA that tightly connects the feature extraction frontend and the segmentation-and-clustering backend with a joint optimization framework, inspired by similar ideas proposed in other fields [20], [21], [22], [23], [24], [25]. For the frontend, a variant

Received 1 October 2024; revised 22 February 2025; accepted 23 February 2025. Date of publication 6 March 2025; date of current version 21 March 2025. This work was supported in part by the JST FOREST under Grant JPMJFR2270, in part by the JST PRESTO under Grant JPMJPR20CB, and in part by the JSPS KAKENHI under Grant JP24H00742, Grant JP24H00748, and Grant JP24KJ1379. The associate editor coordinating the review of this article and approving it for publication was Zafar Rafii. *(Corresponding author: Tsung-Ping Chen.)*

Tsung-Ping Chen is with the Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (e-mail: chen.tsungping.74e@st.kyoto-u.ac.jp).

Kazuyoshi Yoshii is with the Graduate School of Engineering, Kyoto University, Kyoto 606-8501, Japan (e-mail: yoshii.kazuyoshi.3r@kyoto-u.ac.jp).

Digital Object Identifier 10.1109/TASLPRO.2025.3548449

of the Transformer encoder [26] is used to represent acoustic features at the frame level. It internally computes self-attention matrices that mimic multifaceted self-similarities [13] and thus has a high affinity for MSA. For the backend, a Gaussian mixture model (GMM) is integrated into the deep unfolding framework [27], [28] to cluster the latent representations. Specifically, the expectation-maximization (EM) algorithm comprising alternate E and M steps is unfolded to optimize the parameters of the GMM and refine the clustering estimation iteratively. In addition to the GMM, a segmentation branch estimating musical boundaries is employed in the backend to reduce the frequent label switching in the cluster assignments. Given the framewise posterior probabilities of classes (semiotic labels) estimated by the last E step, a contrastive learning strategy is then used to compute losses for optimizing the joint system. The employed contrastive losses are agnostic on the taxonomy of the ground-truth labels and thus can be applied to various datasets with distinct structure annotations.

The main contribution of this study is to propose a deep clustering framework [29], [30], [31] that connects the feature extraction stage with the clustering stage in the context of MSA. Our contrastive learning strategy can be readily used for various datasets in a taxonomy-agnostic manner. We demonstrate through extensive experiments that our method is capable of achieving the state-of-the-art performance of MSA.

II. Related Work

Audio-based MSA has been tackled by a clustering or a classification framework. The clustering-type MSA aims to identify the boundaries of structural units and categorize the units into a finite number of classes. The classification-type MSA additionally estimates the semantic labels (e.g., intro, verse, and chorus) of those classes. We refer the reader to [32] for a detailed review of the audio-based MSA.

The classification-type MSA has recently been approached with deep learning due to its excellent discriminative capability. Wang et al. [5] attempted to jointly predict the boundaries and semantic labels of musical segments with SpecTNT [33], a nested Transformer encoder [26], which learns both the spectral and temporal structures underlying the acoustic features. Shibata et al. [4] proposed a hidden semi-Markov model (HSMM) that represents the generative process of a sequence of acoustic features from a sequence of semantic labels with a long short-term memory (LSTM) network pretrained on paired data; during inference, the most-likely latent labels for a feature sequence can be estimated with the Viterbi algorithm. These methods, however, require a predefined vocabulary of semantic labels and cannot be applied directly to the hierarchical MSA [1], [19] due to the lack of well-defined semantic labels to describe the hierarchical structure of music.

For the clustering-type MSA, the early work took similarity-based approaches to group audio frames represented by handcrafted features. McFee and Ellis [1] performed spectral clustering (SC) on the SSM derived from two types of acoustic features. To improve segmentation accuracy, Wang and Mysore [2] used the variable Markov oracle (VMO) to represent the SSM before applying SC. Nieto and Bello [34] proposed the music structure analysis framework (MSAF) that standardizes the process of segmentation and clustering and implements several existing methods for clustering-type MSA.

Recent work on the clustering-type MSA leverages deep learning to obtain effective representations of handcrafted features before the clustering stage. Buisson et al. [12], [35] proposed an unsupervised contrastive learning framework that urges the representations of adjacent audio frames to be close to each other, and performed SC on the learned representations. Chen et al. [13] built a feature extraction model with a Transformer-based encoder that leverages the convolution-augmented multi-head self-attention (CAMHSA) mechanism to capture repetitive structures; the extracted features were grouped into K classes with a GMM, where the best K is selected according to the Bayesian information criterion (BIC) [36]. Such cascading approaches, however, are limited in performance because the learned features might be suboptimal for the subsequent clustering method. In this study, we propose a globally optimal version of [13] to bridge the gap between the feature extraction and clustering stages. Specifically, we concatenate a GMM-based clustering layer to a variant of the Transformer encoder and train the whole network in a supervised manner with the aid of contrastive learning.

III. Preliminaries

This section elaborates on two key techniques, i.e., deep unfolding and supervised clustering, used to integrate the GMM into a deep neural network (DNN). The former implements the EM algorithm with finite iterations as a parameter-free feed-forward network and the latter allows the entire network to be trained jointly. We first describe unsupervised learning of the GMM via deep unfolding [27], [28], and then describe supervised regularization via contrastive learning [37].

A. Deep Unfolding

The deep unfolding is a versatile technique that implements an iterative optimization algorithm as forward computation for a specific multi-layer network. This enables us to integrate a classical probabilistic model relying on iterative optimization with a deep learning model. The integrated network can thus be trained with backpropagation in a globally-optimal manner using a loss function defined for the entire network.

In this paper, we unfold the expectation-maximization (EM) algorithm [38], which is a convergence-guaranteed optimization algorithm that alternately iterates the expectation (E) and maximization (M) steps so that the marginal likelihood of the GMM parameters is maximized.

1) Gaussian Mixture Model: Consider a clustering problem that aims to categorize N samples into K classes. Let $\mathbf{X} \triangleq \{\mathbf{x}_n\}_{n=1}^N$ be observed variables, where $\mathbf{x}_n \in \mathbb{R}^{\mathrm{D}}$ is a D-dimensional vector. Let $\mathbf{Z} \triangleq \{z_n\}_{n=1}^N$ be the corresponding latent variables, where $z_n \in \{1, \ldots, K\}$ denotes the latent class of the n-th sample that generates $\mathbf{x}_n$. A K-component GMM

assumes a hierarchical generative process given by

$$p(z_n) = \text{Categorical}(z_n \mid \boldsymbol{\omega}), \tag{1}$$

$$p(\mathbf{x}_n \mid z_n) = \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_{z_n}, \boldsymbol{\Sigma}_{z_n}), \tag{2}$$

where $\boldsymbol{\omega} \triangleq \{\omega_k\}_{k=1}^K$ is a set of mixing ratios that sum to 1, i.e., $\sum_{k=1}^K \omega_k = 1$, and $\boldsymbol{\mu} \triangleq \{\boldsymbol{\mu}_k\}_{k=1}^K$ and $\boldsymbol{\Sigma} \triangleq \{\boldsymbol{\Sigma}_k\}_{k=1}^K$ are the mean vectors and covariance matrices of the K Gaussian distributions. Marginalizing z_n out, we get

$$p(\mathbf{x}_n) = \sum_{k=1}^K \omega_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{3}$$

This can be viewed as the marginal likelihood of the parameter set $\boldsymbol{\Theta} \triangleq \{\boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ for a given observation $\mathbf{x}_n$. The goal is to estimate $\boldsymbol{\Theta}$ that maximizes the marginal likelihood $p(\mathbf{x}_n)$.

2) Expectation-Maximization Algorithm: The E step computes the posterior distribution of the latent variables $\mathbf{Z}$ given the observation $\mathbf{X}$, while the M step updates the parameters $\boldsymbol{\Theta}$. These steps are alternately iterated until convergence. In the E step of the i-th iteration, given the latest estimate of $\boldsymbol{\Theta}$, denoted by $\boldsymbol{\Theta}^{(i-1)} \triangleq \{\boldsymbol{\omega}^{(i-1)}, \boldsymbol{\mu}^{(i-1)}, \boldsymbol{\Sigma}^{(i-1)}\}$, we compute the posterior probability as follows:

$$\begin{aligned} p(z_n = k \mid \mathbf{x}_n) &= \frac{p(\mathbf{x}_n, z_n)}{p(\mathbf{x}_n)} = \frac{p(\mathbf{x}_n \mid z_n)p(z_n)}{p(\mathbf{x}_n)} \\ &= \frac{\omega_k^{(i-1)} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k^{(i-1)}, \boldsymbol{\Sigma}_k^{(i-1)})}{\sum_{k'=1}^K \omega_{k'}^{(i-1)} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_{k'}^{(i-1)}, \boldsymbol{\Sigma}_{k'}^{(i-1)})} \triangleq \gamma_{nk}^{(i)}. \end{aligned} \tag{4}$$

In the M step, given the current estimate of the class posterior probability, i.e., $\gamma_{nk}^{(i)}$, the parameters are updated as follows:

$$N_k^{*(i)} = \sum_{n=1}^N \gamma_{nk}^{(i)}, \tag{5}$$

$$\omega_k^{(i)} = \frac{N_k^{*(i)}}{N}, \tag{6}$$

$$\boldsymbol{\mu}_k^{(i)} = \frac{1}{N_k^{*(i)}} \sum_{n=1}^N \gamma_{nk}^{(i)} \mathbf{x}_n, \tag{7}$$

$$\boldsymbol{\Sigma}_k^{(i)} = \frac{1}{N_k^{*(i)}} \sum_{k=1}^K \gamma_{nk}^{(i)} (\mathbf{x}_n - \boldsymbol{\mu}_k^{(i)})(\mathbf{x}_n - \boldsymbol{\mu}_k^{(i)})^\top. \tag{8}$$

After a sufficient number of iterations, denoted by I, we obtain the clustering result, where the n-th sample is categorized into the k-th class with the probability of $\gamma_{nk}^{(I)}$.

3) Network Implementation: The EM algorithm with I iterations can be unfolded into a feed-forward network consisting of I layers. The i-th layer takes as input both the observation $\mathbf{X}$ and the parameter estimate $\boldsymbol{\Theta}^{(i-1)}$ given by the $(i-1)$-th layer and outputs the new estimate $\boldsymbol{\Theta}^{(i)}$, where (4)–(8) are sequentially computed inside the layer. The initial estimate $\boldsymbol{\Theta}^{(0)}$ fed into the first layer is obtained with random sampling outside the network consisting of only deterministic transforms in a sense similar to the reparameterization trick [39]. The final estimate of the class posterior probabilities $\boldsymbol{\gamma}_n \triangleq \{\gamma_{nk}^{(I)}\}_{k=1}^K$ are obtained from the last I-th layer.

Note that the unfolded network is not *trainable* in the sense of deep learning terminology. The forward computation through the stacked I layers intrinsically maximizes the marginal likelihood of $\boldsymbol{\Theta}$ as the EM algorithm does.

4) Clustering vs Classification: The unsupervised clustering model based on the unfolded EM algorithm may seem similar to the typical supervised classification model with the softmax function in a sense that they both yield the posterior probabilities of K classes. In the classification task, one may use a one-layer network given by

$$p(z_n = k \mid \mathbf{x}_n) = \frac{e^{\mathbf{w}_k^\top \mathbf{x}_n}}{\sum_{k'=1}^K e^{\mathbf{w}_{k'}^\top \mathbf{x}_n}}, \tag{9}$$

where $\mathbf{w}_k \in \mathbb{R}^{\mathrm{D}}$ is a trainable representation of the k-th class, which is optimized for paired data of $\mathbf{X}$ and $\mathbf{Z}$ in the training phase such that the posterior probability $p(z_n \mid \mathbf{x}_n)$ for the ground-truth class z_n is maximized. In the test phase, $\mathbf{w}_k$ is *frozen* and is used directly to classify new samples, making the classification model different from the clustering model, where the GMM parameters $\boldsymbol{\Theta}$ are optimized by the EM steps for the given $\mathbf{X}$ in both the training and the test phases.

The clustering model has no *trainable* parameters, but proves its worth when connected to a trainable feature extraction model. Thanks to the differentiability of the clustering model with respect to the posterior probabilities $\gamma_{nk}^{(I)}$, a loss evaluated for $\gamma_{nk}^{(I)}$ can be backpropagated through the I layers of the clustering model to the feature extraction model. The feature extraction model can thus be trained so that the extracted latent features are optimal for clustering.

B. Supervised Clustering

In supervised clustering, where ground-truth classes $\mathbf{Z}$ of observed data $\mathbf{X}$ are given, a DNN can be trained such that the posterior probability $p(\mathbf{Z} \mid \mathbf{X})$ estimated by the DNN is maximized, as in supervised classification. Due to the class-agnostic nature of clustering, the major problem of such supervised training is the alignment between the ground-truth annotations and the class indices. A basic solution to this problem is permutation-invariant training (PIT) [40], [41], [42] that computes the posterior probabilities for all possible alignments and uses the highest posterior probability (best alignment) as a maximization target. This approach, however, is computationally inefficient as it requires $K!$ permutations for K classes.

Another promising solution to the permutation ambiguity in supervised clustering is contrastive learning that computes pairwise losses without referring to class indices [22], [31], [43]. Specifically, one can train a DNN such that the posterior distribution $p(z_a|\mathbf{x}_a)$ given by the DNN is closer to $p(z_b|\mathbf{x}_b)$ than to $p(z_c|\mathbf{x}_c)$ if $\mathbf{x}_a$ and $\mathbf{x}_b$ are drawn from the same class and $\mathbf{x}_c$ is drawn from another class:

$$\sum_{k=1}^K \gamma_{ak} \cdot \gamma_{bk} > \sum_{k=1}^K \gamma_{ak} \cdot \gamma_{ck}, \tag{10}$$

where $\gamma_{nk} = p(z_n = k|\mathbf{x}_n)$. We use this technique to compute the loss function with the posterior probabilities given by the

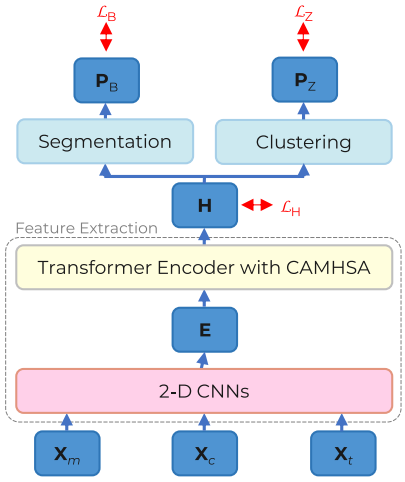


Fig. 2. Schematic diagram of the proposed model.

unfolded GMM layers. This enables us to jointly optimize the GMM and the feature extraction network using the backpropagation algorithm.

IV. Proposed Method

We propose a novel clustering-type MSA method based on joint optimization of feature extraction and clustering.

A. Problem Specification

Given a sequence of acoustic features $\mathbf{X} \triangleq \{\mathbf{x}_n \in \mathbb{R}^F\}_{n=1}^N$ extracted from a music signal, we aim to estimate a set of consecutive non-overlapping segments $\mathbf{S} \triangleq \{s_m\}_{m=1}^M$ covering the whole sequence, where F is the feature dimension, N is the number of frames, M is the number of segments, and $s_m \triangleq (n_m, k_m)$ with n_m and k_m being the start frame and the class of the m-th segment, respectively.

We reformulate this task as a sequence labeling problem and estimate 1) a sequence of binary variables $\mathbf{B} \triangleq \{b_n \in \{0,1\}\}_{n=1}^N$ representing the absence or presence of segment boundaries and 2) a sequence of class indices $\mathbf{Z} \triangleq \{z_n \in \{1,\dots,K\}\}_{n=1}^N$. The boundary information $\mathbf{B}$ is necessary to retrieve consecutive segments of the same class from $\mathbf{Z}$. Specifically, $\mathbf{S}$ can be retrieved from $\mathbf{B}$ and $\mathbf{Z}$ by taking $n_m \in \{n : b_n = 1\}$ and $k_m = z_{n_m}$.

B. Network Design

Our network is based on a feature extraction model followed by clustering and segmentation branches (Fig. 2).

1) Feature Extraction: We aim to extract from the input $\mathbf{X}$ a sequence of latent features $\mathbf{H} \triangleq \{\mathbf{h}_n \in \mathbb{R}^D\}_{n=1}^N$ based on the self-similarity of $\mathbf{X}$, which has been considered to be one of the most effective clues for MSA. This approach raises a research question: how to compute the self-similarity from what? Our approach to this problem is to combine a convolutional neural network (CNN) that locally extracts latent features and a Transformer encoder that globally aggregates these features based on the multi-head self-attention maps, which uncover a multifaceted self-similarity [13].

Specifically, a sequence of intermediate latent features $\mathbf{E} \triangleq \{\mathbf{e}_n \in \mathbb{R}^D\}_{n=1}^N$ is extracted from $\mathbf{X}$ with 2-D CNNs, where the feature dimension is transformed from F to D while the number of frames N is preserved:

$$\mathbf{E} = \mathrm{CNN}(\mathbf{X}). \tag{11}$$

Following [13], three types of acoustic features are used as inputs, i.e., the mel spectrogram ($\mathbf{X}_m$), the chromagram ($\mathbf{X}_c$), and the tempogram ($\mathbf{X}_t$), each of which is processed with a CNN, and the latent features are concatenated to form $\mathbf{E}$.

Then, $\mathbf{H}$ is obtained by representing $\mathbf{E}$ with a variant of the Transformer encoder with the CAMHSA mechanism [13] considering the global dependency:

$$\mathbf{H} = \mathrm{TransformerEncoder}(\mathbf{E}). \tag{12}$$

2) Segmentation: We aim to estimate $\mathbf{B}$ from $\mathbf{H}$ with a segmentation network, which can achieve better boundary detection compared to using only a clustering network. Specifically, the posterior probabilities $p(\mathbf{B}|\mathbf{H}) = \{\beta_n \in [0,1]\}_{n=1}^N \triangleq \mathbf{P}_\mathrm{B}$ are computed as follows:

$$\mathbf{P}_\mathrm{B} = \mathrm{sigmoid}(f_c'(\mathrm{ReLU}(f_c(\mathbf{H})))), \tag{13}$$

where $f_c : \mathbb{R}^{N\times D} \mapsto \mathbb{R}^{N\times D}$ and $f_c' : \mathbb{R}^{N\times D} \mapsto \mathbb{R}^{N\times 1}$ denote 1-D convolutions along the N dimensions.

3) Clustering: We aim to estimate $\mathbf{Z}$ from $\mathbf{H}$ with a K-component GMM. The deep unfolding technique described in Section III-A is used to make the whole network differentiable. For the initial estimate $\boldsymbol{\Theta}^{(0)} \triangleq \{\boldsymbol{\omega}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Sigma}^{(0)}\}$, $\omega_k^{(0)}$ is set to be $1/K$, $\boldsymbol{\mu}_k^{(0)} \in \mathbb{R}^D$ is drawn from a standard Gaussian distribution, $\boldsymbol{\Sigma}_k^{(0)} \in \mathbb{R}^{D\times D}$ is set to an identity matrix. In this paper, each $\boldsymbol{\Sigma}_k^{(i)}$ is assumed to be a diagonal matrix because this worked comparably with the full covariance modeling in our preliminary experiments. The latent features $\mathbf{H}$ (corresponding to $\mathbf{X}$ in Section III-A) are fed to the network, and the posterior probabilities $p(\mathbf{Z}|\mathbf{H}) = \{\boldsymbol{\gamma}_n \in [0,1]^K\}_{n=1}^N \triangleq \mathbf{P}_\mathrm{Z}$ are obtained through the unfolded EM steps:

$$\mathbf{P}_\mathrm{Z} = \mathrm{UnfoldedEM}(\mathbf{H}). \tag{14}$$

To achieve hierarchical MSA, we introduce additional sets of the segmentation and clustering branches into the network. In this scenario, the feature extraction model is shared by all the branches in a multi-task learning manner.

C. Inference

Given the boundary posterior probabilities $\mathbf{P}_\mathrm{B}$ and the class posterior probabilities $\mathbf{P}_\mathrm{Z}$, our goal is to estimate both the segment boundaries $\mathbf{B}$ and the segment classes $\mathbf{Z}$. A naive solution is to determine $\mathbf{Z}$ from $\mathbf{P}_\mathrm{Z}$ such that $\gamma_{nz_n} = p(z_n|\mathbf{h}_n)$ is maximized at each frame n. However, the resulting $\mathbf{Z}$ tends to be fragmented as the GMM does not consider temporal continuity over the frames. We therefore retrieve $\mathbf{B}$ from $\mathbf{P}_\mathrm{B}$, and $\mathbf{Z}$ is determined subsequently by both $\mathbf{B}$ and $\mathbf{P}_\mathrm{Z}$.

More specifically, we detect boundary frames n_m from $\mathbf{P}_{\mathrm{B}}$ using a peak-picking algorithm proposed for MSA [34].[1] $\mathbf{Z}$ is then determined by aggregating the class posterior probabilities between adjacent boundaries:

$$z_n = \arg\max_k \sum_{n'=n_m}^{n_{m+1}-1} \gamma_{n'k} \quad \forall n \in [n_m, n_{m+1}). \tag{15}$$

D. Training

Given the segment boundaries $\mathbf{B}$ and the class indices $\mathbf{Z}$, we aim to jointly train the feature extraction, segmentation, and clustering models in a supervised manner (Fig. 2). The loss function $\mathcal{L}$ to be minimized is given by

$$\mathcal{L} = \mathcal{L}_{\mathrm{H}} + \mathcal{L}_{\mathrm{B}} + \mathcal{L}_{\mathrm{Z}}, \tag{16}$$

where $\mathcal{L}_{\mathrm{H}}$, $\mathcal{L}_{\mathrm{B}}$, and $\mathcal{L}_{\mathrm{Z}}$ are the feature extraction loss, the segmentation loss, and the clustering loss, respectively.

1) Feature Extraction Loss: To encourage the feature extraction model to produce latent features $\mathbf{H}$ with better cluster separability, we use a contrastive learning approach based on the ground-truth $\mathbf{Z}$ in a class-agnostic fashion (Section III-B). We thus minimize the *intra-class* variances and maximize the *inter-class* variances on $\mathbf{H}$.

More specifically, $\mathcal{L}_{\mathrm{H}}$ is defined as a class-agnostic contrastive loss that makes $\mathbf{h}_n$ close to the centroid of the class it belongs to and far away from the centroids of the other classes as proposed in [44], [45], [46]:

$$\mathcal{L}_{\mathrm{H}} = -\frac{1}{N}\sum_{n=1}^{N} \log \frac{e^{-\|\mathbf{h}_n - \mathbf{g}_{z_n}\|_2}}{\sum_{k \neq z_n} e^{-\|\mathbf{h}_n - \mathbf{g}_k\|_2}}, \tag{17}$$

where $\mathbf{g}_k$ is the centroid of class k defined as follows:

$$\mathbf{g}_k = \frac{1}{N_k}\sum_{n=1}^{N} \mathbb{1}_k(z_n)\mathbf{h}_n, \tag{18}$$

where $\mathbb{1}_k(z_n)$ is an indicator function that takes 1 if $z_n = k$ and 0 otherwise, and N_k is the number of frames in class k.

2) Segmentation Loss: Considering that positive samples (boundary frames) are much fewer than negative samples (non-boundary frames) in the ground-truth $\mathbf{B}$ (i.e., $\mathbf{B}$ is *imbalanced*), we use the logarithmic dice loss [47] as well as the standard cross entropy loss for supervised learning. Specifically, $\mathcal{L}_{\mathrm{B}}$ is given by a weighted sum of the binary cross entropy loss $\mathcal{L}_{\mathrm{B}}^{\mathrm{ce}}$ and the logarithmic dice loss $\mathcal{L}_{\mathrm{B}}^{\mathrm{ld}}$:

$$\mathcal{L}_{\mathrm{B}} = \mathcal{L}_{\mathrm{B}}^{\mathrm{ce}} + \lambda_{\mathrm{B}}\mathcal{L}_{\mathrm{B}}^{\mathrm{ld}}, \tag{19}$$

$$\mathcal{L}_{\mathrm{B}}^{\mathrm{ce}} = -\frac{1}{N}\sum_{n=1}^{N} b_n \log(\beta_n) + (1-b_n)\log(1-\beta_n), \tag{20}$$

$$\mathcal{L}_{\mathrm{B}}^{\mathrm{ld}} = -\log\left(\frac{2\sum_{n=1}^{N} b_n\beta_n}{\sum_{n=1}^{N} b_n + \sum_{n=1}^{N}\beta_n}\right), \tag{21}$$

where λ_{B} is a weighting factor. We combine the two losses for they concern different aspects of the input. The cross entropy loss is calculated for each individual frame, while the dice loss underlines the collective behavior of the input as a whole.

3) Clustering Loss: Considering the imbalance of class frequencies in the ground-truth $\mathbf{Z}$, the clustering loss is defined in a way similar to the segmentation loss. Instead of using the computationally-prohibitive PIT, we take a contrastive learning approach. Let $\hat{\mathbf{Z}} \in \{0,1\}^{N\times N}$ be a binary matrix, where $\hat{Z}_{ij}$ takes 1 if frames i and j belong to the same class (i.e., $z_i = z_j$), and 0 otherwise. Similarly, let $\hat{\mathbf{\Gamma}} \in [0,1]^{N\times N}$ be a real-valued matrix, where $\hat{\Gamma}_{ij} \triangleq \boldsymbol{\gamma}_i^{\top}\boldsymbol{\gamma}_j$ is assumed to represent the posterior probability of $\hat{Z}_{ij}$. We define $\mathcal{L}_{\mathrm{Z}}$ as a weighted sum of the binary cross entropy loss $\mathcal{L}_{\mathrm{Z}}^{\mathrm{ce}}$ and the logarithmic dice loss $\mathcal{L}_{\mathrm{Z}}^{\mathrm{ld}}$:

$$\mathcal{L}_{\mathrm{Z}} = \mathcal{L}_{\mathrm{Z}}^{\mathrm{ce}} + \lambda_{\mathrm{Z}}\mathcal{L}_{\mathrm{Z}}^{\mathrm{ld}}, \tag{22}$$

$$\mathcal{L}_{\mathrm{Z}}^{\mathrm{ce}} = -\frac{1}{N^2}\sum_{i,j=1}^{N} \hat{Z}_{ij}\log(\hat{\Gamma}_{ij}) + (1-\hat{Z}_{ij})\log(1-\hat{\Gamma}_{ij}), \tag{23}$$

$$\mathcal{L}_{\mathrm{Z}}^{\mathrm{ld}} = -\log\left(\frac{2\sum_{i,j=1}^{N}\hat{Z}_{ij}\hat{\Gamma}_{ij}}{\sum_{i,j=1}^{N}\hat{Z}_{ij} + \sum_{i,j=1}^{N}\hat{\Gamma}_{ij}}\right), \tag{24}$$

where λ_{Z} is a weighting factor.

V. Evaluation

We report comparative experiments and ablation studies carried out to evaluate the proposed MSA method in terms of segmentation and clustering.

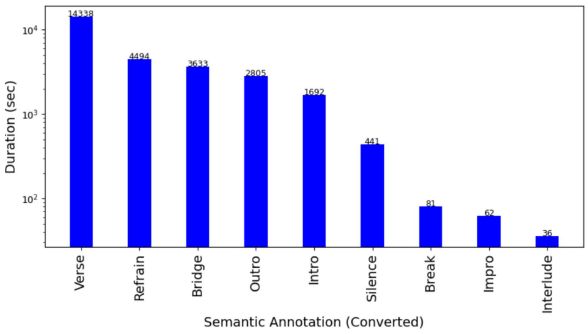
A. Data

For our experiments, we used the Beatles dataset [48] and the SALAMI dataset [49], which have been commonly used in studies on MSA. For the Beatles dataset, we used 174 tracks with refined structure annotations.[2] For the SALAMI dataset, we used 996 tracks (441 from the Internet Archive[3] and 555 from YouTube[4]) with the version 2.0 annotations.[5] The SALAMI annotations contain both semantic labels representing *flat* music structures and semiotic labels representing *hierarchical* music structures. We used the semiotic labels to evaluate the ability of the proposed method to analyze hierarchical structures.

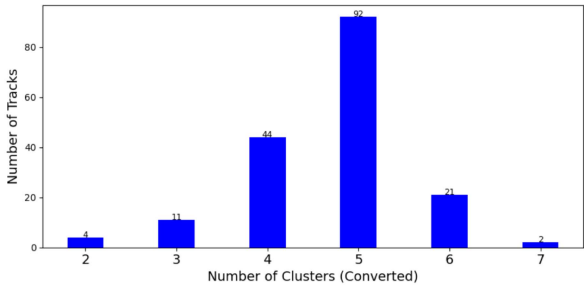
1) Statistics: The statistics of the annotations for the Beatles dataset are summarized in Fig. 3. The most frequent labels are `verse` and `refrain` because many songs take the verse-refrain form (Fig. 3(a)). The original annotations distinguish variations of a section (e.g., repetitions of `verse`) with extra markers (e.g., `verseA` and `verseB`). To standardize the dictionary of section classes, we removed the variation markers while keeping the number of sections unchanged. For example, given two consecutive sections, `verseA` and `verseB`, we converted them into the same class `verse` but preserved the boundaries

[1] The parameters of the peak-picking algorithm were configured based on a preliminary experiment using a small amount of data.

[2] [Online]. Available: https://pythonhosted.org/msaf/datasets.html
[3] [Online]. Available: https://archive.org/
[4] [Online]. Available: https://github.com/jblsmith/matching-salami
[5] [Online]. Available: https://github.com/DDMAL/salami-data-public

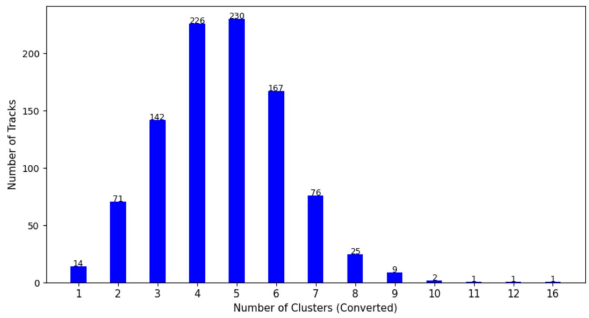


(a) Durations of section labels.

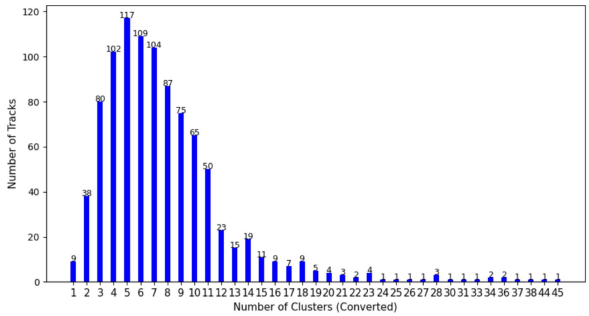


(b) The histogram of the numbers of classes.

Fig. 3. Statistics of structure annotations for the Beatles dataset.



(a) The number of classes at the coarse level.



(b) The number of classes at the fine level.

Fig. 4. Statistics of structure annotations for the SALAMI dataset.

between them. We explored the number of classes in each song and found that the most typical number of classes is 5 (Fig. 3(b)).

The statistics of the annotations for the SALAMI dataset are summarized in Fig. 4. Similarly, we converted the original semiotic labels by neglecting the repetition indicators. For instance, `A′` and `A″` were converted into the same class `A`. We found that around 96% of the songs comprise a maximum of 7 classes at the coarse level (Fig. 4(a)) and a maximum of 18 classes at the fine level (Fig. 4(b)).

2) Splitting: Considering that the two datasets have different musical characteristics (the SALAMI dataset comprises music tracks of various genres), we tested the proposed model on the two datasets separately. For the Beatles dataset, the 14 songs from the first album were used for evaluation, and the remaining 160 songs were used for training. As for the SALAMI dataset, the 555 tracks from YouTube were used for training and the 441 tracks from the Internet Archive were used for evaluation. Note that some tracks in the SALAMI dataset have two versions of annotations, and we used only the first version for evaluation if multiple annotations are available. For both datasets, we augmented the training data by applying pitch shifting (± 1 semitone) to each track.

B. Configurations

The audio signal of each track resampled at 32 kHz was analyzed using the short-time Fourier transform (STFT) and the constant-Q transform (CQT) with a Hann window of 3200 samples (100 ms) and a shifting interval of 1600 samples (50 ms). The mel spectrogram $\mathbf{X}_m = \{\mathbf{x}_{mn} \in \mathbb{R}^{80}\}_{n=1}^{N}$ was obtained by feeding the STFT spectrogram to the 80 mel filterbanks whose center frequencies were equally spaced on the mel-frequency scale from 80 Hz to 1600 Hz. The chromagram $\mathbf{X}_c = \{\mathbf{x}_{cn} \in \mathbb{R}^{12}\}_{n=1}^{N}$ was obtained by accumulating the amplitudes of the 12 pitch classes over 7 octaves (from C1 to B7) on the CQT-spectrogram. The temporgram $\mathbf{X}_t = \{\mathbf{x}_{tn} \in \mathbb{R}^{384}\}_{n=1}^{N}$ was obtained by analyzing the local autocorrelation of the onset strength envelope with a window size of 384 [8]. In addition, all features were downsampled by a factor of 10, resulting in a frame size of 500 ms.

The dimension of the latent features $\mathbf{E}$ and $\mathbf{H}$ was set to $D = 80$, the number of attention heads in the Transformer encoder was set to 8, and the number of EM iterations was set to $I = 10$. For the Beatles dataset, the number of classes was set to $K = 7$, which was equal to the maximum number of classes included in a track. For the SALAMI dataset, the number of classes was set to $K = 7$ for the coarse level and $K = 18$ for the fine level.

C. Compared Methods

To evaluate the proposed joint feature extraction and clustering method, we tested a cascading approach that extracts latent features $\mathbf{H}$ with our feature extraction model (denoted by **CAMHSA**) and then performs clustering on $\mathbf{H}$ with spectral clustering (**SC**) [1], variable Markov oracle (**VMO**) [2], or GMM [13]. When the GMM was used, the number of classes,

K, was either determined with BIC (denoted by **GMM-BIC**) for each track as proposed in [13] or fixed to $K = 7$ or 18 (denoted by **GMM-**K) as in the proposed method. Note that the feature extraction and clustering models were trained or performed independently, unlike the proposed method.

We also tested a modified version of SpecTNT [5], the state-of-the-art method for the classification-type MSA. Specifically, we extracted intermediate features from each of $\mathbf{X}_m$, $\mathbf{X}_c$, and $\mathbf{X}_t$ with a compact network comprising two convolutional layers and a residual connection [50], and then fed the concatenated intermediate features into the SpecTNT blocks. The final classification layer of the original SpecTNT was replaced with the proposed clustering model. Note that the feature extraction and clustering models (denoted by **SpecTNT-GMM**) were jointly trained as in the proposed method (**CAMHSA-GMM**).

D. Ablation Studies

To validate the design choices concerning the loss function, the inference strategy, and the input features, we tested ablated versions of the proposed method defined as follows:

- *Without* $\mathcal{L}_{\mathrm{H}}$*:* To validate the contrastive learning on the latent features $\mathbf{H}$, we tested a version obtained by ablating the feature extraction loss $\mathcal{L}_{\mathrm{H}}$ in (16).
- *Without* $\mathcal{L}_{\mathrm{B}}^{\mathrm{ld}}$ *or* $\mathcal{L}_{\mathrm{Z}}^{\mathrm{ld}}$*:* To validate the imbalance-aware learning of the boundaries $\mathbf{B}$ and the classes $\mathbf{Z}$, we tested versions obtained by ablating the logarithmic dice losses $\mathcal{L}_{\mathrm{B}}^{\mathrm{ld}}$ and $\mathcal{L}_{\mathrm{Z}}^{\mathrm{ld}}$ in (19) and (22), i.e., $\lambda_{\mathrm{B}} = 0$ and $\lambda_{\mathrm{Z}} = 0$, respectively.
- *Without Bound:* To validate the supervision of estimating the boundaries $\mathbf{B}$, we tested a version that did not use the information of $\mathbf{P}_{\mathrm{B}}$. In other words, both $\mathbf{B}$ and $\mathbf{Z}$ were estimated from the posterior probabilities $\mathbf{P}_{\mathrm{Z}}$ with the Viterbi algorithm[6] unlike the proposed method that detected $\mathbf{B}$ from the posterior probabilities $\mathbf{P}_{\mathrm{B}}$.
- *Oracle Bound:* To investigate the potential degradation of the clustering performance caused by segmentation errors, we tested an oracle version that used the ground-truth $\mathbf{B}$ for estimating $\mathbf{Z}$ with (15). This represented the upper limit of the clustering performance given the perfect segmentation.
- *Without* $\mathbf{X}_t$ *or* $\mathbf{X}_{\{c,t\}}$*:* To validate the integration of multiple types of acoustic features (i.e., the mel spectrogram $\mathbf{X}_m$, chromagram $\mathbf{X}_c$, and tempogram $\mathbf{X}_t$), we subsequently ablated $\mathbf{X}_t$ and $\mathbf{X}_c$ for the input representation.

E. Evaluation Measures

The MSA results were evaluated in terms of segmentation and clustering performances. To evaluate the estimated structure boundaries, we calculated the precision and recall rates with an error tolerance of ± 0.5 or ± 3 sec and their harmonic mean called the F-score [51]. To evaluate the performance of flat clustering at the frame level, we computed the precision ($\mathcal{P}$), recall ($\mathcal{R}$), and the F-score ($\mathcal{F}$) of the pairwise agreement [11], which are given by

$$\mathcal{P} = \frac{|A_{\mathrm{gt}} \cap A_{\mathrm{est}}|}{|A_{\mathrm{est}}|}, \tag{25}$$

$$\mathcal{R} = \frac{|A_{\mathrm{gt}} \cap A_{\mathrm{est}}|}{|A_{\mathrm{gt}}|}, \tag{26}$$

$$\mathcal{F} = \frac{2\mathcal{P}\mathcal{R}}{\mathcal{P} + \mathcal{R}}, \tag{27}$$

where $A_* \triangleq \{(i,j)|z_{*,i} = z_{*,j}\}$ $(* \in \{\mathrm{gt}, \mathrm{est}\})$ represents a set of frame pairs (i,j) with the same class in the ground-truth or estimated data and $|X|$ represents the cardinality of a set X. To evaluate the performance of hierarchical clustering, we also used a generalization of the pairwise agreement named the L-measure [52]. It deals with a set of frame triplets (i,j,k), where the frame pair (i,j) agrees at a deeper level than the pair (i,k). The precision and recall rates and the F-score were computed in the same way as the pairwise agreement. Note that the L-measure was applied to only the SALAMI dataset, where the hierarchical annotations were available.

These metrics have been used commonly in the literature [1], [4], [5], [12], [13], [15] for evaluating the clustering performance among other metrics such as the normalized conditional entropy scores [53] and the V-measures [54]. All the metrics are implemented in `mir_eval` package [55], and we used the default parameters unless otherwise specified.

F. Results on the Beatles Dataset

The performances of the six methods on the Beatles dataset are summarized in Table I. The proposed **CAMHSA-GMM** worked best in terms of the segmentation performance and the F-score of the pairwise agreement. It outperformed the four cascading methods in almost all metrics. This indicates that our joint training strategy could benefit the standard framework of MSA. Although the sequential combination of **CAMHSA** and **GMM** was architecturally the same as **CAMHSA-GMM**, a significant performance difference was observed. This could be attributed to the convergence issue of the GMM. During inference, the GMM of the cascading method was trained based on the maximum likelihood estimation, which would lead to poor convergence when the number of components was larger than the true number of classes. In contrast, the GMM of our method used only a fixed number of forward steps and thus was not subject to the convergence process.

Comparing **CAMHSA-GMM** and **SpecTNT-GMM**, we confirmed the effectiveness of the CAMHSA mechanism over the vanilla self-attention mechanism in feature extraction. Although both CAMHSA and SpecTNT involve temporal self-attention, the augmented convolutions of CAMHSA captured the dependencies of multiple attention maps and thus contributed to performance improvement. In **SpecTNT-GMM**, the low precision (0.562) and high recall (0.937) of the pairwise agreement indicated that the representations generated by the SpecTNT were indiscriminable to the clustering layer, as exemplified in Fig. 5.

[6] TensorFlow API: tfa.text.viterbi_decode.

TABLE I
COMPARATIVE EVALUATION ON THE BEATLES DATASET

Feature extraction	Feature clustering	Segmentation (0.5) P	Segmentation (0.5) R	Segmentation (0.5) F	Segmentation (3) P	Segmentation (3) R	Segmentation (3) F	Pairwise agreement P	Pairwise agreement R	Pairwise agreement F
CAMHSA	SC	0.654	0.526	0.578	0.826	0.676	0.737	0.586	0.547	0.544
CAMHSA	VMO	0.654	0.526	0.578	0.826	0.676	0.737	0.657	0.454	0.528
CAMHSA	GMM-BIC	0.662	0.607	0.628	0.845	0.788	0.809	0.774	0.571	0.640
CAMHSA	GMM-7	0.662	0.607	0.628	0.845	0.788	0.809	**0.780**	0.593	0.657
SpecTNT-GMM		0.593	0.558	0.568	0.697	0.666	0.673	0.562	**0.937**	0.689
CAMHSA-GMM		**0.662**	**0.607**	**0.628**	**0.845**	**0.788**	**0.809**	0.699	0.867	**0.758**

TABLE II
ABLATION STUDY ON THE BEATLES DATASET

	Segmentation (0.5) P	Segmentation (0.5) R	Segmentation (0.5) F	Segmentation (3) P	Segmentation (3) R	Segmentation (3) F	Pairwise agreement P	Pairwise agreement R	Pairwise agreement F
w/o $\mathcal{L}_{\mathrm{Z}}^{\mathrm{ld}}$	0.611	0.604	0.602	0.738	0.743	0.733	0.583	0.887	0.696
w/o $\mathcal{L}_{\mathrm{B}}^{\mathrm{ld}}$	0.585	0.438	0.480	0.746	0.579	0.625	0.640	0.764	0.677
w/o $\mathcal{L}_{\mathrm{H}}$	**0.660**	0.586	**0.617**	**0.817**	0.729	0.765	0.659	0.898	0.754
w/o $\mathbf{X}_t$	0.588	**0.634**	0.607	0.742	0.808	**0.770**	0.712	0.733	0.709
w/o $\mathbf{X}_{\{c,t\}}$	0.618	0.606	0.606	0.725	0.707	0.709	0.552	0.811	0.644
w/o bound	0.187	0.510	0.270	0.309	**0.824**	0.443	0.687	0.461	0.548
Oracle bound	-	-	-	-	-	-	**0.773**	**0.923**	**0.825**

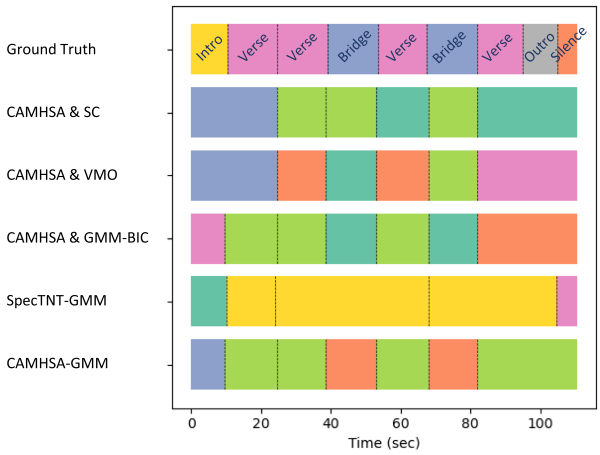


Fig. 5. Music structures estimated by the five methods for the song "Misery" by the Beatles.

The results of the ablation study are summarized in Table II. Using hybrid losses provided a significant performance gain compared to using individual cross entropy losses. The segmentation F-score was improved compared to **Without** $\mathcal{L}_{\mathrm{B}}^{\mathrm{ld}}$ (0.628/0.809vs. 0.480/0.625), and the F-score of the pairwise agreement was improved compared to **Without** $\mathcal{L}_{\mathrm{Z}}^{\mathrm{ld}}$ (0.758vs. 0.696). Moreover, the performance was boosted compared to **Without** $\mathcal{L}_{\mathrm{H}}$, especially in terms of the segmentation F-scores (0.628/0.809vs. 0.617/0.765). In addition, performance degradation was observed in both **Without** $\mathbf{X}_t$ and **Without** $\mathbf{X}_{\{c,t\}}$, revealing the importance of exploiting multiple types of features. Finally, the segmentation F-score dropped severely in **Without Bound** (0.628/0.809 vs. 0.270/0.443), showing that the extra boundary prediction was beneficial to the MSA task. In particular, the results obtained by **Oracle Bound** implied that accurate boundary prediction could further improve the clustering performance.

G. Results on the SALAMI Dataset

The performances of the six methods on the SALAMI dataset are summarized in Table III. The proposed method outperformed the other methods at the fine level in terms of the F-scores, although such superiority was not clear at the coarse level. Considering that the number of ground-truth boundaries at the coarse level was less than that at the fine level, the deep learning-based methods (**SpecTNT-GMM** and **CAMHSA-GMM**) might have suffered from the extremely imbalanced training data and thus obtained lower segmentation performances than the cascading methods. Nevertheless, the cascade of **CAMHSA** and **GMM-BIC**, which can be regarded as a two-stage alternative to the proposed method, obtained better performances at the coarse level (in terms of the F-scores of segmentation and the L-measure), indicating that our approach has a great potential to outperform the cascading methods by leveraging data augmentation techniques.

Regarding the loss functions, the use of the hybrid losses had a positive impact on the model performance, especially the L-measure, as shown in Table IV(a). Compared with **Without** $\mathbf{X}_t$ and **Without** $\mathbf{X}_{\{c,t\}}$, employing multiple types of acoustic features was crucial to the segmentation task. In comparison to **Without Bound**, the joint boundary estimation gained consistent improvements in the F-score of the pairwise agreement at both the coarse and fine levels. Nonetheless, as shown in Table IV(b), the use of ground-truth boundary annotations did not provide a better clustering performance at the fine level.

TABLE III
COMPARATIVE EVALUATION ON THE SALAMI DATASET

(a) Coarse level

Feature extraction	Feature clustering	Segmentation (0.5)			Segmentation (3)			Pairwise agreement			L-measure		
		P	R	F	P	R	F	P	R	F	P	R	F
CAMHSA	SC	**0.397**	0.592	**0.460**	**0.517**	0.763	**0.598**	0.525	0.553	0.507	0.347	**0.429**	0.376
CAMHSA	VMO	0.397	0.592	0.460	0.517	0.763	0.598	0.501	0.462	0.451	0.378	0.330	0.346
CAMHSA	GMM-BIC	0.325	0.703	0.431	0.407	0.878	0.540	**0.605**	0.437	0.482	**0.387**	0.428	**0.398**
CAMHSA	GMM-7	0.325	0.703	0.431	0.407	0.878	0.540	0.595	0.407	0.457	0.381	0.381	0.374
SpecTNT-GMM		0.365	0.533	0.418	0.483	0.700	0.552	0.437	**0.909**	**0.563**	0.298	0.283	0.274
CAMHSA-GMM		0.325	**0.703**	0.431	0.407	**0.878**	0.540	0.498	0.717	0.561	0.330	0.397	0.352

(b) Fine level

Feature extraction	Feature clustering	Segmentation (0.5)			Segmentation (3)			Pairwise agreement			L-measure		
		P	R	F	P	R	F	P	R	F	P	R	F
CAMHSA	SC	**0.658**	0.307	0.393	**0.908**	0.427	0.548	0.506	0.350	0.375	-	-	-
CAMHSA	VMO	0.658	0.307	0.393	0.908	0.427	0.548	**0.654**	0.190	0.276	-	-	-
CAMHSA	GMM-BIC	0.525	0.638	0.544	0.717	0.875	0.746	0.538	0.332	0.377	-	-	-
CAMHSA	GMM-18	0.525	0.638	0.544	0.717	0.875	0.746	0.562	0.195	0.270	-	-	-
SpecTNT-GMM		0.649	0.293	0.381	0.889	0.403	0.525	0.398	**0.715**	0.474	-	-	-
CAMHSA-GMM		0.525	**0.638**	**0.544**	0.717	**0.875**	**0.746**	0.421	0.653	**0.475**	-	-	-

TABLE IV
ABLATION STUDY ON THE SALAMI DATASET

(a) Coarse level

Method	Segmentation (0.5)			Segmentation (3)			Pairwise agreement			L-measure		
	P	R	F	P	R	F	P	R	F	P	R	F
w/o $\mathcal{L}_{\mathrm{Z}}^{\mathrm{ld}}$	0.301	0.697	0.408	0.382	**0.880**	0.517	0.438	0.873	0.554	0.283	0.235	0.232
w/o $\mathcal{L}_{\mathrm{B}}^{\mathrm{ld}}$	0.984	0.220	0.348	**0.984**	0.220	0.348	0.417	0.999	0.565	0.292	0.015	0.021
w/o $\mathcal{L}_{\mathrm{H}}$	0.353	0.669	**0.440**	0.441	0.829	**0.548**	0.416	**1.000**	0.564	0.233	0.187	0.192
w/o $\mathbf{X}_t$	0.929	0.221	0.346	0.938	0.223	0.349	0.416	1.000	0.565	0.311	0.025	0.031
w/o $\mathbf{X}_{\{c,t\}}$	0.623	0.248	0.339	0.770	0.311	0.424	0.416	1.000	0.564	0.318	0.123	0.140
w/o bound	0.259	0.489	0.259	0.349	0.663	0.355	0.486	0.656	0.532	0.300	0.383	0.329
Oracle bound	-	-	-	-	-	-	**0.534**	0.800	**0.613**	**0.350**	**0.421**	**0.376**

(b) Fine level

Method	Segmentation (0.5)			Segmentation (3)			Pairwise agreement			L-measure		
	P	R	F	P	R	F	P	R	F	P	R	F
w/o $\mathcal{L}_{\mathrm{Z}}^{\mathrm{ld}}$	0.497	0.676	0.542	0.670	**0.912**	0.733	0.388	0.746	0.471	-	-	-
w/o $\mathcal{L}_{\mathrm{B}}^{\mathrm{ld}}$	**0.812**	0.075	0.129	0.904	0.081	0.141	0.359	**0.992**	0.496	-	-	-
w/o $\mathcal{L}_{\mathrm{H}}$	0.549	**0.688**	**0.577**	0.710	0.886	**0.745**	0.376	0.744	0.461	-	-	-
w/o $\mathbf{X}_t$	0.802	0.076	0.130	**0.905**	0.083	0.144	0.360	0.981	0.494	-	-	-
w/o $\mathbf{X}_{\{c,t\}}$	0.648	0.090	0.147	0.872	0.122	0.202	0.376	0.915	**0.497**	-	-	-
w/o bound	0.276	0.473	0.258	0.419	0.643	0.377	0.406	0.608	0.452	-	-	-
Oracle bound	-	-	-	-	-	-	**0.426**	0.665	0.484	-	-	-

This implies that the extracted fine-level features might be indistinctive for clustering.

H. Observation

Compared with the evaluation results on the Beatles dataset, the lower performances on the SALAMI dataset reflected the diversity of the audio tracks in terms of music genre and sound quality. Nevertheless, we observed several consistent effects on both datasets. First, the standard cascading approaches of MSA benefited from our joint clustering framework. Second, the proposed method worked better than **SpecTNT-GMM** on average, highlighting the effectiveness of the CAMHSA mechanism in capturing structure-related features. Moreover, the auxiliary dice losses (i.e., $\mathcal{L}_{\mathrm{B}}^{\mathrm{ld}}$ and $\mathcal{L}_{\mathrm{Z}}^{\mathrm{ld}}$) provided performance gains in the segmentation and clustering tasks, respectively. Finally, the extra boundary prediction in the clustering layer not only smoothed the outcome of the frame-level clustering but also improved the F-score of the pairwise agreement.

I. Cross-Dataset Evaluation

To further investigate the generalization capability of the proposed method, we performed a cross-dataset evaluation where the training data of the Beatles and SALAMI datasets (Section V-A2) were swapped. Only the course-level evaluation

TABLE V
CROSS-DATASET EVALUATION

Training	Test	Segmentation (0.5)			Segmentation (3)			Pairwise agreement		
		P	R	F	P	R	F	P	R	F
SALAMI	Beatles	0.620	0.489	0.539	0.751	0.591	0.653	0.434	0.820	0.556
Beatles	SALAMI	0.431	0.496	0.443	0.576	0.663	0.593	0.391	0.996	0.539

was conducted in this scenario because the Beatles dataset has no fine-level annotations. The evaluation results are summarized in Table V. When the proposed method was trained on the SALAMI dataset and tested on the Beatles dataset, both the segmentation and clustering performances were degraded, compared to those obtained in the within-dataset evaluation (Table I). In contrast, when the proposed method was trained on the Beatles dataset and tested on the SALAMI dataset, the segmentation performance was improved slightly, compared to the within-dataset counterpart (Table III(a)). This could be partly explained by the facts that the SALAMI training data contain audio tracks of various genres including classical music, which is especially challenging for the MSA task, and that the SALAMI test data mainly comprise audio tracks of live performances in noisy recording environments. Further study would thus be required to improve the robustness of the proposed method against diverse music genres and recording environments.

VI. CONCLUSION

This paper presented a deep learning framework that jointly optimizes the feature extraction and clustering stages of MSA. We used a GMM with a deep unfolding technique for soft clustering and computed a contrastive loss for supervising the clustering results. The comprehensive investigation demonstrated an improvement over two-stage approaches in the performance of frame-level segmentation and clustering. In particular, the proposed model showed its superiority over the baseline methods on the Beatles dataset. The experiment results also indicated that accurate estimation of the section boundaries can boost the clustering performance significantly. In this regard, we could integrate a sophisticated segmentation method into the proposed system and take into account other elements that are closely related to the identification of musical boundaries such as harmony and rhythm.

The main limitation of the proposed model is that a heuristic postprocessing stage is required to accommodate the section boundaries and classes estimated separately with the two branches, making the clustering result suboptimal. To overcome this limitation, we plan to design a joint segmentation and clustering method with a consistency guarantee. For further performance improvement, we could use the hidden Markov model (HMM) instead of the GMM as the clustering module to favor the temporal smoothness of section classes. For hierarchical (multi-level) structure analysis, a novel peak-picking strategy that takes more than one level as input would be required to obtain a reliable consistent segmentation.

We took a clustering rather than a classification approach to MSA due to its flexibility in analyzing diverse music data. This enabled us to analyze music structure from a macro perspective without being limited by vocabulary and music genre. Such a process might provide interesting and valuable insights into the universal construction of music than may be observed otherwise within a certain music style. We hope that our research will draw more attention to music structure and encourage deep learning techniques for MSA.

REFERENCES

[1] B. McFee and D. Ellis, "Analyzing song structure with spectral clustering," in *Proc. 15th Int. Soc. Music Inf. Retrieval Conf.*, 2014, pp. 405–410.

[2] C. Wang and G. J. Mysore, "Structural segmentation with the Variable Markov Oracle and boundary adjustment," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 291–295.

[3] C. J. Tralie and B. McFee, "Enhanced hierarchical music structure annotations via feature level similarity fusion," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 201–205.

[4] G. Shibata, R. Nishikimi, and K. Yoshii, "Music structure analysis based on an LSTM-HSMM hybrid model," in *Proc. 21st Int. Soc. Music Inf. Retrieval Conf.*, 2020, pp. 23–29.

[5] J. Wang, Y. Hung, and J. B. L. Smith, "To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 416–420.

[6] S. Stevens, J. Stanley, J. Volkman, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," in *J. Acoust. Soc. Amer.*, vol. 8, pp. 185–190, 1937.

[7] T. Fujishima, "Realtime chord recognition of musical sound: A system using common Lisp music," in *Proc. Int. Comput. Music Conf.*, 1999, pp. 464–467.

[8] P. Grosche, M. Müller, and F. Kurth, "Cyclic tempogram–A mid-level tempo representation for music signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 5522–5525.

[9] R. J. Weiss and J. P. Bello, "Unsupervised discovery of temporal structure in music," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1240–1251, Oct. 2011.

[10] O. Nieto and T. Jehan, "Convex non-negative matrix factorization for automatic music structure identification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 236–240.

[11] M. Levy and M. B. Sandler, "Structural segmentation of musical audio by constrained clustering," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 2, pp. 318–326, Feb. 2008.

[12] M. Buisson, B. McFee, S. Essid, and H. C. Crayencour, "Learning multi-level representations for hierarchical music structure analysis," in *Proc. 23rd Int. Soc. Music Inf. Retrieval Conf.*, 2022, pp. 591–597.

[13] T. Chen, L. Su, and K. Yoshii, "Learning multifaceted self-similarity for musical structure analysis," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2023, pp. 165–172.

[14] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2000, pp. 452–455.

[15] T. Cheng, J. B. L. Smith, and M. Goto, "Music structure boundary detection and labelling by a deconvolution of path-enhanced self-similarity matrix," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 106–110.

[16] R. Chen and M. Li, "Music structural segmentation by combining harmonic and timbral information," in *Proc. 12th Int. Soc. Music Inf. Retrieval Conf.*, 2011, pp. 477–482.

[17] H. Grohganz, M. Clausen, N. Jiang, and M. Müller, "Converting path structures into block structures using eigenvalue decompositions of self-similarity matrices," in *Proc. 14th Int. Soc. Music Inf. Retrieval Conf.*, 2013, pp. 209–214.

[18] F. Kaiser and G. Peeters, "A simple fusion method of state and sequence segmentation for music structure discovery," in *Proc. 14th Int. Soc. Music Inf. Retrieval Conf.*, 2013, pp. 257–262.
[19] B. McFee and D. P. W. Ellis, "Learning to segment songs with ordinal linear discriminant analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 5197–5201.
[20] G. Yang, C. Tuan, H. Lee, and L. Lee, "Improved speech separation with time-and-frequency cross-domain joint embedding and clustering," in *Proc. 20th Annu. Conf. Int. Speech*, 2019, pp. 1363–1367.
[21] D. Zhang et al., "Supporting clustering with contrastive learning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2021, pp. 5419–5430.
[22] Y. Li, P. Hu, J. Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 8547–8555.
[23] M. Ronen, S. E. Finder, and O. Freifeld, "DeepDPM: Deep clustering with an unknown number of clusters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9851–9860.
[24] V. Gupta, H. Shi, K. Gimpel, and M. Sachan, "Deep clustering of text representations for supervision-free probing of syntax," in *Proc. 36th AAAI Conf. Artif. Intell.*, 2022, pp. 10720–10728.
[25] K. Kinoshita, M. Delcroix, and T. Iwata, "Tight integration of neural- and clustering-based diarization through deep unfolding of infinite Gaussian mixture model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 8382–8386.
[26] A. Vaswani et al., "Attention is all you need," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[27] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," Mitsubishi Elect. Res. Lab., Tech. Rep. TR2014-117, 2014.
[28] C. Lee and J. Chien, "Deep unfolding inference for supervised topic model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 2279–2283.
[29] J. R. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 31–35.
[30] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. D. Reid, "Deep subspace clustering networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 24–33.
[31] Y. Jiao, N. Xie, Y. Gao, C. Wang, and Y. Sun, "Fine-grained fashion representation learning by online deep clustering," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 19–35.
[32] O. Nieto et al., "Audio-based music structure analysis: Current trends, open challenges, and applications," *Trans. Int. Soc. Music Inf. Retrieval*, vol. 3, no. 1, pp. 246–263, 2020.
[33] W. T. Lu, J. Wang, M. Won, K. Choi, and X. Song, "SpecTNT: A time-frequency transformer for music audio," in *Proc. 22nd Int. Soc. Music Inf. Retrieval Conf.*, 2021, pp. 396–403.
[34] O. Nieto and J. P. Bello, "Systematic exploration of computational music structure research," in *Proc. 17th Int. Soc. Music Inf. Retrieval Conf.*, 2016, pp. 547–553.
[35] M. Buisson, B. McFee, S. Essid, and H. C. Crayencour, "Self-supervised learning of multi-level audio representations for music segmentation," *IEEE ACM Trans. Audio, Speech Lang. Process.*, vol. 32, pp. 2141–2152, 2024.
[36] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.
[37] P. Khosla et al., "Supervised contrastive learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 18661–18673.
[38] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," in *J. Roy. Stat. Soc.: Ser. B*, vol. 39, 1977, pp. 1–38.
[39] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds., 2014.
[40] D. Yu, M. Kolbæk, Z. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 241–245.
[41] M. Kolbaek, D. Yu, Z. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 10, pp. 1901–1913, Oct. 2017.
[42] X. Liu and J. Pons, "On permutation invariant training for speech source separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 6–10.
[43] X. Ji, A. Vedaldi, and J. F. Henriques, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9864–9873.
[44] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, vol. 119, pp. 1597–1607.
[45] O. J. Hénaff, "Data-efficient image recognition with contrastive predictive coding," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, vol. 119, pp. 4182–4192.
[46] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, vol. 12356, pp. 776–794.
[47] K. C. L. Wong, M. Moradi, H. Tang, and T. F. Syeda-Mahmood, "3D segmentation with exponential logarithmic loss for highly unbalanced object sizes," in *Proc. 21st Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2018, pp. 612–619.
[48] M. Mauch et al., "OMRAS2 metadata project 2009," in *Proc. 10th Int. Soc. Music Inf. Retrieval Conf. - Late-Breaking Session*, 2009, Art. no. 1.
[49] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. D. Roure, and J. S. Downie, "Design and creation of a large-scale database of structural annotations," in *Proc. 12th Int. Soc. Music Inf. Retrieval Conf.*, 2011, pp. 555–560.
[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[51] D. Turnbull, G. R. G. Lanckriet, E. Pampalk, and M. Goto, "A supervised approach for detecting boundaries in music using difference features and boosting," in *Proc. 8th Int. Conf. Music Inf.*, 2007, pp. 51–54.
[52] B. McFee, O. Nieto, M. M. Farbood, and J. P. Bello, "Evaluating hierarchical structure in music annotations," *Front. Psychol.*, vol. 8, 2017, Art. no. 1337.
[53] H. M. Lukashevich, "Towards quantitative measures of evaluating song segmentation," in *Proc. 9th Int. Conf. Music Inf. Retrieval*, 2008, pp. 375–380.
[54] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2007, pp. 410–420.
[55] C. Raffel et al., "mir_eval: A transparent implementation of common MIR metrics," in *Proc. 15th Int. Soc. Music Inf. Retrieval Conf.*, 2014, pp. 367–372.

Tsung-Ping Chen received the M.S. degree in musicology from National Taiwan University, Taipei, Taiwan, in 2013. He is currently working toward the Ph.D. degree with the Graduate School of Informatics, Kyoto University, Kyoto, Japan. His research interests include music information retrieval and machine learning.

Kazuyoshi Yoshii (Senior Member, IEEE) received the M.S. and Ph.D. degrees in informatics from Kyoto University, Kyoto, Japan, in 2005 and 2008, respectively. He is currently a Professor with the Graduate School of Engineering, Kyoto University, and concurrently, the Leader with the Sound Scene Understanding Team, Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan. His research interests include music informatics, audio signal processing, and statistical machine learning.