Learning Multi-Level Representations for Hierarchical Music Structure Analysis

Morgan Buisson, Brian Mcfee, Slim Essid, Helene-Camille Crayencour

LEARNING MULTI-LEVEL REPRESENTATIONS FOR HIERARCHICAL MUSIC STRUCTURE ANALYSIS

Morgan Buisson[1] **Brian McFee**[2,3] **Slim Essid**[1] **Hélène C. Crayencour**[4]
[1] LTCI, Télécom Paris, Institut Polytechnique de Paris, France
[2] Music and Audio Research Laboratory, New York University, USA
[3] Center of Data Science, New York University, USA
[4] L2S, CNRS-Univ.Paris-Sud-CentraleSupélec, France

ABSTRACT

Recent work in music structure analysis has shown the potential of deep features to highlight the underlying structure of music audio signals. Despite promising results achieved by such representations, dealing with the inherent hierarchical aspect of music structure remains a challenging problem. Because different levels of segmentation can be considered as equally valid, specifically designed representations should be optimized to improve hierarchical structure analysis. In this work, unsupervised learning of such representations using a contrastive approach operating at different time-scales is explored. The proposed system is evaluated on flat and multi-level music segmentation. By leveraging both time and the hierarchical organization of music structure, we show that the obtained deep embeddings can encode meaningful patterns and improve segmentation at various levels of granularity.

1. INTRODUCTION

Common approaches for music structure analysis can usually be broken down into two main steps: segmentation and structural grouping [1]. The segmentation task aims at determining the boundary locations between consecutive musical sections while the grouping step consists in assigning labels to each of the retrieved segments based on certain musical similarities. Traditional algorithms for structural segmentation use different hand-crafted features [2] and their combinations to detect abrupt changes of particular musical characteristics or repetitions of certain patterns throughout the song. However, recent progress in deep learning has given rise to new systems automatically producing more robust representations which manage to combine several acoustic characteristics to enhance the recognition of musical sections [3–5]. While these representations have consistently improved downstream segmentation methods, their performance is mostly evaluated on *flat* structural annotations and metrics. However, musical structure naturally exhibits a hierarchical organization where a variety of cues can trigger boundaries between segments of different length [6], depending on the time scale at which they are observed [1]. At the lowest temporal level, short segments might only last a few measures. Coarser annotation levels are generally composed of longer segments, grouping various shorter fragments into larger musically meaningful units (ex: chorus, verse ...). This nested organization of musical events at different levels holds crucial information about music structure [7]. While the original task of music structure analysis is commonly performed at a pre-defined level of granularity (i.e. *flat* segmentation), the problem of *hierarchical* structural analysis consists in predicting a set of segmentation candidates called hierarchy, ordered by their amount of detail (from the coarsest to the most refined level). Recent efforts have been made to compile datasets with multi-level structural annotations [2, 8], which greatly facilitates the study of musical structure in a hierarchical manner. Although a few methods have been proposed for such task, the role of hierarchy in music structure has never been explicitly considered while building better-suited representations of audio music signals prior to segmentation.

1.1 Our contributions

In this work, we propose a deep unsupervised hierarchical metric learning approach for music structure analysis. We show that leveraging both time information and the hierarchical structure of music can help building efficient representations for music segmentation at different levels without requiring any supervision from structural annotations. We demonstrate the effectiveness of these representations for both flat and multi-level segmentation and show that they can accommodate structural annotations of varying styles and levels.

1.2 Related work

The method proposed here builds upon recent work in music structure analysis devoted to finding efficient representations using deep learning methods to improve already existing downstream algorithms. The work by McCallum [3] proposes an unsupervised method to learn deep features

using a triplet-based approach. It relies on the assumption that frames temporally close to each other are more likely to belong to the same musical section than those separated by a certain amount of time. Therefore, triplets are sampled in such a way that the temporal distance between the anchor point and the positive example is smaller than the distance separating the anchor from the negative example. Wang et al. [4] adopt a similar approach by using structural annotations in a supervised fashion to mine informative sets of frames.

One of the main challenges in estimating the structure of a musical piece is to account for the different temporal levels at which it can be decomposed. Up to now, only a few approaches have been proposed for the task of multi-level segmentation. McFee and Ellis [9] use spectral clustering to decompose an enhanced self-similarity matrix and produce segmentations at different temporal levels. This approach is later improved by Tralie and McFee [10] where the input self-similarity matrix is obtained by combining different features using Similarity Network Fusion. Salamon et al. [5] further extend this method by employing two types of deep embeddings along with CQT features. They capture local timbral patterns with few shot-learning and long-term similarities with disentangled deep metric learning [11]. While these works demonstrate the advantage of combining multiple representations of a same signal to extract meaningful structural patterns, our approach shows instead that these can be directly encoded into the representations using time proximity and the hierarchy of music structure.

2. HIERARCHICAL REPRESENTATIONS

The method introduced here constructs deep representations which allow for structural segmentations at various time-scales. To facilitate the decomposition of a song at different levels, these representations should provide strong discriminative capabilities for time frames belonging to different musical sections and separated by a large amount of time. Conversely, they should be more homogeneous for frames belonging to the same section and happening within a short time interval. As section lengths might vary from one annotator to another due to the ambiguity of the task [1], the aforementioned constraint is imposed at different temporal scales. Additionally, most datasets for music structure analysis come with only one level of annotations, which motivates us to learn such representations in an unsupervised fashion, taking advantage of large quantities of unlabelled data. A base convolutional neural network is used to output embeddings which are divided into multiple sub-regions. Each of them is optimized independently using specific triplets of frames efficiently sampled to encode the temporal structure of the song at different levels. We show that each level of the final representations can model frames proximity with its own amount of granularity.

2.1 Sampling

The objective of the sampling method introduced by McCallum [3] is to build triplets of frames where the anchor and the positive example belong to the same musical section, while the anchor and the negative example are labelled differently. The method proposed here can be viewed as its multi-level extension. A hierarchy is defined as a set of L levels of structural segmentations ordered from the coarsest to the most refined. For each level $\ell \in \{0; \dots; L-1\}$ in the hierarchy, triplets of beat indices are sampled using a specific set of parameters $\delta = \{\delta^\ell_{p,min}, \delta^\ell_{p,max}, \delta^\ell_{n,min}, \delta^\ell_{n,max}\}$. Intuitively, they rule how "close" or "far away" from the anchor the positive and negative examples will be sampled throughout the song. More specifically, for a given anchor beat index i_a, positive and negative examples respectively located at beat indices i_p and i_n are uniformly sampled from the interval I^ℓ_p defined by $\delta^\ell_{p,min}$ and $\delta^\ell_{p,max}$ and I^ℓ_n specified by $\delta^\ell_{n,min}$ and $\delta^\ell_{n,max}$. An example for an arbitrary level ℓ is shown in Figure 1. The δ parameters actually define a notion of temporal distance d_ℓ between frames at a given level of the hierarchy (analogous to a notion of dissimilarity). Therefore, the set of triplets T_ℓ at level ℓ can be expressed as:

$$T_\ell = \{(i_a, i_p, i_n; l) \mid d_\ell(x_a, x_p) < d_\ell(x_a, x_n)\} \quad (1)$$

where x_k is the input feature patch observed at beat index i_k, $i^\ell_p \sim \mathcal{U}(I^\ell_p)$ and $i^\ell_n \sim \mathcal{U}(I^\ell_n)$.

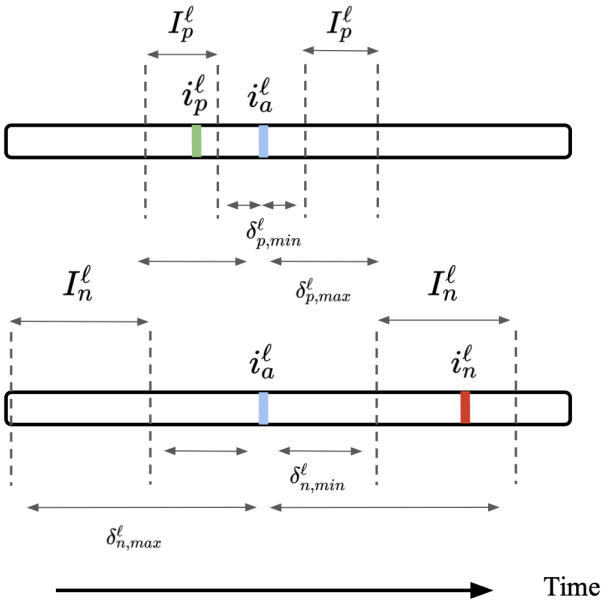

Figure 1. Initial triplet sampling method at level ℓ.

In order for the learned hierarchy levels to remain consistent with one another, monotonicity is encouraged by modifying the initial triplet mining technique. In addition to the time constraint imposed on triplets of the same level, their probability of being sampled is restricted from one level in the hierarchy to the next. For a randomly sampled anchor index at level $\ell = 0$, a complete triplet (i^a_0, i^p_0, i^n_0) is built only using time proximity (*i.e.* δ parameters). Then

for each level $\ell \in \{1; \ldots; L-1\}$, the positive example is sampled closer and closer to the same anchor (*i.e.* $\delta^{\ell}_{p,min}$ and $\delta^{\ell}_{p,max}$ decrease), whereas the negative is obtained by selecting the positive example from level $\ell - 1$. This way, going deeper into the hierarchy means that the representations get more refined to detect short-term musical patterns. The modified sampling method is summarized in Figure 2. The process is then repeated by starting over from level 0, going down the hierarchy with the same anchor index, transferring the negative example from the current to the next level, and uniformly sampling the positive ones using the right δ parameters. At the end, the whole training set for all levels of the hierarchy is given by combining every set of triplets T_ℓ level-wise: $T = \{T_\ell\}_{\ell=0}^{L-1}$.

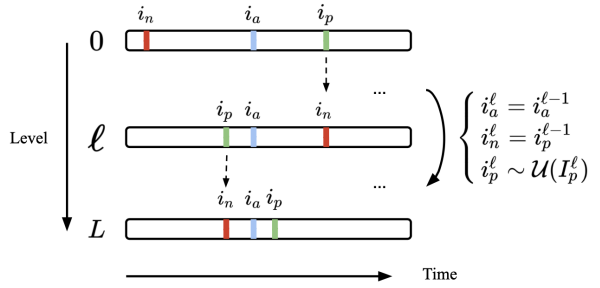

Figure 2. Modified triplet sampling, moving downwards in the hierarchy.

2.2 Disentangled hierarchy levels

During training, the model is shown triplets sampled at different hierarchy levels and should optimize the corresponding sub-regions of the output embeddings. We adapt the method introduced by Veit et al. [12], called Conditional Similarity Networks. This method has already proven to be efficient in the context of multi-dimensional music similarity learning [11], where a joint model learns compact representations of music audio signals complying with different similarity criteria, namely genre, mood, instrumentation and tempo. We propose to extend it to the hierarchical case: to model the different temporal distances d_ℓ, a set of L masking functions $m_\ell \in \{0,1\}^n$ that are applied to the embedding space of size n is defined. Each mask can be interpreted as an element-wise gating function selecting the relevant dimensions of the embedding corresponding to a particular level of the hierarchy. For a given triplet (x_a, x_p, x_n) at level ℓ, the training objective becomes:

$$\mathcal{L}(x_a, x_p, x_n) = [D_\ell(x_a, x_p) - D_\ell(x_a, x_n) + \alpha]_+, \quad (2)$$

$$D_\ell(x_i, x_j) = \| \, m_\ell \circ [f(x_i) - f(x_j)] \, \|_2^2 \quad (3)$$

where $\circ$ is the Hadamard product, $[.]_+$ denotes the Hinge loss, α the margin parameter and $f(x)$ is the projection of x into the embedding space by the convolutional neural network. An example is illustrated in Figure 3, where $L = 3$ and $\ell = 1$. Since going deeper into the hierarchy results in triplets of frames getting temporally closer to each other, it is unnecessary for the model to separate samples by the same distance margin at all levels. Therefore, margin values were evenly distributed within the range $[0.05, 0.1]$ so that for each level $\ell \in \{0; \ldots; L-2\}$, we have $\alpha_\ell > \alpha_{\ell+1}$.

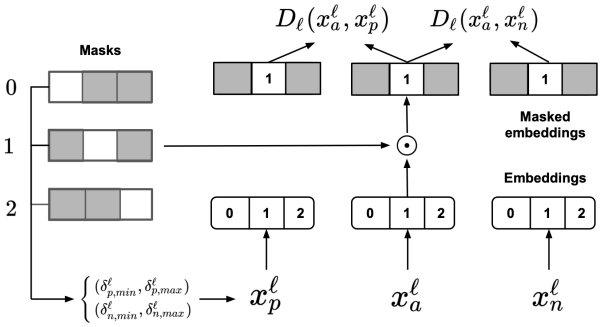

Figure 3. Training pipeline for $\ell = 1$ and $L = 3$. At each iteration, the current hierarchy level defines the set of δ parameters to sample the positive example. The mask here conserves the sub-region corresponding to level $\ell = 1$.

3. EXPERIMENTS

The evaluation of our method is divided into three distinct parts. First, we consider the problem of boundary detection on flat annotations using the SALAMI dataset. Second, we verify if the learned hierarchical representations improve multi-level segmentation predictions on that same dataset using the two-level structural annotations available. We finally demonstrate the flexibility of our approach and provide additional results on other commonly used datasets for music structure analysis where their original flat annotations have been automatically expanded beforehand [13].

3.1 Datasets

We use five different datasets in our evaluation:

SALAMI: the Structural Annotations for Large Amounts of Music Information (SALAMI) [8] is the most substantial dataset for music structure analysis. It contains $1,359$ tracks ranging from classical, jazz, popular to world and live music. Each track is provided with two levels of structural annotations. We use a subset of 884 songs labelled by two different annotators. Therefore, for each track contained in this subset, we end up with a total of 4 segmentation ground-truths (2 annotators $\times$ 2 levels of granularity). In the rest of this work, this subset is referred as SALAMI.

BeatlesTUT: a revised version of 174 annotated Beatles songs, originally released in the Isophonics dataset [14] and corrected by researchers from Tampere University of Technology.

RWC-Pop: the Popular subset of the RWC dataset [15] contains 100 songs with section annotations. Note that two versions of these annotations are available online; here the ones originally provided by the authors (AIST) are used.

RWC-Jazz: the Jazz subset of the RWC dataset [15] is composed of 50 songs from various Jazz sub-genres such as Vocal, Big Band, Modal, Funky, Free or Fusion Jazz.

JAAH: the Audio-aligned jazz harmony dataset (JAAH) [16] is composed of 113 tracks selected from "The Smithsonian Collection of Classic Jazz" and "Jazz: The Smithsonian Anthology", covering various performers, sub-genres and historical periods.

3.1.1 Obtaining multi-level annotations:

For all datasets but SALAMI, we apply automatic hierarchy expansion [13] before evaluating multi-level segmentation. As can be seen in the descriptive statistics from Table 1, both the distributions of section labels and segment durations vary from one dataset to another. This difference can either be explained by the style of annotations (*i.e.* label taxonomy, desired level of detail...) or the music genre. As a consequence, the average number of levels obtained after automatic hierarchy expansion is dependent on the repetition of section labels and their semantic structure, which varies with the annotation process as well.

Dataset	N	Uni	Seg	Dur	Levels
SALAMI^0 (upper)	884	5.3	10.9	63.5	2.0
SALAMI^0 (lower)	884	10.0	33.1	18.4	2.0
SALAMI^1 (upper)	884	5.0	11.2	61.1	2.0
SALAMI^1 (lower)	884	9.2	34.1	18.0	2.0
BeatlesTUT	174	5.6	10.1	36.1	2.5
RWC-Pop	100	8.9	16.4	28.5	2.9
RWC-Jazz	50	14.2	19.9	32.1	2.8
JAAH	113	6.2	8.0	63.1	2.0

Table 1. Datasets descriptive statistics. N: number of annotated songs. Uni: average number of unique section labels per song. Seg: average number of segments per song. Dur: average duration of each section per song (in beats). Levels: average number of annotation levels per song after automatic hierarchy expansion. SALAMI^i: $i th$ annotator.

3.1.2 Training data:

Since this work falls under the scope of unsupervised learning, a non annotated external audio collection is used for training. It is composed of $23,725$ tracks, spanning various musical genres such as rock, popular, rap, jazz, electronic or classical. These were retrieved from publicly available playlists and the audio obtained from Youtube. Care has been taken to discard any track from this external collection also present in one of the testing datasets.

3.2 Evaluation metrics

3.2.1 Flat segmentation:

For boundary detection, we report the F-measure [1] of the trimmed boundary detection hit-rate with a 3-second tolerance window (F_3) on the original annotations. We also report the F-measure of frame pairwise clustering [18] (F_{pairwise}), which gives another view on flat segmentation performance in terms of frame-wise section assignment.

[1] All evaluations are done using the mir_eval package [17].

3.2.2 Multi-level segmentation:

The second part of the evaluation on multi-level segmentation is carried out using the L-measure [7]. This metric allows for comparing hierarchies of segmentations operating at different scales. First, the reference hierarchy H^R is decomposed into a finite number of time instants (*i.e.* frames). Then, the set $A(H^R)$ of all triplets of frames (i, j, k) such that i and j receive the same label deeper in the hierarchy than i and k is retrieved. The same process is repeated with the same set of time instants for the estimated hierarchy H^E to obtain $A(H^E)$. Finally, the L-precision, L-recall and L-measure are derived by comparing $A(H^R)$ against $A(H^E)$. As noted in previous work [5, 10], hierarchies estimated with greater depth than reference annotations can make the L-precision metric unreliable. Therefore, our evaluation focuses on the L-recall, indicating how much of the reference hierarchy is retrieved in the estimated one. For this part of the evaluation, the expanded version of each dataset is used except for SALAMI, where for comparison purposes, the reference hierarchy only comprises both of the original annotation levels provided by each annotator (*upper* and *lower*).

3.3 Input features

All tracks are resampled at 22.05 kHz. Previous work has demonstrated that homogeneous regions and sharp changes of timbral content can be a good indicator of section transitions [19]. Therefore, we use log-scaled mel-spectrograms, with a window and hop size of 2048 and 256 respectively. We compute 60 mel-bands per frame. Beats are estimated for all tracks using the Librosa [20] implementation of the beat tracking algorithm from Ellis [21]. For both feature types, patches of 512 frames ($\simeq 5.94$s) are observed, centered at each detected beat location.

3.4 Implementation details

3.4.1 Network architecture:

We use a basic convolutional neural network architecture composed of 3 convolutional blocks, each comprising a convolutional and a max-pooling layer and Relu activation, followed by two fully-connected layers with Relu activations and a third fully-connected layer with linear activation. All convolutional layers use a kernel size of $(6, 4)$. A common practice in contrastive learning is to constrain the learned representations to lie within the unit hypersphere [22]. Therefore, the output embeddings are L2-normalized prior to distance calculations. The models were implemented with Pytorch 1.7.1 [23]. The RMSProp optimizer with default parameters is used. All models are trained on the non-annotated external audio collection described in Section 3.1 for a maximum of 200 epochs. The learning rate is set to 10^{-4} and dropout [24] is applied with probability 0.1 after each convolutional block and 0.2 after each fully-connected layer. All models [2] return embeddings of dimension $n = 128$.

[2] Code: `github.com/morgan76/HE`

3.4.2 Masks design:

In previous work, it was found beneficial to learn the masks during training to promote information sharing across similarity dimensions [12]. As in the method proposed by Lee et al. [11], we found that this did not bring any major improvement. Since information is already shared implicitly among the different hierarchy levels by the sampling strategy detailed in Section 2.1, the masks are kept disjoint from one another with equal length. After some preliminary experiments, the number of hierarchy levels $L = 4$ has been found as a good compromise between the diversity of triplets at each level and the temporal scale between the top and bottom ones.

3.4.3 Batch sampling scheme:

During training, mini-batches of size 120 are composed of 10 anchor points uniformly sampled from one song, and from which 12 triplets are derived (3 for each level). To choose good sampling parameters, we used both annotation levels of the held-out subset of SALAMI and measured the amount of true positive and true negative examples while varying $\delta_{p,min}$ and $\delta_{p,max}$ of level $\ell = 0$. It was found that setting $\delta_{p,min} = 32$ and $\delta_{p,max} = 64$ provided a good balance between the true positives rate at level $\ell = 0$ and the true negatives rate at level $\ell = 1$. For the case where $L = 4$, the rest of the parameters were set such that each level spans the same duration in beats (i.e. 16 beats) under the maximum value of 64 beats. All sampling parameters δ used for each level are summarized in Table 2.

L	ℓ	$\delta_{p,min}$	$\delta_{p,max}$	$\delta_{n,min}$	$\delta_{n,max}$
1	0	1	16	1	128
4	0	48	64	64	128
	1	32	48	48	64
	2	16	32	32	48
	3	1	16	16	32

Table 2. Sampling parameters (in beats) used in our experiments for $L = 1$ and $L = 4$ hierarchy levels.

3.5 Downstream algorithms and baselines

A common way of evaluating deep representations for music structure analysis is to measure the improvement made when combined with downstream segmentation methods. While there exists a variety music segmentation algorithms in the literature [1, 2], the one employed in these experiments was chosen to facilitate comparison against previous work. Boundary detection and section grouping on flat annotations as well as multi-level segmentation are performed with spectral clustering [9], as it remains the only unsupervised method that can output multiple levels of segmentation while being competitive. Additionally, it appears as a well-suited downstream method for hierarchical features since it operates on a graph decomposition of the audio signal. The proposed triplet sampling method forces the learned features to discriminate frames temporally close to one another at different levels in the hierarchy. Consequently, each sub-region in the embeddings learns one possible decomposition of the song. Applied on each of these sub-regions, spectral clustering can take advantage of the graph sub-structures proper to each level in order to efficiently retrieve the overall structure of the song. The original algorithm [9] takes two distinct audio features as input (MFCC and CQT), here, both features are replaced by the representations proposed in this work. Results obtained with the whole embedding matrices are denoted by HE (Hierarchical Embeddings). As an upper-bound of the proposed system, section grouping and multi-level segmentation are also performed using each individual sub-region of the embeddings (*i.e.* hierarchy levels), and the best results obtained across levels (denoted by HE_{best}) are reported. In a use case scenario, this can be seen as selecting the most adapted level of representation for each track in the testing set given a desired amount of granularity. For SALAMI, boundary detection is performed per annotator. For each, the scores obtained on both annotation levels (*upper* and *lower*) are computed both separately and combined together (best score between both levels per annotator is kept, noted *combined*). As an example, "$\text{HE}_{0,\text{best}}$" corresponds to the score obtained for the first annotator, selecting for each track the embedding level which maximizes the metric considered. In addition to results from previous work [5,9], those obtained here are compared against the method proposed by McCallum [3] (which comes down to setting $L = 1$ as described in Table 2), it is denoted as FE (Flat Embeddings).

4. RESULTS

4.1 Flat segmentation

Flat segmentation results on SALAMI are given in Table 3. The representations proposed in this work yield competitive results against the reported baselines on all the metrics considered. This trend is accentuated when the best embedding sub-region is selected. For *lower* annotations, the learned representations improve over traditional features. However, they do not perform better than those from McCallum [3], since this method uses sampling parameters that are more adapted to this level of annotation. The best-level scenario shows that the smallest temporal scales used during training (levels $\ell = 2, 3$) allow for the detection of very small regions of homogeneous timbral content, which helps detecting section changes at this level of annotation. The higher pairwise clustering scores indicate that these small detected regions are homogeneous enough to be identically labelled with spectral clustering (k-means step).

For the *upper* annotations, the results for boundary detection and pairwise clustering constantly improve over the reported baselines, indicating that for higher levels in the hierarchy, the proposed representations improve homogeneity inside annotated sections. Long-term similarities are implicitly captured by the highest embedding levels ($\ell = 0, 1$), yielding discriminative features able to separate consecutive musical sections at that level.

Finally, for both annotation levels combined, all models

perform better than when considering each level independently. The fact that difficult examples at the *lower* level are better managed at the *upper* one and vice-versa indicates that the representations learned are not specific to any particular annotation level. The very small performance gap across annotators also shows that these same representations capture relevant structure characteristics that are shared between them.

Level	Method	F_3	$F_{pairwise}$
lower	LSD [9]	0.525 ± 0.19	0.561 ± 0.16
	FE_0 [3]	0.624 ± 0.14	0.561 ± 0.14
	HE_0	0.611 ± 0.16	0.580 ± 0.15
	$HE_{0,best}$	0.643 ± 0.15	0.580 ± 0.15
	FE_1 [3]	0.611 ± 0.14	0.563 ± 0.14
	HE_1	0.600 ± 0.15	0.581 ± 0.14
	$HE_{1,best}$	0.635 ± 0.15	0.580 ± 0.14
upper	SNF [10]	0.456	0.567
	DEF [5]	0.564	0.600
	LSD [9]	0.579 ± 0.15	0.652 ± 0.13
	FE_0 [3]	0.568 ± 0.17	0.694 ± 0.14
	HE_0	0.597 ± 0.18	0.714 ± 0.14
	$HE_{0,best}$	0.627 ± 0.16	0.719 ± 0.14
	FE_1 [3]	0.559 ± 0.17	0.697 ± 0.14
	HE_1	0.595 ± 0.18	0.718 ± 0.14
	$HE_{1,best}$	0.625 ± 0.16	0.720 ± 0.14
combined	HE_0	0.665 ± 0.13	0.730 ± 0.14
	$HE_{0,best}$	0.711 ± 0.12	0.733 ± 0.14
	HE_1	0.662 ± 0.13	0.731 ± 0.14
	$HE_{1,best}$	0.707 ± 0.12	0.731 ± 0.14

Table 3. Boundary detection and section grouping results on SALAMI.

4.2 Multi-level segmentation

The results obtained for multi-level segmentation are reported in Table 4. When employing the full embedding representation, the performance on multi-level segmentation is competitive in terms of L-recall with previous work. As well as for boundary detection, selecting the best embedding sub-region leads to even further improvement. The importance of keeping inter-annotator agreement as a reference for comparison in multi-level segmentation has previously been argued [10]. It is found that the proposed representations result in multi-level segmentations that adapt to both annotators, within the range of the inter-annotator agreement reported by Tralie and McFee [10].

Method	L-precision	L-recall	L-measure
Inter-annot	0.664	0.664	0.654
LSD [7]	0.419	0.636	0.498
SNF [10]	0.431	0.668	0.517
DEF [5]	0.435	0.673	0.520
FE_0 [3]	0.412 ± 0.10	0.677 ± 0.13	0.505 ± 0.11
HE_0	0.413 ± 0.11	0.680 ± 0.13	0.507 ± 0.11
$HE_{0,best}$	0.432 ± 0.11	0.694 ± 0.13	0.527 ± 0.11
FE_1 [3]	0.413 ± 0.10	0.663 ± 0.12	0.503 ± 0.10
HE_1	0.418 ± 0.11	0.671 ± 0.13	0.509 ± 0.11
$HE_{1,best}$	0.423 ± 0.11	0.686 ± 0.13	0.517 ± 0.11

Table 4. Multi-level segmentation results on SALAMI. Inter-annot denotes the inter-annotator agreement.

4.3 Additional evaluation

In Table 5, the results obtained for boundary detection, section grouping and multi-level segmentation on additional datasets using the whole embedding matrices are summarized. The boundary detection scores obtained for BeatlesTUT and RWC-Pop fall within the same range, where more specifically, the score on RWC-Pop is higher than the one obtained by Wang et al. [4] with representations learned via supervised contrastive learning. However, a significant drop is observed for the two remaining datasets: RWC-Jazz and JAAH. If the music genre might play a role in this performance gap, it is also worth considering some statistics of these datasets summarized in Table 1. For RWC-Jazz, the high number of unique section labels compared to the total number of segments might cause some errors during the section grouping step done at the frame level with k-means (last step of the spectral clustering method). Regarding the JAAH dataset, given the low average number of segments per track, the segmentation method returns more boundaries than those originally annotated, therefore reducing the hit-rate precision and F-measure. For all metrics considered, other experiments have shown that hierarchical representations also performed better than their flat counterparts [3], of which due to space constraints, the results are not reported here.

Dataset	F_3	$F_{pairwise}$	L-P	L-R	L-M
BeatlesTUT	71.77	72.25	49.32	75.25	59.37
RWC-Pop	68.07	65.35	47.02	77.06	58.30
RWC-Jazz	55.05	58.51	32.89	81.80	45.76
JAAH	55.57	76.72	46.49	81.18	58.55

Table 5. Boundary detection, section grouping and multi-level segmentation results on additional datasets (in percentage) with the whole embedding matrix. L-P: L-precision, L-R: L-recall, L-M: L-measure.

The L-recall values obtained across datasets remain within the same range, regardless of the performance achieved on flat segmentation or section grouping. The temporal notion induced during sampling helps adapting to different musical genres or annotation sources. Even though the learned representations may not always fit with one specific level in the annotations, most of the reference structure hierarchies are captured and more refined levels of segmentation are discovered.

5. CONCLUSION

In this work, unsupervised contrastive learning of deep representations for music structure analysis at different time-scales has been explored. By leveraging time information and the hierarchical aspect of music structure, the resulting representations facilitate single and multi-level segmentation while being robust against different types of annotations. Future work includes searching for better-suited architectures to detect musical patterns at different time scales and automatically combine them to accommodate specific annotation styles or levels.

6. REFERENCES

[1] O. Nieto, G. J. Mysore, C.-i. Wang, J. B. Smith, J. Schlüter, T. Grill, and B. McFee, "Audio-based music structure analysis: Current trends, open challenges, and applications," *Transactions of the International Society for Music Information Retrieval*, 2020.

[2] O. Nieto and J. P. Bello, "Systematic exploration of computational music structure research." in *ISMIR*, 2016.

[3] M. C. McCallum, "Unsupervised learning of deep features for music segmentation," in *ICASSP*, 2019.

[4] J.-C. Wang, J. B. L. Smith, W. T. Lu, and X. Song, "Supervised metric learning for music structure features," in *ISMIR*, 2021.

[5] J. Salamon, O. Nieto, and N. J. Bryan, "Deep embeddings and section fusion improve music segmentation," in *ISMIR*, 2021.

[6] J. B. Smith and E. Chew, "Using quadratic programming to estimate feature relevance in structural analyses of music," in *Proceedings of the 21st ACM international conference on Multimedia*, 2013.

[7] B. McFee, O. Nieto, M. M. Farbood, and J. P. Bello, "Evaluating hierarchical structure in music annotations," *Frontiers in psychology*, 2017.

[8] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie, "Design and creation of a large-scale database of structural annotations." in *ISMIR*, 2011.

[9] B. McFee and D. Ellis, "Analyzing song structure with spectral clustering." in *ISMIR*, 2014.

[10] C. J. Tralie and B. McFee, "Enhanced hierarchical music structure annotations via feature level similarity fusion," in *ICASSP*, 2019.

[11] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Disentangled multidimensional metric learning for music similarity," in *ICASSP*, 2020.

[12] A. Veit, S. Belongie, and T. Karaletsos, "Conditional similarity networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[13] B. McFee and K. M. Kinnaird, "Improving structure evaluation through automatic hierarchy expansion," in *ISMIR*, 2019.

[14] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler, "Omras2 metadata project 2009," in *ISMIR*, 2009.

[15] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical and jazz music databases." in *ISMIR*, 2002.

[16] V. Eremenko, E. Demirel, B. Bozkurt, and X. Serra, "Audio-aligned jazz harmony dataset for automatic chord transcription and corpus-based research." in *ISMIR*, 2018.

[17] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "mir_eval: A transparent implementation of common mir metrics," in *ISMIR*, 2014.

[18] M. Levy and M. Sandler, "Structural segmentation of musical audio by constrained clustering," *IEEE transactions on audio, speech, and language processing*, 2008.

[19] F. Kaiser and G. Peeters, "A simple fusion method of state and sequence segmentation for music structure discovery." in *ISMIR*, 2013.

[20] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015.

[21] D. P. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, 2007.

[22] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *ICML*, 2020.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, 2019.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, 2014.