A Repetition-based Triplet Mining Approach for Music Segmentation

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

A REPETITION-BASED TRIPLET MINING APPROACH FOR MUSIC SEGMENTATION

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

Contrastive learning has recently appeared as a well-suited method to find representations of music audio signals that are suitable for structural segmentation. However, most existing unsupervised training strategies omit the notion of repetition and therefore fail at encompassing this essential aspect of music structure. This work introduces a triplet mining method which explicitly considers repeating sequences occurring inside a music track by leveraging common audio descriptors. We study its impact on the learned representations through downstream music segmentation. Because musical repetitions can be of different natures, we give further insight on the role of the audio descriptors employed at the triplet mining stage as well as the trade-off existing between the quality of the triplets mined and the quantity of unlabelled data used for training. We observe that our method requires less non-annotated data while remaining competitive against other unsupervised methods trained on a larger corpus.

1. INTRODUCTION

The task of music structure analysis consists in locating the boundaries between consecutive segments and grouping them into relevant categories, called musical sections. This problem has gained attention in the field of music information retrieval and has numerous applications, such as music generation [1, 2], music recommendation [3] or music similarity estimation [4]. Structure is also strongly linked to other musical elements such as harmony, melody and rhythm [5] and has been leveraged to address other tasks such as beat and downbeat tracking [6] or chord transcription [7].

Most methods that have been proposed for the task of music structure analysis can be categorized according to the structure trait they rely on, namely: *homogeneity*, *novelty* and *repetition* [8]. The homogeneity rule states that musical attributes should be relatively homogeneous inside musical segments or sections. Consequently, transitions from one segment to the next should result in points of important changes in musical features (*i.e.* novelty). The idea of repetition in structure assumes that sections of the same type are rather similar sequences. In other words, musical sections are generally characterized by the degree at which they repeat throughout the entire music piece, which has been the starting point of many algorithms to infer song structures [9–11]. However, both the extent to which two sequences can be considered as repetitions, or how homogeneous a given musical segment is, imply a certain definition of similarity between time instants. Such similarity criteria are usually derived from frame representations based on common audio descriptors such as harmonic and timbral features, or their combinations [8].

A line of work has focused on finding better-suited audio representations so as to make sure that frames from the same musical sections yield similar features and therefore, sharpen transitions between consecutive musical segments. Methods based on contrastive learning have recently been proposed to find such representations [12–15], as they can leverage commonalities from large quantities of music data to learn a distance metric that complies with the aforementioned requirements. Training such models either involves the use of structural annotations [13] or some pre-defined proxies to select frames that should be brought close to one another in the latent space [12, 15]. In the latter case, these heuristics mainly rely on the homogeneity principle and discard the notion of repetition occurring inside a track, preventing them from fully exploiting unlabelled data.

The method introduced in this work aims at bridging the gap between current unsupervised deep metric learning methods for music segmentation and both ideas of homogeneity and repetition that are inherent to musical structure. As in previous work [12, 15], a contrastive learning pipeline using a triplet loss is adopted. However, triplets are mined by seeking repeating sequences inside the input track with respect to various hand-crafted audio features. In a preliminary analysis, a qualitative evaluation of the triplets generated is performed by direct comparison with structural annotations of a manually annotated test dataset. We then measure how these representations impact down-

stream segmentation on two datasets for music structure analysis. Finally, we demonstrate that our approach requires less non-annotated data than previous similar methods. We also give further insight on how the choice of the input features used to mine triplets affects training and its relationship with the music genre that the resulting representations are tested on.

2. RELATED WORK

Numerous methods for music structure analysis rely on measuring similarity between every point of a music recording to retrieve homogeneous segments and transitions between them. Since music is naturally multi-dimensional, many factors such as harmony, timbre or instrumentation can be associated with boundaries between musical sections [16]. Therefore, several strategies have been adopted to capture short-term similar regions, and it has been shown that sharp timbre changes can be a good cue for section transitions [17–19].

However, not all boundaries can be explained solely by such changes in musical features, as the perception of structure is also greatly affected by additional characteristics of a music recording such as parallelism, pauses or musical rules proper to the music genre considered [16]. Therefore, other approaches tend to rely on the repetition principle to characterise the structure of a music piece. For example, early work on music segmentation has attempted to find audio representations to identify repeating elements inside music recordings, such as pitch estimation or polyphonic transcription [10]. Generally, repetition-based methods rely on harmony-related information from the audio, as the instrumentation or other factors are subject to variations between different occurrences of a given musical section [18, 20].

Several algorithms have also been proposed to unify these two types of approaches by recognizing similar regions and repetitions of varying lengths. For example, integrating structural information at different scales into frame representations has led to considerable improvements in the recognition of musical segments [21, 22].

Even though these methods are theoretically well grounded and have proven to be efficient on commonly used datasets, the traditional hand-crafted descriptors they use can fail at accommodating different structure types and music genres. On the other hand, deep learning-based methods are able to extract efficient features from large quantities of data, thus, surpassing traditional audio descriptors [12]. Approaches based on contrastive learning also have the advantage to be easily incorporated into the classical music structure analysis pipeline, by simply replacing the original input features by the deep embeddings they learn from training data. To this end, Wang *et al.* [13] use structural annotations from a labelled training dataset to find positive and negative pairs of frames and a multi-similarity loss function [23]. They additionally employ a mining mechanism to further improve convergence of their model. Using structural annotations allows for explicitly enforcing frames of identical sections to yield similar features regardless of their appearance throughout the track. Despite not relying on annotations, the method in this work is similar to theirs, in the sense that it explicitly considers section repetitions inside a music recording.

A similar method proposed by McCallum [12] proceeds in an unsupervised manner with a triplet loss. This time, positive and negative frames are sampled using time proximity as a proxy: frames occurring within a small time interval are more likely to belong to the same musical sections than those separated by a larger amount of time. While this assumption generally holds true, it completely discards the notion of repetition, which can limit the efficacy of the approach. In the present work, this limitation is addressed by using pairwise frame similarity measures as prior information to guide the triplet sampling mechanism. This temporal-based mining method [12] is used as a baseline in this work and referred to as *temporal sampling*.

3. METHOD

The core of the triplet mining method proposed in this work resides in the estimation of a self-similarity matrix, which should reflect as much as possible section label assignment corresponding to structural annotations. This approximation of ideal pairwise frame similarities should yield high values for frames belonging to the same musical section, and low values otherwise. This self-similarity matrix is used as a probability mass function according to which are sampled, for each given frame, positive and negative examples across the whole input track.

3.1 Triplet loss

The method proposed in this work consists in finding triplets of audio feature patches (x_a, x_p, x_n) where x_a is the anchor, x_p is a positive example from the same musical section and x_n the negative example sampled from a different one without using structural annotations. The models are trained using the triplet loss, which for a given triplet $\mathcal{T} = (x_a, x_p, x_n)$ is expressed as:

$$\mathcal{L}(\mathcal{T}) = [d(f(x_a), f(x_p)) - d(f(x_a), f(x_n)) + \delta]_+ \ , \quad (1)$$

where $d(x, y)$ is a pre-defined distance metric, $[.]_+$ denotes the Hinge loss, $\delta > 0$ is the margin parameter, and $f(x)$ is the projection of x into the embedding space by a deep neural network.

3.2 Finding repetitions

The choice of the input features from which frame-wise similarities are extracted greatly influences the final triplet sampling mechanism. As the goal is to jointly detect homogeneous regions and overall repetitions throughout the input track, we employ a combination of timbral and harmonic features as done in previous work [24, 25]. These features are beat-synchronized beforehand, using the algorithm from Korzeniowski *et al.* [26] implemented in the *madmom* package [27]. One way to emphasize repetition is to encode features into time-delay embeddings, so

that pairwise comparisons are performed over short time-windows: given a sequence $X = \{\mathbf{X}_i\}_{i\in\{1,\dots,N\}}$ of feature vectors, the ith time embedding vector $\tilde{\mathbf{X}}_i$ is obtained by stacking the m feature vectors ranging from $i-(m-1)$ to i:

$$\tilde{\mathbf{X}}_i = \left[\mathbf{X}_i^T \mathbf{X}_{i-1}^T \dots \mathbf{X}_{i-(m-1)}^T\right]^T, \tag{2}$$

where m denotes the embedding dimension, ruling how much of past information is considered. Such transformations have successfully been used for music structure analysis [22], structure-based music similarity [4] and more generally in the field of non-linear time series analysis [28]. The final representation's temporal dimension remains N, as X is first zero-padded before transformation. Then, a self-similarity matrix is built from the obtained sequence of time-lag features such that:

$$M(i,j) = \begin{cases} \exp\left(-\frac{d(\tilde{\mathbf{X}}_i,\tilde{\mathbf{X}}_j)}{b}\right), & \tilde{\mathbf{X}}_j \in \mathrm{NN_k}(\tilde{\mathbf{X}}_i) \\ 0, & \tilde{\mathbf{X}}_j \notin \mathrm{NN_k}(\tilde{\mathbf{X}}_i) \end{cases} \tag{3}$$

where $d(x,y)$ is the euclidean distance, b the bandwidth parameter, $\mathrm{NN_k}(x)$ denotes the k-nearest neighbors of x and $i,j = 1,\dots,N$. The self-similarity matrix M is then filtered with a sigmoid activation, such that:

$$\hat{M}(i,j) = \sigma\left(\frac{M(i,j)}{\max_k M(i,k)}\right), \tag{4}$$

where $i,j = 1,\dots,N$ and the σ function defined as:

$$\sigma(x) = \frac{1}{1+e^{-\alpha(x-\beta)}}, \tag{5}$$

where $\alpha > 0$ is a parameter ruling the steepness of the curve and $\beta \in [0,1]$ a threshold above which the components of S are set to values close to 1. This process is applied both using MFCC and chroma features, from which we obtain their respective filtered self-similarity matrices S_M and S_C using Equation (4) (first row of Figure 1). The matrix S is then obtained by linear combination, such that:

$$S = \gamma S_M + (1-\gamma) S_C, \tag{6}$$

where $\gamma \in [0,1]$ weights the contributions of each feature type. The matrix S (second row, left column of Figure 1) is row-wise min-max normalized and filtered with the sigmoid function defined in Equation (5), diagonal stripes indicating repeating sequences are enhanced by median filtering similar to the one used by McFee *et al.* [18].

3.3 Imposing segment homogeneity

The obtained pairwise similarity S provides information about the repetitions present inside the input track. However, using it as it is to mine positive (large $S(a,p)$) and negative examples (small $S(a,n)$) would result in many trivial triplets, as positives would be located at exact points of repetitions. Therefore, a dilation operation is applied to the matrix S to enlarge these detected regions of repetition. Similar to the method by Serra *et al.* [22], a two-dimensional Gaussian kernel G of size K is convolved with S:

$$S_p = S * G, \tag{7}$$

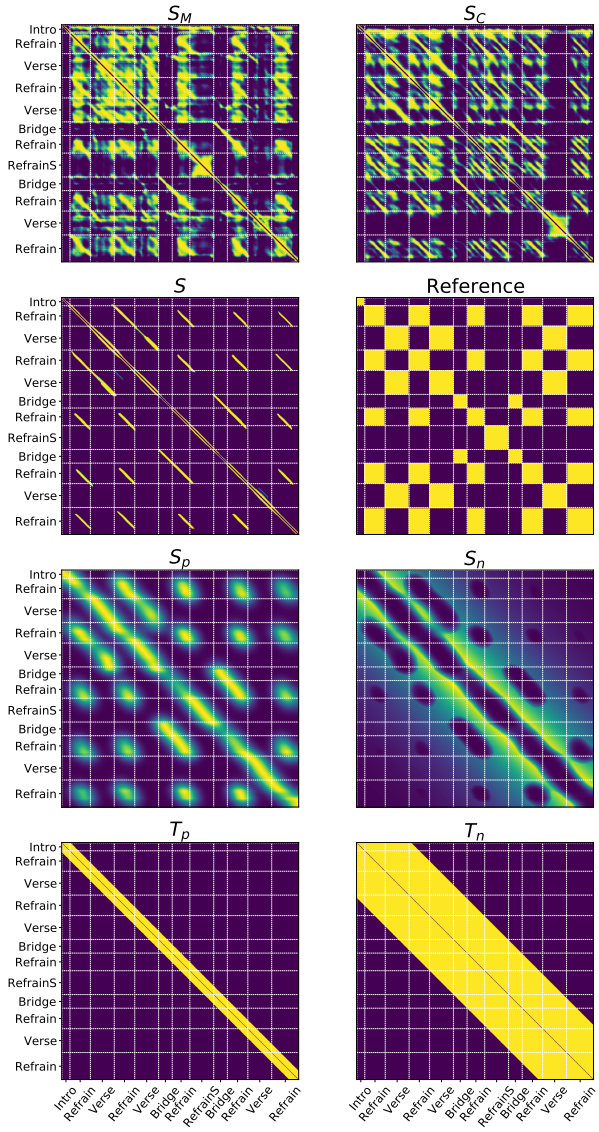

Figure 1. Example of the self-similarity approximation process for *The Beatles — Baby's In Black*. Top to bottom, left to right: self-similarity lag-matrices obtained using MFCC (S_M), chroma features (S_C), median filtered combination (S), reference self-similarity matrix (supervised scenario), positive matrix (S_p), negative matrix (S_n), positive (T_p) and negative (T_n) sampling matrices using *temporal sampling* [12]. White dotted lines denote boundary instants.

This has the effect of blurring the regions of S around its diagonal stripes, which approximates the width of the corresponding musical segments in a more uniform manner than directly using the unfiltered matrix S. The size of the kernel K logically impacts the extent to which this dilation is performed. It was found that setting $K = 8$ (beats) provided a good balance between the amount of dilation and its alignment with segment boundaries (third row, left column of Figure 1), as it blurs repetitions over 2 bars when songs follow a $4/4$ time signature [1].

[1] Such value might induce a bias towards specific western music genres. This parameter should ideally be adapted to each training track.

3.4 Negative mining

While the matrix S_p guides the selection of positive examples for any frame of the input track, the triplet loss requires to find a third point with a different label, called negative example. In our case, such example should belong to a different musical section, which could be easily solved by searching for the least similar frames from the anchor (*i.e.* using the matrix $S_n = 1 - S_p$ for sampling). However, doing so is likely to result in trivial triplets where the relative difference between $d(f(x_a), f(x_p))$ and $d(f(x_a), f(x_n))$ from Equation (1) might already be larger than the margin δ, thus, yielding small gradients that prevent the network from learning features that are discriminative enough [29]. Instead, we enforce negative examples to be chosen close to the anchor's location while still avoiding homogeneous regions indicated by the positive matrix S_p. To this end, the negative sampling matrix S_n is obtained by applying an exponential decay to $1 - S_p$ such that:

$$S_n(i,j) = (1 - S_p(i,j))e^{-\lambda \max\left(\frac{|i-j|}{N}, S_p(i,j)\right)}, \quad (8)$$

where $\lambda > 0$ is a parameter that defines the strength of the smoothing. As a consequence, components near the main diagonal of S_n (third row, right column of Figure 1) receive greater values than those close the opposite edges, thus favoring frames located within consecutive segments of that of the anchor.

The final sampling process works as follows: given an anchor point i_a chosen among the N frames of the input track, the weight attributed to a certain index i_k when sampling the positive example follows the discrete probability distribution defined by the a-th row of S_p, such that $Pr(I = i_k) = S_p(a,k)$. The negative example is chosen in a similar fashion with the matrix S_n, such that $Pr(I = i_k) = S_n(a,k)$.

4. EXPERIMENTAL SETTING

This section details the experiments performed to assess the efficacy of the proposed triplet mining method. First, a preliminary evaluation of the triplets generated is done against structural annotations from a commonly used dataset for music structure analysis. Secondly, we train two separate convolutional neural networks using triplets obtained by *temporal sampling* and those from our method. The obtained embeddings are fed as input to a downstream music segmentation algorithm and performance on both boundary detection and structural grouping is measured. Finally, to gain more insight on the quality of the triplets generated, training is performed on different fractions of the unlabelled training dataset.

4.1 Datasets

Since this work falls under the scope of unsupervised learning, a non annotated external audio collection is used for training. It is composed of $20,000$ 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 following testing datasets. Training is separately done on 10%, 50% and 100% of this dataset.

SALAMI: the Structural Annotations for Large Amounts of Music Information (SALAMI) [30] contains $1,359$ tracks ranging from classical, jazz, popular to world and live music. For evaluation, we use the *upper* annotations of a subset of 884 songs labelled by two different annotators.

JSD: the Jazz Structure Dataset [31] gathers 340 jazz recordings provided with two-level annotations: the chorus level (a full cycle of the harmonic schema, which is the annotation level used for evaluation) and a solo level, consisting of one more choruses. These annotations follow the common jazz structure schema that includes the introduction of the main melody (theme), followed by alternating solos from the different musicians and a final return towards the main theme at the end of the track.

4.2 Evaluation metrics

Common evaluation metrics for automatic structure analysis are employed throughout our experiments. For boundary detection, we report the F-measure [2] of the trimmed [3] boundary detection hit-rate with a 0.5 and 3-second tolerance windows (HR.5F, HR3F respectively). For structural grouping, we report the F-measure of frame pairwise clustering [21] (PFC), which gives another view on flat segmentation performance in terms of frame-wise section assignment. Additionally, the normalized conditional entropy score (NCE) [33] is also calculated, in order to indicate from a probabilistic perspective the amount of information shared between predicted label distributions and their corresponding reference annotations. In the case where the test dataset has more than one annotator, the best score across annotators is kept, as the goal of the evaluation process is to measure how close to human ground-truth the predicted segmentations are. The average score obtained per metric is reported and the statistical significance is assessed using a paired-sample T-test with $p < 0.05$.

4.3 Implementation details

Input features: All tracks are resampled at 22.05 kHz. We use log-scaled Mel-spectrograms as input to the deep network, with a window and hop size of 2048 and 256 respectively. We compute 60 Mel-band coefficients per frame. Feature patches are composed of 512 frames ($\simeq 5.94$s) and centered at each detected beat location.

Mining parameters: Chroma features are extracted using a minimum frequency of 27.5 Hz over 8 octaves. 20 MFCC coefficients are calculated per frame and the very first one is discarded. Both are calculated with the librosa

[2] All evaluations are done using the mir_eval package [32].

[3] The first and last boundaries are discarded during evaluation, as they correspond to the beginning and the end of the track and therefore, do not provide any information regarding the system's performance.

library [34]. The features are encoded into time-delay representations using context values of $m = 16$ and $m = 8$ beats respectively. The parameters α and β of the sigmoid filtering step are set to 60 and 0.85. We give equal weight to each feature by setting the $\gamma = 0.5$ in Equation (6). Finally, the negative matrix S_n is calculated with a smoothing parameter $\lambda = 5$. These parameters were found using simple grid searches and visual inspections of the obtained self-similarity matrices.

Network architecture: The encoder consists of a convolutional neural network composed of 3 convolutional layers, each followed by a max-pooling layer and Elu activation, and two fully-connected layers comprising 128 units with Elu and linear activations respectively. All convolutional layers use a kernel size of size $(3, 3)$ with 32 filters each. The output embeddings are ℓ_2-normalized before calculating the triplet loss. The models are implemented[4] with Pytorch 1.7.1 [35]. The SGD optimizer with 10^{-4} weight decay and 0.9 momentum is used, the models are trained for a maximum of 200 epochs, where each batch is composed of 256 triplets obtained from one single track. Similar to previous work [12], the margin parameter δ is set to 0.1 and the embedding dimension to $d = 128$.

Downstream segmentation: For all experiments, the embeddings returned by each model are fed as input to spectral clustering [24], as this algorithm jointly performs both boundary detection and structural grouping in an unsupervised manner and has proven to be efficient in previous studies [13, 14]. This also allows one to compare the influence of each of the tested representations into a single unified framework. The original algorithm takes two distinct beat-synchronized audio features as input (MFCC and CQT). We consider this method as a second baseline which we denote as LSD (Laplacian Structural Decomposition). However in our case, it is directly applied to the self-similarity S_p of each track. When this algorithm is combined with deep representations, we simply replace both input features by the embedding matrix. Finally, because spectral clustering outputs multiple levels of segmentation, only the one maximizing the considered metric is reported (HR.5F and HR3F for boundary detection, PFC and NCE for structural grouping).

5. RESULTS

5.1 Preliminary evaluation

We generate 256 triplets per track contained in the SALAMI dataset and report the proportions of true positives, true negatives and correct triplets in Table 1. For comparison purposes, we also provide a random baseline, where each anchor, positive and negative example is uniformly sampled over the whole track. The sampling method proposed significantly improves the selection of negative examples compared to the *temporal sampling* approach. However, random negative sampling performs better than our approach. This was to be expected, since the latter samples negatives over the whole track while our method greatly narrows down the number of probable candidates (see Equation (8)). Conversely, the *temporal sampling* returns a higher proportion of true positives than ours, since these are sampled in a relatively short time window around their respective anchor, thus omitting any section repetition occurring inside the input track. All in all, our approach returns a much higher proportion of correct triplets than either of the comparison strategies while guaranteeing that positive examples are located within the right musical sections and the negative within a relatively short time window around their anchor's.

Sampling	TP	TN	CT
Random	$.401 \pm .22$	$.595 \pm .21$	$.194 \pm .06$
Temporal [12]	$.886 \pm .32$	$.398 \pm .49$	$.325 \pm .47$
Ours	$.800 \pm .40$	$\mathbf{.583 \pm .49}$	$\mathbf{.432 \pm .50}$

Table 1. Triplet mining results on *upper* annotation level of SALAMI dataset. TP, TN, CT: proportions of true positives, true negatives and correct triplets respectively. Results highlighted in bold denote statistically significant improvements over *temporal sampling* according to a paired-sample T-test with $p < 0.05$.

5.2 Segmentation and structural grouping

Table 2 shows the performance of our approach against *temporal sampling* on the *upper* annotations of the SALAMI dataset. Regardless of the amount of training data, our method constantly improves both boundary detection and structural grouping in a significant manner. It is also interesting to see that such improvement is already achieved when the proposed method uses only 10% of the training dataset. This corroborates the results from Section 5.1, showing that improving the triplets quality provides a cleaner training signal and makes learning more efficient.

Method (Split)	HR.5F	HR3F	PFC	NCE
LSD	.195	.486	.707	.682
Temp. (10%) [12]	.280	.665	.770	.677
Ours (10%)	**.291**	**.676**	**.777**	**.691**
Temp. (50%) [12]	.288	.671	.773	.678
Ours (50%)	**.296**	**.682**	**.778**	**.690**
Temp. (100%) [12]	.284	.670	.773	.678
Ours (100%)	**.297**	**.683**	**.781**	**.694**

Table 2. Flat segmentation results on SALAMI (*upper* annotations). Results in bold denote statistically significant improvement over *temporal sampling* on same split (denoted as *Temp.*).

From a more qualitative perspective, Figure 2 shows examples of self-similarity matrices derived from the embeddings trained with *temporal sampling* and our method. In the latter case, consecutive musical sections are better discriminated (clearer block structures on the main diagonal). Section repetitions (visible as diagonal stripes and off-diagonal blocks) are more straightforward to recog-

[4] Code: `github.com/morgan76/Triplet_Mining`

nize, especially those with relatively small durations (sections A, B or D).

Method (Split)	HR.5F	HR3F	PFC	NCE
LSD	.195	.486	.707	.682
Temp. (10%) [12]	.221	.568	.739	.745
Ours (10%)	.219	**.586**	.744	.749
Temp. (50%) [12]	.243	.586	.763	.766
Ours (50%)	.222	.583	.755	.758
Temp. (100%) [12]	.229	.590	.766	.767
Ours (100%)	.225	.592	.754	.760

Table 3. Flat segmentation results on JSD (*chorus* annotation level). Results in bold denote statistically significant improvement over *temporal sampling* (denoted as *Temp.*) on same split.

Results on the JSD dataset are given in Table 3. Here, the improvements made are not as consistent. However, when using only 10% of the training dataset, the performance of our approach remains within the same range than that of the baseline when trained on larger splits. Compared to the results obtained on SALAMI, the small improvements made here can be associated with the way structure is defined in terms of feature similarity in jazz.

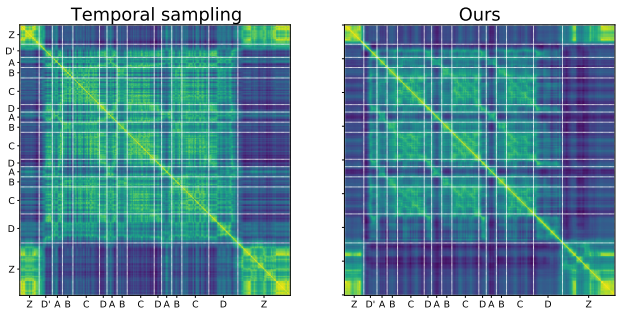

Figure 2. Example of self-similarity matrices for the track SALAMI 1380. Left: encoder trained with *temporal sampling*. Right: encoder trained using the proposed triplet mining method. White dotted lines denote boundary instants.

5.3 Discussion on mining parameters

Impact on triplet selection: The sampling parameters could further be tuned to improve performance. More specifically, the audio descriptors employed at the first stage and their combination could be adapted to the training data in order to better emphasize more specific aspects of the audio. For example, some music genres such as pop music or rock generally rely on the repetition of certain chord progressions [1]. However, introducing a degree of timbral homogeneity allows for differentiating two sections that are semantically similar, such as in the example from Figure 1, 'refrain' and 'refrain-Solo'. Putting more emphasis on timbral features might be better adapted to music genres such as jazz, where structure is highly influenced by changes in soloists. As an example, Figure 3 displays the positive sampling matrices obtained when varying the γ parameter from Equation (6). It is clear to see that favoring timbral similarity helps better approximating segment transitions and mutual dissimilarities between the successive solos of saxophone, piano and guitar.

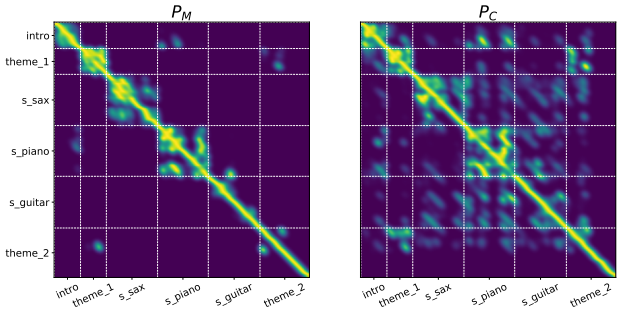

Figure 3. Example of positive sampling matrices for *Michael Brecker — Song for Bilbao*. Left: emphasis on timbral content ($\gamma = 0.9$). Right: emphasis on harmonic content ($\gamma = 0.1$). White dotted lines denote boundary instants.

Impact on segmentation: To illustrate how the balance between harmonic and timbral features impacts the final segmentation, the encoder is trained on the 10% and 50% splits of the dataset with $\gamma = 0.9$, thus putting a stronger emphasis on the MFCC-based similarity at the triplet mining stage. All other parameters are kept to their initial values described in Section 4.3. The segmentation results summarized in Table 4 show that the choice of the parameter γ does impact the training process. In this case, putting more weight on timbral information seems to make the representations more sensitive to timbral changes and improves boundary detection (HR3F) in a significant manner compared to *temporal sampling*.

Method (Split)	HR.5F	HR3F	PFC	NCE
Temp. (10%) [12]	.221	.568	.739	.745
Ours (10%, $\gamma = 0.9$)	.223	**.585**	.743	.750
Temp. (50%) [12]	.243	.586	.763	.766
Ours (50%, $\gamma = 0.9$)	.234	**.607**	.769	.772

Table 4. Flat segmentation results on JSD (*chorus* annotation level) with emphasis on timbral features ($\gamma = 0.9$). Results in bold denote statistically significant improvement over *temporal sampling* (denoted as *Temp.*) on same split.

6. CONCLUSION

This work introduced a repetition-based triplet mining mechanism to learn efficient audio representations prior to music segmentation, which can significantly improve both boundary detection and structural grouping, while needing less data than previous similar methods. Complementary experiments demonstrate that this sampling process can be further adapted to the final type of segmentation desired by either emphasizing harmonic or timbral information from the input track.

7. ACKNOWLEDGEMENTS

This work was performed using HPC resources from GENCI-IDRIS (Grant 2022-AD011013255R1).

8. 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*, vol. 3, no. 1, 2020.

[2] S. Dai, Z. Jin, C. Gomes, and R. B. Dannenberg, "Controllable deep melody generation via hierarchical music structure representation," *ISMIR*, 2021.

[3] A. Bozzon, G. Prandi, G. Valenzise, M. Tagliasacchi *et al.*, "A music recommendation system based on semantic audio segments similarity," *Proceeding of Internet and Multimedia Systems and Applications-2008*, pp. 182–187, 2008.

[4] J. P. Bello, "Measuring structural similarity in music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2013–2025, 2011.

[5] S. Dai, H. Zhang, and R. B. Dannenberg, "Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music," 2020.

[6] M. Fuentes, B. McFee, H. C. Crayencour, S. Essid, and J. P. Bello, "A music structure informed downbeat tracking system using skip-chain conditional random fields and deep learning," in *ICASSP*, 2019.

[7] M. Mauch, K. C. Noland, and S. Dixon, "Using musical structure to enhance automatic chord transcription." in *ISMIR*, 2009.

[8] J. Paulus, M. Müller, and A. Klapuri, "State of the art report: Audio-based music structure analysis." in *ISMIR*, 2010.

[9] M. Goto, "A chorus section detection method for musical audio signals and its application to a music listening station," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1783–1794, 2006.

[10] R. B. Dannenberg and N. Hu, "Pattern discovery techniques for music audio," *Journal of New Music Research*, vol. 32, no. 2, pp. 153–163, 2003.

[11] M. Müller and F. Kurth, "Towards structural analysis of audio recordings in the presence of musical variations," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–18, 2006.

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

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

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

[15] M. Buisson, B. McFee, S. Essid, and H. C. Crayencour, "Learning multi-level representations for hierarchical music structure analysis," in *ISMIR*, 2022.

[16] J. B. Smith, C.-H. Chuan, and E. Chew, "Audio properties of perceived boundaries in music," *IEEE transactions on multimedia*, vol. 16, no. 5, pp. 1219–1228, 2014.

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

[18] B. McFee and D. P. Ellis, "Learning to segment songs with ordinal linear discriminant analysis," in *ICASSP*, 2014.

[19] K. Ullrich, J. Schlüter, and T. Grill, "Boundary detection in music structure analysis using convolutional neural networks." in *ISMIR*, 2014.

[20] J. Paulus and A. Klapuri, "Music structure analysis using a probabilistic fitness measure and a greedy search algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1159–1170, 2009.

[21] M. Levy and M. Sandler, "Structural segmentation of musical audio by constrained clustering," *IEEE transactions on audio, speech, and language processing*, vol. 16, no. 2, pp. 318–326, 2008.

[22] J. Serra, M. Müller, P. Grosche, and J. L. Arcos, "Unsupervised music structure annotation by time series structure features and segment similarity," *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1229–1240, 2014.

[23] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5022–5030.

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

[25] G. Shibata, R. Nishikimi, and K. Yoshii, "Music structure analysis based on an lstm-hsmm hybrid model." in *ISMIR*, 2020.

[26] F. Korzeniowski, S. Böck, and G. Widmer, "Probabilistic extraction of beat positions from a beat activation function." in *ISMIR*, 2014.

[27] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "Madmom: A new python audio and music signal processing library," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 1174–1178.

[28] H. Kantz and T. Schreiber, *Nonlinear time series analysis*. Cambridge university press, 2004, vol. 7.

[29] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[30] 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.

[31] S. Balke, J. Reck, C. Weiß, J. Abeßer, and M. Müller, "Jsd: A dataset for structure analysis in jazz music," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, 2022.

[32] 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.

[33] H. M. Lukashevich, "Towards quantitative measures of evaluating song segmentation." in *ISMIR*, 2008.

[34] 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*, vol. 8, 2015, pp. 18–25.

[35] 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.