SELF-SIMILARITY-BASED AND NOVELTY-BASED LOSS FOR MUSIC STRUCTURE ANALYSIS

Geoffroy Peeters
LTCI, Télécom-Paris, Institut Polytechnique de Paris, France

ABSTRACT

Music Structure Analysis (MSA) is the task aiming at identifying musical segments that compose a music track and possibly label them based on their similarity. In this paper we propose a supervised approach for the task of music boundary detection. In our approach we simultaneously learn features and convolution kernels. For this we jointly optimize - a loss based on the Self-Similarity-Matrix (SSM) obtained with the learned features, denoted by SSM-loss, and - a loss based on the novelty score obtained applying the learned kernels to the estimated SSM, denoted by novelty-loss. We also demonstrate that relative feature learning, through self-attention, is beneficial for the task of MSA. Finally, we compare the performances of our approach to previously proposed approaches on the standard RWC-Pop, and various subsets of SALAMI.

1 Introduction

Music Structure Analysis (MSA) is the task aiming at identifying musical segments that compose a music track (a.k.a. segment boundary estimation) and possibly label them based on their similarity (a.k.a. segment labeling). We deal here with MSA from audio. MSA is one of the oldest task in Music Information Retrieval [1] but still one of the most challenging. This is due to the difficulty to exactly define what music structure is and hence be able to create annotated datasets to measure progress or train systems. People agree that the structure can be considered from multiple viewpoints [2] [2] [3], is hierarchical [4] and is partly subjective [5]. Probably because of this complexity, the number of contributions in MSA has remained low despite its large number of applications: audio summarization [6], interactive browsing [7–9], musical analysis [10], tools for researcher (to help chord recognition [11], source separation [12] or downbeat estimation [13]).

To solve the two MSA tasks (boundary detection and segment labeling), three assumptions [14] are commonly used: (1) *novelty* (we assume that segments are defined by large —novel— changes of the musical content over time), (2) *homogeneity* (the musical content is homogeneous within a given segment) and (3) *repetition* (the musical content —homogeneous or not— can be repeated over time). This has been extended by [15] to a fourth *regularity* assumption (the segment's durations are regular over time). Combining those allows to construct MSA systems.

1.1 Related works

Over time, a large palette of approaches has been proposed for MSA. We only review the ones related to our work and refer the reader to Nieto et al. [16] for a good overview. We consider three periods according to the nature of the audio features –hand-crafted (HC) or learned by deep learning (DL) –, and the nature of the detection system which uses the audio features – HC or trained by DL –.

First period: HC detection system applied to HC audio features. In these systems HC audio features (such as MFCC or Chroma) were given as input to HC detection system (such as the checkerboard kernel, novelty-score [17]), unsupervised training (such as HMM [6], NMF [18]), supervised (such as OLDA [19]) or pattern matching algorithms (such as DTW [20] or variants [21]) .

Second period: DL detection system applied to HC audio features. Over time, more and larger annotated datasets for MSA have been developed; which concomitantly with the development of DL has allowed to reformulate the MSA task in terms of supervised learning. The detection system developed here mainly target the task of boundary detection. For example, [22] [23] [24] propose to train in a supervised way a Convolutional Networks (ConvNet) $\hat{y} = f^{\theta}(\mathbf{X})$ to estimate if the center of a patch of HC audio features $\mathbf{X}$ is a boundary (y=1). Various HC audio features (or combinations of) are used here: Log-Mel-Spectrogram, Pich-Class-Profile through SSM expressed in (time,time) or (time,lag).

Third period: HC detection system applied to DL audio features. To deal with the endless debate about the choice of HC audio features, McCallum et al. [25] propose to learn them. For this, they train an encoder f^{θ} by minimizing a Triplet Loss (TL) [26] between patches of beat-synchronous Constant-Q-Transform (CQT). For the TL, they propose a Self-Supervised-Learning (SSL) paradigm [3] to define the anchor A patch, positive P patch and negative N patch. Using the homogeneity assumption, neighboring times are supposed to be more similar to each

[1] Foote's paper [1] on SSM was published in 1999.

[2] musical role, acoustic similarity, instrument role, perceptual tests

[3] which does not require any annotated segments and labels

other (therefore used to define A and P) than to distant ones (used to define N). For training they use a very large unlabeled dataset of 28345 songs. This method however does not consider the repetition assumption [4].

Wang et al. [27] revised McCallum approach in a supervised setting. In this, the patches P (resp. N) are now explicitly chosen so as to have the same (resp. different) annotated segment label than the patches A. This supervised method now consider both the homogeneity and repetition assumption. In another work [28], they propose a spectral-temporal Transformer-based model (SpecTNT) trained with a connectionist temporal localization (CTL) loss to jointly estimate music segments ad their labels.

McCallum approach has also been extended by Buisson et al. [29] to take benefit from the hierarchy of structure in music. They show that the obtained deep embeddings can improve segmentation at various levels of granularity.

Rather than learning features for MSA, Salamon et al. [30] proposed to re-use pretrained ones. Those are obtained using encoders previously trained on different tasks (Few-Shot Learning sound event and music auto-tagging). Those are then used as input to a Laplacian Structural Decompositon algorithm for MSA.

1.2 Proposal and paper organization

Following the previous taxonomy, our proposal would belong to the category "DL detection system applied to DL audio features" . Unlike previous feature learning approaches (that rely on a Triplet Loss paradigm), we utilize a more straightforward paradigm (illustrated in Figure 1) which is a succession of two steps, each with its own objective. The two objectives are jointly optimized.

In the **first step**, we learn the parameters θ of an encoder f^θ such that when applied to the sequence of inputs $\{\mathbf{X}_i\}_{i\in\{1...T\}}$ that represent a given track (where T is the length of temporal sequence), the encoded features allows the estimation of a SSM, $\hat{\mathbf{S}}^\theta_{ij}$, which attempts to reproduce a ground-truth SSM, $\mathbf{S}_{ij}$. For training f^θ we use an approach similar to the SSM-Net approach proposed in [31], i.e. defining a loss which directly compare the obtained SSM $\hat{\mathbf{S}}^\theta_{ij}$ to a ground-truth SSM $\mathbf{S}_{ij}$.

In the **second step**, we learn a set of kernels $\mathbf{K}^\theta$ such that when convolved over the main diagonal of the estimated SSM $\hat{\mathbf{S}}^\theta_{ij}$ it allows the estimation of a novelty score $\hat{\mathbf{n}}^\theta_i$, which attempts to reproduce a ground-truth novelty score, $\mathbf{n}_i$. This novelty score is usually obtained using a fixed checkerboard kernel [32]. The resulting function is named novelty score since high values in it indicate times where the content change (it is homogeneous before and after). It has been shown that better kernels can be used (for example using multi-scale kernels [33]) and that it is possible to train such kernels $\mathbf{K}^\theta$ considered as the kernels of a ConvNet (for example [22] and [23] in the case of a (time,lag) SSM or [24] in the case of a (time,time) SSM, which is our case). This is the approach we follow here.

[4] N could potentially be in a segment which is a repetition of the segment containing A

Another proposal we make in this paper, is to consider the learning of relative features, i.e. features which are relative to the given track.

Paper organization. We provide an overview of our system in part 2, describe the inputs to our system (part 2.1), detail the two losses (parts 2.2 and 2.3), motivate relative feature learning (part 2.4), detail the architecture of our encoder f^θ (part 2.5) and the training process (part 2.6). In part 3, we provide a large-scale evaluation of our proposal. It should be noted that although we only evaluate our method for the task of segment boundary detection, it can also be used for segment labeling given the clearness of the obtained SSM.

2 Proposal

2.1 Input data $\{\mathbf{X}_i\}$

The inputs $\{\mathbf{X}_i\}$ to our system are simple patches [5] of Log-Mel-Spectrogram. We didn't consider beat-synchronous features as in [25] given the non-reliability of beat estimation outside popular music. Using `librosa` [34], we first computed Mel-spectrogram with 80 mel-bands, using a 92ms window length and 23ms hop size. Those are then converted to log-amplitude using $\log(1+100\cdot mel)$. We then aggregate them (mean operator) over time to lead to a 0.1s hop size. The final $\{\mathbf{X}_i\}$ are then patches of 40 successive frames (corresponding to 4s.) with a hop size of 5 frames (corresponding to 0.5s.).

2.2 SSM-loss

Given a sequence of inputs $\{\mathbf{X}_i\}_{i\in\{1...T\}}$, we apply the same encoder f^θ individually to each $\mathbf{X}_i$ to obtain the corresponding sequence of embeddings $\{\mathbf{e}^\theta_i\}_{i\in\{1...T\}}$. Those are then L2-normalized. We can then easily construct an estimated SSM, $\hat{\mathbf{S}}^\theta_{ij}$, using a distance/similarity/divergence g between all pairs of projections:

$$\hat{\mathbf{S}}^\theta_{ij} = g(\mathbf{e}^\theta_i = f^\theta(\mathbf{X}_i), \mathbf{e}^\theta_j = f^\theta(\mathbf{X}_j)), \quad \forall i,j \qquad (1)$$

We use here a "scaled" cosine-similarity for g which, because the embeddings are L2-normalized, reduces to

$$\hat{\mathbf{S}}^\theta_{ij} = 1 - \frac{1}{4}\|\mathbf{e}^\theta_i - \mathbf{e}^\theta_j\|_2^2 \quad \in [0,1] \qquad (2)$$

It is then possible to compare $\hat{\mathbf{S}}^\theta_{ij}$ to a ground-truth binary SSM, $\mathbf{S}_{ij}$, derived from annotations. We consider this as a multi-class (a set of T^2 binary classifications) problem and hence minimize the sum of Binary-Cross-Entropy (BCE) losses. However, given the unbalancing between the two classes in $\mathbf{S}_{ij}$ (which contains much more 0 than 1), we used a weighting factor λ computed as the percentage of positive values in $\mathbf{S}_{ij}$. The lower λ is, the more we put emphasis on positive ($\mathbf{S}_{ij}$=1) examples:

$$\mathcal{L}^\theta_{SSM} = -\frac{1}{T^2}\sum_{i,j=1}^{T}(1-\lambda)\left[\mathbf{S}_{ij}\log(\hat{\mathbf{S}}^\theta_{ij})\right] + \lambda\left[(1-\mathbf{S}_{ij})\log(1-\hat{\mathbf{S}}^\theta_{ij})\right] \qquad (3)$$

Since the computation of the SSM $\hat{\mathbf{S}}^\theta_{ij}$ is differentiable w.r.t. to the embeddings $\{\mathbf{e}^\theta_i\}$, we can compute $\frac{\partial\mathcal{L}^\theta_{SSM}}{\partial\theta}$:

[5] We utilized patches as input (rather than frames) because we believe that homogeneity exists at the pattern level rather than the frame level.

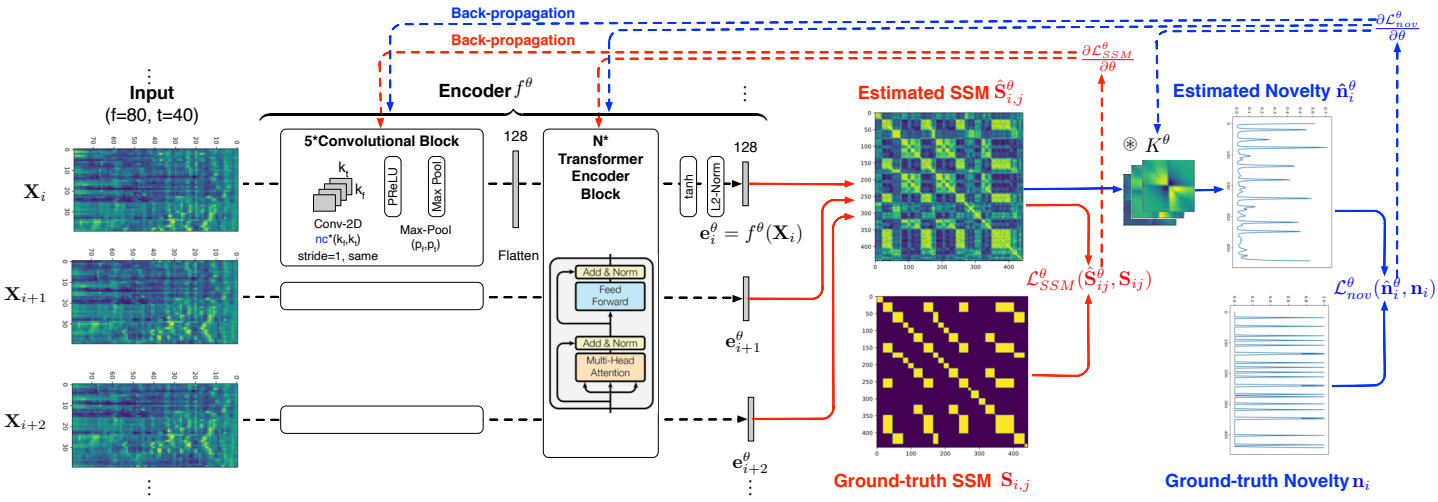

Figure 1. Proposed architecture and training paradigm minimizing a SSM loss $\mathcal{L}_{SSM}^\theta$ and a novety loss $\mathcal{L}_{nov}^\theta$.

$$\frac{\mathcal{L}_{SSM}^\theta}{\theta} = \sum_{i\ j=1}^{T} \frac{\mathcal{L}_{SSM}^\theta}{\mathbf{S}_{ij}^\theta} \left(\frac{\mathbf{S}_{ij}^\theta}{\mathbf{e}_i^\theta} \frac{\mathbf{e}_i^\theta}{\theta} + \frac{\mathbf{S}_{ij}^\theta}{\mathbf{e}_j^\theta} \frac{\mathbf{e}_j^\theta}{\theta} \right) \tag{4}$$

We can then use standard gradient-descent algorithms to optimize θ which will jointly optimize f^θ for all the $\mathbf{X}_i$.

Optimizing directly $\mathbf{S}_{ij}^\theta$ has relationship with Metric Learning / Contrastive Learning in which the A,P,N are chosen based on their similarity (such as in Wang et al. [27]). In comparison, we consider here simultaneously all possible pairs of time as A,P,N. This is actually in line with the fact that we aim at learning features relative to a track (see part 2.4) and we therefore need to consider simultaneously the interaction between all projections $\mathbf{e}_i^\theta$.

2.3 Novelty-loss

We propose to learn the kernels $\mathbf{K}^\theta$ such that when convolved with the estimated SSM $\mathbf{S}_{ij}^\theta$ (see eq.(2)) along its main diagonal the resulting estimated novelty score $\mathbf{n}_i^\theta$ approximate a ground-truth novelty score $\mathbf{n}_i$. This kernel convolution can be simply implemented as an extra convolution layer (without bias) on top of the estimated SSM $\mathbf{S}_{ij}^\theta$ with a sigmoid output activation. We then define the novelty-loss as

$$\mathcal{L}_{nov}^\theta = \frac{1}{T} \sum_{i=1}^{T} BCE(\mathbf{n}_i^\theta\ \mathbf{n}_i) \tag{5}$$

2.4 Relative feature learning

In previous works dealing with feature learning for MSA it is assumed that, once trained, the network f^θ always projects a given segment $\mathbf{X}_i$ in the same way whatever its surrounding context.

We advocate here that for the task of MSA the projection of $\mathbf{X}_i$ should depend on its context. The motivation for doing so is that the features that highlight the temporal structure of a music track usually depend on the track itself. For example, if the instrumentation or the timbre remains constant over the track, the structure may arise from variation of the harmonic content; in other cases, it will be the opposite. Therefore, feature learning for MSA should be made relative-to-a-track.

To let each feature $\mathbf{X}_i$ "know" about surrounding times features $\mathbf{X}_1$ $\mathbf{X}_{i-1}$ $\mathbf{X}_{i+1}$ $\mathbf{X}_T$ we introduce layers of Self-Attention (SA) [35] in our encoder [6] .

2.5 Network architecture f^θ

The architecture of the encoder f^θ is given in Figure 1. It is made of a succession of 5 consecutive convolution blocks followed by N blocks of Transformer-Encoder.

Each convolution block is made of a 2D convolution followed by a PReLU [36] activation and a 2D max-pooling. The kernel size $(k_f\ k_t)$, the number of channels n_c and pooling size $(p_f\ p_t)$) of each layer are the following: layer-1: $(k_f\ k_t)$=(5,5) n_c=32 $(p_f\ p_t)$=(2,2), layer-2: (5,5) 32 (2,2), layer-3: (5,5) 64 (2,2), layer-4: (5,5) 64 (2,2), layer-5: (5,2) 128 (5,2). The output of the last convolutional blocks has dimension (1,1) with n_c=128 channels and is flattened to a 128-dim vector.

Each input $\mathbf{X}_i$ is independently projected using the convolutional blocks. These outputs are then considered as a temporal sequence which is fed to N blocks of Transformer Encoder (each made up of a SA layer with 8 heads, skip-connection, a normalization layer and two fully-connected layers with an internal dimension of 128). The outputs are then passed to a tanh and L2-normalized. They form a sequence of embeddings $\mathbf{e}_i^\theta$ i 1 T with $\mathbf{e}_i^\theta$ R^{128} which are used to compute $\mathbf{S}_{ij}^\theta$.

The size of the kernels $\mathbf{K}^\theta$ is fixed to (41,41) which roughly corresponds to 20s. The kernels $\mathbf{K}^\theta$ are either initialized randomly or initialized with checkerboard kernels similar to the ones of [32]. In this case, checkerboard kernels have the same size (41,41) but are damped with Gaussian function with different σ (randomly chosen in the range $[3s\ 5s]$). We used 3 different kernels $\mathbf{K}^\theta$ which are

[6] Note that the use of the SSM-loss alone does not allows f^θ to encode relative features; this is the task of the SA.

then combined using (1x1) convolution. The diagonal of the resulting feature-map then goes to a sigmoid activation and is considered as the estimated novelty $\mathbf{n}_i^\theta$.

Our architecture remains lightweight with a number of parameters ranging from 268K to 567K depending on the number of Transformer Encoder blocks (from N=0 to 3).

2.6 Training.

We train our network by minimizing jointly the two losses defined by eq. (3) and eq. (5):

$$\mathcal{L}^\theta = \alpha \mathcal{L}^\theta_{SSM} + (1-\alpha)\mathcal{L}^\theta_{nov} \quad (6)$$

We used the ADAM optimizer with a learning rate of 0.001, used early-stopping monitoring $\mathcal{L}^\theta$ on the validation data with a patience of 50 and a maximum of 500 epochs.

Considering that we need the whole sequence of embeddings $\mathbf{e}_i^\theta$ of a track to compute $\mathbf{S}_{ij}^\theta$ and get the gradients $\frac{\mathcal{L}^\theta}{\theta}$, the mini-batch-size m is here defined as the number of tracks. We used a value of m=10 tracks.

2.6.1 Generating ground-truth for training

Ground-truth SSM $\mathbf{S}_{ij}$. The ground-truth SSM, $\mathbf{S}_{ij}$, is constructed using annotated segments (start and end time) and their associated labels. We rely on the homogeneity assumption, i.e. we suppose that all times t_i that fall within a segment are identical since they share the same label. If we denote by $\text{seg}(t_i)$ the segment t_i belongs to and by $\text{label}(\text{seg}(t_i))$ its label, we assign the value $\mathbf{S}_{ij} = 1$ if $\text{label}(\text{seg}(t_i)) = \text{label}(\text{seg}(t_j))$ and 0 otherwise. Note that we could relax this identity constraint to allow representing similarity between labels (for example using an edit distance between labels). This is for example important for RWC-POP dataset, where labels denotes some proximities (`verse A` and `verse B`) but are here considered as different. Also, it could be important to consider the case of non-homogeneity of the repetitions and create a ground-truth $\mathbf{S}_{ij}$ made of "sub-diagonals" rather than "blocks".

Ground-truth novelty score $\mathbf{n}_i$. The ground-truth novelty score, $\mathbf{n}_i$, is also constructed using the annotated segments (start and end time). We set $\mathbf{n}_i$ to 1 when segment changes at time i, 0 otherwise. As proposed by [37] we smooth over time the boundary annotations by applying a low-pass filter with a triangular-shape $\begin{bmatrix} 0.25 & 0.5 & 1 & 0.5 & 0.25 \end{bmatrix}$.

3 Evaluation

We assess here the performance of our proposal using various test sets, compare it to previously published results, conduct an ablation study, and illustrate its results.

3.1 Datasets

For training we used a subset of 693 tracks from the **Harmonix** dataset [38] [7] and the 298 tracks of the **Isophonics** dataset [39]. For testing we used

[7] Given the non-accessibility of Harmonix audio, those have been downloaded from `YouTube` and re-annotation has been necessary because of non-synchronicity of the original annotations.

Datasets	T	S	L	S	L
Harmonix	693	13	17.15		
Isophonics	298	11	15.98		
RWC-Pop-AIST	100	17	14.31		
		Upper		Lower	
SA-Pop (An1)	276	12	15.49	30	5.73
SA-Pop (An2)	175	12	14.64	31	5.67
SA-IA (An1)	444	14	18.32	50	4.43
SA-IA (An2)	244	12.5	18.67	37	7.00
SA-Two (An1)	882	11	18.25	30	6.89
SA-Two (An2)	882	11	17.76	31	6.39

Table 1. Description of the datasets used in our evaluation: number of tracks T, median value of the number of segments per track S , median value of segment duration L in seconds (note that [29] indicate L in number of beats).

- **RWC-Pop-AIST** the 100 tracks of the RWC-Pop [40] with AIST annotations [41] and the following three subsets of the SALAMI [3] dataset:
- **SA-Pop** is the subset of SALAMI tracks with CLASS equal to Popular,
- **SA-IA** is the subset of SALAMI tracks with SOURCE equal to IA (Internet Archive),
- **SA-Two** is the subset of SALAMI tracks with at least two annotations.

For each SALAMI subset we considered the two annotations (An1, An2) and the two levels of flat annotations (Upper, Lower); those correspond to the files `textfile{1,2}_{upper,lowercase}.txt`.

In Table 1 we describe these datasets. According to the values of L our training-sets better match the Upper annotations than the Lower ones of SALAMI. Also, the size of our kernels $\mathbf{K}^\theta$ (roughly 20s., see part 2.5) assumes homogeneous segments of roughly 10s. and are therefore closest to the L of Upper annotations.

3.2 Segment detection from novelty score

To get the estimated segment boundaries from the estimated novelty score $\mathbf{n}_i^\theta$ we used a simple peak-to-mean ratio algorithm similar to [25]. Using the same notations as [25] eq. (5), we compute the mean with a window of duration $2T$=20s, and then detect local peaks with a threshold τ=1.35 and a minimum inter-distance of 7s.

3.3 Performance metrics

We evaluate the performance of segment boundary detection using the common Hit-Rate metrics using precision-windows of 3s and 0.5s. We only display here the Hit-Rate F-measures denoted by `HR3F` and `HR0.5F`. We used `mir_eval` [43] with `mir_eval.segment.detection` ignoring track start and end annotations (`Trim=True`). We point out that without "trimming" (the start and end time) we would gain +3% on average (from .713 to .743 for RWC-Pop).

	RWC-Pop-AIST		SA-Pop		SA-IA		SA-Two		
	HR.5F	HR3F	HR.5F	HR3F	HR.5F	HR3F	HR.5F	HR3F	Annotation
Grill [23, 42] GS1	.506	**.715**	-	-	-	-	.541	.623	Up./An-*
McCallum [25] Unsynch.	-	-	-	-	-	.497	-	-	
Beat-synch.	-	-	-	-	-	**.535**	-	-	
Salamon [30] $\text{DEF}_{0.5,05}/*_{\mu^H\ \gamma^H}$	-	-	-	-	-	-	.337	.563	Up./An-*
Wang [27] scluster/D/eu/mul	.438	.653	.447	.623	-	-	.356	.553	Up./An-*
Buisson [29] HE_0/HE_1	-	.681	-	-	-	-	-	**.597 / .595**	Up./An-1/2
			-	-	-	-	-	**.611 / .600**	Low./An-1/2
Ours (best conf.)	.399	**.713**	.298 / .295	**.631/ .624**	.250 / .261	**.520 / .511**	.231 / .237	.521 / .530	Up./An-1/2
			.296 / .318	.570 / .610	.302 / .336	.547 / .612	.287 / .287	.589 / .589	Low./An-1/2
Ablation study N									
N=3/α=0.5/K:train-Init:chck		**.713**		.532		.472		.448	Up./An-1
N=2/α=0.5/K:train-Init:chck		.701		.535		.474		.449	Up./An-1
N=1/α=0.5/K:train-Init:chck		.677		**.631**		**.520**		**.521**	Up./An1
N=0/α=0.5/K:train-Init:chck		.696		.535		.459		.443	Up./An-1
Ablation study α									
N=3/α=1/K:train-Init:chck		.154		.121		.102		.111	Up./An-1
N=3/α=0/K:train-Init:chck		.007		.120		.026		.095	Up./An-1
Ablation study $\mathbf{K}^\theta$									
N=3/α=0.5/K:train-Init:randn		**.713**		.543		.470		.457	Up./An-1
N=1/α=0.5/K:train-Init:randn		.709		.547		.470		.457	Up./An-1
N=3/α=0.5/K:fix-Init:chck		.330		.250		.199		.196	Up./An-1

Table 2. Results of segment boundary detection using various test-sets and configurations

3.4 Comparison with previous works

In the following we will compare our results with the ones previously published by Grill and Schlüter in [23, 42], McCallum et al. in [25], Salamon et al. in [30], Wang et al. in [27] and Buisson et al. in [29]. We first check if their test-sets match ours.

For SA-Pop, Wang [27] used *"a subset with 445 annotated songs (from 274 unique songs) in the "popular" genre"*. This roughly matches our SA-Pop (An1)+(An2) which provides 276+175=451 annotations. They used the Upper-case annotations (personal communication).

For SA-IA, McCallum [25] used *"the internet archive portion of the SALAMI dataset (SALAMI-IA) consisting of 375 hand annotated recordings"*. This is much lower than our SA-IA (An1)+(An2) which provides 444+244=688 annotations. Moreover, it is not clear whether they used the Upper, Lower or Functional annotations.

Finally, for SA-Two, Salamon [30] Table 3 used the Upper-case annotations of tracks with at least 2 annotations (884 tracks); Wang et al. [27] *"we treat each annotation of a song separately, yielding 2243 annotated songs in total"* and Buisson et al. [29] used the Upper and Lower-case annotations of tracks with at least 2 annotations (884 tracks). This roughly corresponds to our SA-Pop (An1)+(An2) which has 882 tracks.

3.5 Results and discussions

Results are given in Table 2. The upper part shows previously published results, although not all systems were evaluated on all test sets. The middle part shows the results achieved with the best configuration of our system.

For **RWC-Pop-AIST**, we obtained a HR3F= .713 [8] which is comparable to those of Grill and Schlüter (.715). However, for HR.5F our results are below (.399 < .506). This can be explained by the fact that the hop-size of our features $\mathbf{X}_i$ was chosen large (0.5s) and does not allow to have a precise boundary estimation. We have chosen a large hop size to reduce the size of $\mathbf{S}_{ij}^\theta$ (hence the computation time and memory footprints); it also allows to keep the size of the $\mathbf{K}^\theta$ manageable. Because of this, all our results with HR.5F are actually low. Therefore, we only discuss the results for HR3F in the following.

For **SA-Pop**, we obtained a HR3F of .631/.624 [9] for the two Upper annotations (Up./An-1/2) which is slightly above those of Wang et al. (.623). For the two lower annotations (Low./An-1/2) we get a HR3F of .570/.610 [10]. Wang et al. does not provide these results.

For **SA-IA**, we obtained a HR3F of .520/.511 [11] for the two Upper annotations and .547/.612 [12] for the two Lower annotations. This has to be compared to the .497 (unsynchronized) and .535 (beat-synchronized) obtained by McCallum et al., but as explained, it is not clear whether they used Upper, Lower or Functional annotations.

For **SA-Two**, we obtained a HR3F of .521/.530 [13] for the two Upper annotations. This is slightly lower than the results of Wang et al. (.553), Salamon et al. (.563), Buisson et al. (.597) and largely below the ones of Grill and Schlüter (.623). For the Lower annotations, we obtained a HR3F of .589/.589 [14] which is slightly below the ones of Buisson et al. (.611). It should be noted however that in our work we didn't used any data from SALAMI, neither for training or validation (such as early stopping).

For SA-IA and SA-two, our results are higher for the Lower annotations than the Upper ones. This is surprising since according to Table 1 the characteristics (L value) of our training sets are closer to the Upper case. Also (see footnotes 8–14), our algorithm tends to over-segment when

[8] The Precision and Recall at 3seconds are P3F=.735, R3F=0.715

[9] P3F=.581, R3F=0.760/ P3F=.566, R3F=0.771 → over-segmentation

[10] P3F=.860, R3F=0.468/ P3F=.877, R3F=0.497 → under-segment.

[11] P3F=.435, R3F=0.718/ P3F=.411, R3F=0.751 → over-segment.

[12] P3F=.811, R3F=0.451/ P3F=.756, R3F=0.546 → under-segment.

[13] P3F=.433, R3F=0.749/ P3F=.442, R3F=0.754→ over-segment.

[14] P3F=.768, R3F=0.523/ P3F=.768, R3F=0.523→ under-segment.

considering the Upper annotation and under-segment when considering the Lower ones. Our kernel size is actually between the L values of the Upper and Lower annotations.

3.6 Ablation study

In the lower part of Table 2 we perform an ablation study of our system. For the SA-{Pop,IA,Two} test-sets, we only perform the study using the Upper/An1 annotations

We first check the optimal number N 0 1 2 3 of layers of Transformer Encoder. We see that for all test-sets the use of Transformer Encoder ($N > 0$) is beneficial. For RWC-Pop-AIST, the optimal number is N=3 while for all three SA-{Pop,IA,Two} test-sets it is always N=1.

We then check whether jointly optimizing the two losses L^{θ}_{SSM} and $\mathcal{L}^{\theta}_{nov}$ of eq. (6) is necessary. We considered three cases: α=1 (only optimizing L^{θ}_{SSM}), α=0.5 (optimizing both), α=0 (only optimizing $\mathcal{L}^{\theta}_{nov}$). For all test-sets, we see that optimizing jointly the two losses is highly beneficial: for example, for RWC-Pop-AIST, HR3F=.713 with α=0.5, .154 with α=1 and .007 for α=0.

Finally, we check various configurations of the kernels $\mathbf{K}^{\theta}$. $\mathbf{K}^{\theta}$ is either [K:train-Init:chck]: trained starting from checkerboard kernels initialisation, [K:train-Init:randn]: trained starting from random initialisations, [K:fix-Init:chck]: fixed (not trained) to checkerboard kernels (we still train the 1x1 convolution). We see that for all test-sets it is beneficial to train $\mathbf{K}^{\theta}$ (the worst results are obtained with [K:fix-Init:chck]). For RWC-Pop-AIST, the results are the same whether kernels are initialized randomly or with checkerboard kernels. For SA-{Pop,IA,Two} the checkerboard kernels initialization is beneficial.

3.7 Examples

Figure 2 illustrates the three kernels $\mathbf{K}^{\theta}$ learned using the [N=3/α=0.5/K:train-Init:chck] configuration. As one can see, while the middle one looks close to the classical checkerboard kernel of Foote [32] (but with an emphasis on the diagonal), the first seems to highlight the transition from a non-homogeneous to an homogeneous part; while the third seems a re-scaled version of the second (homogeneity at a larger scale). Figure 3 illustrates the $\mathbf{S}^{\theta}_{ij}$ and $\mathbf{n}^{\theta}_{i}$ obtained by our system on track-01 from RWC-Pop-AIST (chosen as the first item of our test-set). We compare the results when trained in the [N=3 / α=0.5 / K:train-Init:chck] configuration and with the untrained system using [K:fix-Init:chck] for the kernels. For comparison we indicate the ground-truth $\mathbf{S}_{ij}$ and $\mathbf{n}_{i}$. In this figure, the benefits of training both L^{θ}_{SSM} and$\mathcal{L}^{\theta}_{nov}$ appears clearly.

Reproducibility. The code and the datasets used in this work are available at: https://github.com/geoffroypeeters/ssmnet_ISMIR2023

4 Conclusion

In this work, we proposed a simple approach for deep learning-based Music Structure Analysis algorithm: we learn an encoder f^{θ} such that the resulting learned features allow to best approximate a ground-truth SSM; we jointly learn segmentation kernels that when applied to the estimated SSM we best approximate a ground-truth novelty score. We also propose to learn relative features, i.e. features relative to a track, by introducing Self-Attention layers in our encoder. According to HR3F, our results are either better than previous state-of-the-art (SA-Pop, SA-IA unsynchronous), similar (RWC-Pop-AIST) or worst (SA-Two). Our approach has the advantage to be lightweight (around 500K parameters) and based on criteria which are semantically linked to the task of MSA. Future works will concentrate on making our approach applicable to finer temporal resolutions, therefore allowing improving our performances at HR.5F.

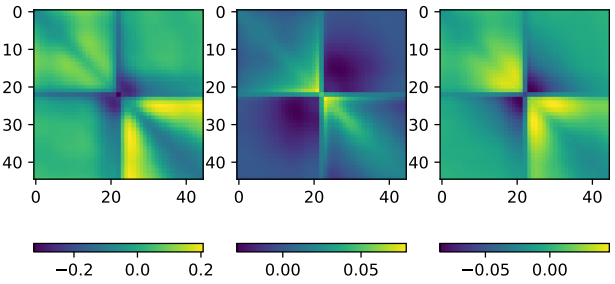

Figure 2. The three kernels $\mathbf{K}^{\theta}$ learned using the [N=3 / α=0.5 / K:train-Init:chck] configuration.

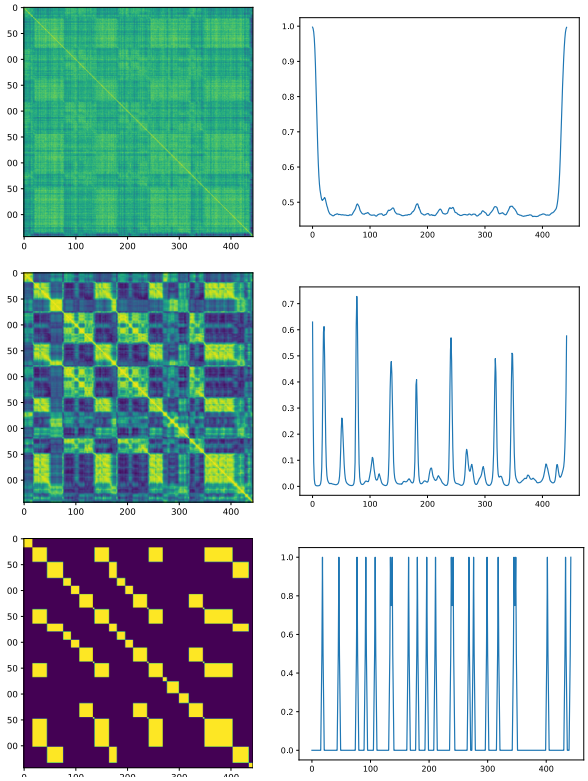

Figure 3. [Top] $\mathbf{S}^{\theta}_{ij}$ and $\mathbf{n}^{\theta}_{i}$ obtained with untrained system using [K:fix-Init:chck] for the kernels, [Middle] same with [N=3 / α=0.5 / K:train-Init:chck], [Bottom] ground-truth $\mathbf{S}_{ij}$ and $\mathbf{n}_{i}$.

5 References

[1] J. Foote, "Visualizing music and audio using self-similarity," in *Proc. of ACM Multimedia*, Orlando, Florida, USA, 1999, pp. 77–80.

[2] G. Peeters and E. Deruty, "Is music structure annotation multi-dimensional ? a proposal for robust local music annotation," in *Proc. of LSAS (International Workshop on Learning the Semantics of Audio Signals)*, Graz, Austria, 2009.

[3] J. B. L. Smith, J. Burgoyne, I. Fujinaga, D. Roure, and J. S. Downie, "Design and creation of a large-scale database of structural annotations," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Miami, Florida, USA, 2011.

[4] B. McFee, O. Nieto, M. M. Farbood, and J. P. Bello, "Evaluating hierarchical structure in music annotations," *Frontiers in Psychology*, vol. 8, p. 1337, 2017.

[5] M. Bruderer, M. McKinney, and A. Kohlrausch, "Structural boundary perception in popular music," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Victoria, Canada, 2006.

[6] G. Peeters, A. Laburthe, and X. Rodet, "Toward automatic music audio summary generation from signal analysis," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Paris, France, 2002, pp. 94–100.

[7] G. Peeters, D. Fenech, and X. Rodet, "MCIpa: A music content information player and annotator for discovering music," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Philadelphia, PA, USA, 2008.

[8] G. Peeters, F. Cornu, D. Tardieu, C. Charbuillet, J. J. Burred, M. Ramona, M. Vian, V. Botherel, J.-B. Rault, and J.-P. Cabanal, "A multimedia search and navigation prototype, including music and video-clips," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Porto, Portugal, October 2012.

[9] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web service for active music listening improved by user contributions." in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Miami, Florida, USA, 2011.

[10] M. Mueller and N. Jiang, "A scape plot representation for visualizing repetitive structures of music recordings," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Porto, Portugal, 2012.

[11] M. Mauch, K. Noland, and S. Dixon, "Using musical structure to enhance automatic chord transcription," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Kobe, Japan, 2009.

[12] Z. Rafii and B. Pardo, "Repeating pattern extraction technique (repet): A simple method for music/voice separation," *Audio, Speech and Language Processing, IEEE Transactions on*, vol. 21, no. 1, pp. 73–84, January 2013.

[13] 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 *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Brighton, UK, 2019, pp. 481–485.

[14] J. Paulus, M. Müller, and A. Klapuri, "Audio-based music structure analysis," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Utrecht, The Netherlands, 2010.

[15] G. Sargent, F. Bimbot, and E. Vincent, "A regularity-constrained viterbi algorithm and its application to the structural segmentation of songs," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Miami, Florida, USA, 2011.

[16] 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.

[17] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. of IEEE ICME (International Conference on Multimedia and Expo)*, New York City, NY, USA, 2000.

[18] F. Kaiser and T. Sikora, "Music structure discovery in popular music using non-negative matrix," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Utrecht, The Netherlands, 2010.

[19] B. McFee and D. P. W. Ellis, "Learning to segment songs with ordinal linear discriminant analysis," in *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Florence, Italy, 2014.

[20] M. Müller, N. Jiang, and P. Grosche, "A robust fitness measure for capturing repetitions in music recordings with applications to audio thumbnailing," *Audio, Speech and Language Processing, IEEE Transactions on*, vol. 21, no. 3, pp. 531–543, 2013.

[21] J. Serrà, 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, Aug. 2014.

[22] K. Ullrich, J. Schlüter, and T. Grill, "Boundary Detection in Music Structure Analysis using Convolutional

Neural Networks," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Taipei, Taiwan, 2014.

[23] T. Grill and J. Schlüter, "Music Boundary Detection Using Neural Networks on Combined Features and Two-Level Annotations," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Malaga, Spain, 2015.

[24] A. Cohen-Hadria and G. Peeters, "Music Structure Boundaries Estimation Using Multiple Self-Similarity Matrices as Input Depth of Convolutional Neural Networks," in *AES International Conference on Semantic Audio*, Erlangen, Germany, June, 22–24, 2017.

[25] M. C. McCallum, "Unsupervised Learning of Deep Features for Music Segmentation," in *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Brighton, UK, May 2019.

[26] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 815–823, iSSN: 1063-6919.

[27] J.-C. Wang, J. B. L. Smith, W.-T. Lu, and X. Song, "Supervised metric learning for music structure features," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Online, November, 8–12 2021.

[28] J.-C. Wang, Y.-N. Hung, and J. B. L. Smith, "To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions." in *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Virtual and Singapore, May 2022.

[29] M. Buisson, B. McFee, S. Essid, and H.-C. Crayencour, "Learning multi-level representations for hierarchical music structure analysis," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, 2022.

[30] J. Salamon, O. Nieto, and N. J. Bryan, "Deep embeddings and section fusion improve music segmentation," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Online, November, 8–12 2021.

[31] G. Peeters and F. Angulo, "Ssm-net: Feature learning for music structure analysis using a self-similarity-matrix based loss," in *Late-Breaking/Demo Session of ISMIR (International Society for Music Information Retrieval)*, 2022.

[32] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. of IEEE ICME (International Conference on Multimedia and Expo)*, New York City, NY, USA, 2000, pp. 452–455.

[33] F. Kaiser and G. Peeters, "Multiple hypotheses at multiple scales for audio novelty computation within music," in *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Vancouver, British Columbia, Canada, May 2013.

[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.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[36] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 972–981.

[37] J. Schlüter and S. Böck, "Improved musical onset detection with Convolutional Neural Networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6979–6983, iSSN: 2379-190X.

[38] O. Nieto, M. McCallum, M. E. P. Davies, A. Robertson, A. Stark, and E. Egozy, "The Harmonix Set: Beats, Downbeats, and Functional Segment Annotations of Western Popular Music," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Delft, The Netherlands, 2019.

[39] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Klozali, D. Tidhar, and M. Sandler, "Omras2 metadata project 2009," in *Late-Breaking/Demo Session of ISMIR (International Society for Music Information Retrieval)*, Kobe, Japan, 2009.

[40] M. Goto, "Development of the RWC Music Database," *Proc. of ICA (18th International Congress on Acoustics)*, 2004.

[41] ——, "Aist annotation for the rwc music database," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Victoria, BC, Canada, 2006, pp. pp.359–360.

[42] T. Grill and J. Schlüter, "Structural segmentation with convolutional neural networks MIREX submission," 2015.

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